VM/CMS ISIS
REXX EXECs

W. Simon
VM/CMS ISIS
REXX EXECs

W. Simon
1. Introduction
2. Data set naming conventions

**REXX EXECs**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDQAFLST</td>
<td>Print CDS/ISIS Library Acquisition system</td>
<td>CDQAFLST</td>
</tr>
<tr>
<td>CDPDLLST</td>
<td>Print VR/VMS ISIS REXX EXECs</td>
<td>CDPLLST</td>
</tr>
<tr>
<td>CDDEFRAP</td>
<td>Define Backup Tapes</td>
<td>CDDEFRAP</td>
</tr>
<tr>
<td>CDGDLIST</td>
<td>Print CDS/ISIS Data Manipulation Language Manual documentation</td>
<td>CDGDLIST</td>
</tr>
<tr>
<td>CDGETGDC</td>
<td>Get Existing Member of the Simulated CDS</td>
<td>CDGETGDC</td>
</tr>
<tr>
<td>CDIMLIST</td>
<td>Print VR/VMS ISIS System Installation Documentation</td>
<td>CDIMLIST</td>
</tr>
<tr>
<td>CDIPPL</td>
<td>Recover Extrapartition data sets</td>
<td>CDIPPL</td>
</tr>
<tr>
<td>COPEP</td>
<td>Process Extrapartition data sets</td>
<td>COPEP</td>
</tr>
<tr>
<td>COPILIST</td>
<td>Print CDS/ISIS Photocomposition Documentation</td>
<td>COPILIST</td>
</tr>
<tr>
<td>COPUTGDC</td>
<td>Put New Member at the Simulated CDS</td>
<td>COPUTGDC</td>
</tr>
<tr>
<td>CORCVREP</td>
<td>Backup and log CICS Extrapartition data sets</td>
<td>CORCVREP</td>
</tr>
<tr>
<td>CORGRLPS</td>
<td>Reorganize and backup a partitioned data set</td>
<td>CORGRLPS</td>
</tr>
<tr>
<td>CORTV53</td>
<td>List of Partitioned/Sequential data set</td>
<td>CORTV53</td>
</tr>
<tr>
<td>CORTV61</td>
<td>Hitfile post processor (sorted output)</td>
<td>CORTV61</td>
</tr>
<tr>
<td>CORTV6IN</td>
<td>Hitfile post processor (unsorted output)</td>
<td>CORTV6IN</td>
</tr>
<tr>
<td>COTHDLST</td>
<td>Print CDS/ISIS Thesaurus Subsystem manual documentation</td>
<td>COTHDLST</td>
</tr>
<tr>
<td>COPPLST</td>
<td>Print CDS/ISIS Terminal Operator manual documentation</td>
<td>COPPLST</td>
</tr>
<tr>
<td>COPPOS</td>
<td>Update a partitioned data set</td>
<td>COPPOS</td>
</tr>
<tr>
<td>COUTL01</td>
<td>Load Field Select Tables and Print forests</td>
<td>COUTL01</td>
</tr>
<tr>
<td>COUTL02</td>
<td>Print (ISIS) documentation</td>
<td>COUTL02</td>
</tr>
<tr>
<td>COUTL05</td>
<td>Integrated Hitfile sort</td>
<td>COUTL05</td>
</tr>
<tr>
<td>COYSLIST</td>
<td>Print Switching to YSAM documentation</td>
<td>COYSLIST</td>
</tr>
</tbody>
</table>

**TABLE OF CONTENTS PAGE NO PREFIX**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVBKUPIF</td>
<td>Backup Inverted file on tape</td>
<td>CVBKUPIF</td>
</tr>
<tr>
<td>CVBKUPMF</td>
<td>Backup Master file on tape</td>
<td>CVBKUPMF</td>
</tr>
<tr>
<td>CVBKUPTR</td>
<td>Backup Transaction file on tape</td>
<td>CVBKUPTR</td>
</tr>
<tr>
<td>CVCHECK</td>
<td>Global validation of Transaction file (for one data base)</td>
<td>CVCHECK</td>
</tr>
<tr>
<td>CVDELDEF</td>
<td>Delete/Define a VSAM file</td>
<td>CVDELDEF</td>
</tr>
<tr>
<td>CVEXPORT</td>
<td>ISIS to ISO Format conversion</td>
<td>CVEXPORT</td>
</tr>
<tr>
<td>CVGLCHA</td>
<td>Update Master file (ISIS input format - Global changes)</td>
<td>CVGLCHA</td>
</tr>
<tr>
<td>CVGFIEN</td>
<td>Create Inverted file</td>
<td>CVGFIEN</td>
</tr>
<tr>
<td>CVHFMTR</td>
<td>ISO to ISIS Format conversion</td>
<td>CVHFMTR</td>
</tr>
<tr>
<td>CVLISTF</td>
<td>List/Dump Inverted file</td>
<td>CVLISTF</td>
</tr>
<tr>
<td>CVLISTFM</td>
<td>List/Dump Master/Transaction file records</td>
<td>CVLISTFM</td>
</tr>
<tr>
<td>CVLOADTR</td>
<td>Transaction file - Restore cluster</td>
<td>CVLOADTR</td>
</tr>
<tr>
<td>CVPLMTR</td>
<td>Permutated Dictionary generator</td>
<td>CVPLMTR</td>
</tr>
<tr>
<td>CVPMTHMS</td>
<td>Photocomposition (ATMS Text Editor)</td>
<td>CVPMTHMS</td>
</tr>
<tr>
<td>CVPHDROC</td>
<td>Photocomposition</td>
<td>CVPHDROC</td>
</tr>
<tr>
<td>CVPROOF</td>
<td>Proof copy print of all records in error (for one data base)</td>
<td>CVPROOF</td>
</tr>
<tr>
<td>CVQUERY</td>
<td>Batch retrieval (unsorted output)</td>
<td>CVQUERY</td>
</tr>
<tr>
<td>CVQUERYS</td>
<td>Batch retrieval (sorted output - diek sort)</td>
<td>CVQUERYS</td>
</tr>
<tr>
<td>CVQUERYT</td>
<td>Batch retrieval (sorted output - tape sort)</td>
<td>CVQUERYT</td>
</tr>
<tr>
<td>CVQSAVE</td>
<td>Batch retrieval (sorted output - diek sort) input from copy of Search</td>
<td>CVQSAVE</td>
</tr>
<tr>
<td>CVRESTR</td>
<td>Transaction file - Restore data base</td>
<td>CVRESTR</td>
</tr>
<tr>
<td>CVRYT36</td>
<td>Load Lookup file</td>
<td>CVRYT36</td>
</tr>
<tr>
<td>CYTERMNL</td>
<td>Process Entry Processor Extrapartition data set (Input TPINP produced by</td>
<td>CYTERMNL</td>
</tr>
<tr>
<td>CVTRINIT</td>
<td>Initialize Transaction file</td>
<td>CVTRINIT</td>
</tr>
<tr>
<td>CVUPDF</td>
<td>Update Inverted file Dictionary</td>
<td>CVUPDF</td>
</tr>
<tr>
<td>CVUPDATIC</td>
<td>Update Master &amp; Inverted file (ISIS input forest)</td>
<td>CVUPDATIC</td>
</tr>
<tr>
<td>CVUPDATE</td>
<td>Update Master &amp; Inverted file (CDS input forest)</td>
<td>CVUPDATE</td>
</tr>
<tr>
<td>CVUTL11</td>
<td>Translate Hitfile headings</td>
<td>CVUTL11</td>
</tr>
<tr>
<td>CVUTL12</td>
<td>Insert External Cross-references into Hitfile</td>
<td>CVUTL12</td>
</tr>
<tr>
<td>CVUTL13</td>
<td>Master file extraction and reformatting</td>
<td>CVUTL13</td>
</tr>
<tr>
<td>CVUTL17</td>
<td>Convert YRAM Access file</td>
<td>CVUTL17</td>
</tr>
<tr>
<td>CVUTL20</td>
<td>Analyze attributes of VSAM file</td>
<td>CVUTL20</td>
</tr>
</tbody>
</table>
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXECS

1. INTRODUCTION

This manual describes the standard VM/CMS ISIS REXX EXECS.

They have been written in such a way that they are as much
installation independent as possible, by using parameters whenever
appropriate.

EXECS beginning with 'CV' are VSAM EXECS (e.g. CYQUERY)
and those EXECS which are independent from the Access Method begin
with 'CD' (e.g. CDUTIL81).

The parameters used in the EXECS are set in an EXEC called PROLOG
which is called from every VM/CMS ISIS REXX EXEC. The parameters may
be displayed by typing PROLOG, i.e. executing PROLOG. At installation
time PROLOG has to be modified to reflect the local needs. In particular,
the available VSAM disk (user, hexadecimal address) as well as the name
of the VSAM catalog have to be specified.

A VM/CMS ISIS EXEC is executed under CMS by typing:

name operand(s) / options

e.g.

CYQUERY VS801A / DBN=PREP CLN=PREP

where:

name is the name of the EXEC. The names of CDS/ISIS are retained.

operands are information about the CMS data sets used. It is different
in every EXEC and may be found there. In the particular example
the input is expected on the file VS801 CARDS. Note, that
there are reasonable defaults. CYQUERY ( DBN=PREP CLN=PREP
would expect input on VS801 CARDS).

options are what the parameters of the cataloged procedures are in
CDS/ISIS. They follow an opening parenthesis and are of the
form keyword=value. Options are separated by blanks or commas.
Available keywords may be displayed by executing PROLOG.

2. DATA SET NAMING CONVENTIONS

The following standard data set names used by the REXX EXECS:

VM/CMS ISIS (Mainframe Version)
VM/CMS ISIS data set

VSAM data sets

- FDT file
- Transaction file
- Master file
- Master file log file
- Inverted file
- Inverted file log file
- Any file
- Lookup file
- Alternate master file

CMS data sets

- Transaction file backup
- Sort tables and print formats file
- Link file

At CERN, pfx is set to ISIS, scin to ISIS, cin to PERI, anycin to BOOK, ikcin to PERI, acin to PERI and DBN to PERI.

###

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

```rexx
/*
/ * VM/CMS ISIS - PRINT REXX EXECs
/ */

ADDRESS COMMAND

"FILEDEF * CLEAR"
"FILEDEF CARDS DISK CP4TTC DOC *= "
"FILEDEF CARDS DISK CP4INC DOC *= APPEND"

"FILEDEF CARDS DISK RMC01 DOC *= APPEND"
// DD DSN=PROCLIB(COADL1LIST),DISP=SHR *

"FILEDEF CARDS DISK RMC02 DOC *= APPEND"
"FILEDEF CARDS DISK CDCL1LIST EXEC *= APPEND"
"FILEDEF CARDS DISK CP452 DOC *= APPEND"
"FILEDEF CARDS DISK CODEFTAP EXEC *= APPEND"
"FILEDEF CARDS DISK CP444 DOC *= APPEND"
"FILEDEF CARDS DISK CDCL1LIST EXEC *= APPEND"
"FILEDEF CARDS DISK RMC03 DOC *= APPEND"
"FILEDEF CARDS DISK CDCL1LIST EXEC *= APPEND"
"FILEDEF CARDS DISK CP451 DOC *= APPEND"
"FILEDEF CARDS DISK CODEFT0DG EXEC *= APPEND"
"FILEDEF CARDS DISK RMC04 DOC *= APPEND"
"FILEDEF CARDS DISK CODMLIST EXEC *= APPEND"
"FILEDEF CARDS DISK CP441 DOC *= APPEND"
"FILEDEF CARDS DISK CP01L EXEC *= APPEND"
"FILEDEF CARDS DISK CP481 DOC *= APPEND"
"FILEDEF CARDS DISK CP0M EXEC *= APPEND"
"FILEDEF CARDS DISK RMC12 DOC *= APPEND"
"FILEDEF CARDS DISK CP0L EXEC *= APPEND"
"FILEDEF CARDS DISK CP482 DOC *= APPEND"
"FILEDEF CARDS DISK CP0UTDG EXEC *= APPEND"
"FILEDEF CARDS DISK RMC14 DOC *= APPEND"
"FILEDEF CARDS DISK COREVTMP EXEC *= APPEND"
"FILEDEF CARDS DISK RMC19 DOC *= APPEND"
"FILEDEF CARDS DISK CDML1LIST EXEC *= APPEND"

"FILEDEF CARDS DISK RMC21 DOC *= APPEND"
// DD DSN=PROCLIB(CORRGRS),DISP=SHR *

"FILEDEF CARDS DISK CP5C3 DOC *= APPEND"
// DD DSN=PROCLIB(CORTYS3),DISP=SHR *
```

VM/CMS ISIS (Mainframe Version)
"FILEDEF CARDS DISK RNC67 DOC *(APPEND*)
  DD DSN=PROCLIB(CRTV61),DISP=SHR
#
"FILEDEF CARDS DISK RNC68 DOC *(APPEND*)
  DD DSN=PROCLIB(CRTV63),DISP=SHR
#
"FILEDEF CARDS DISK CP446 DOC *(APPEND*)
"FILEDEF CARDS DISK COTNLST EXEC *(APPEND*)
"FILEDEF CARDS DISK RMC72 EXEC *(APPEND*)
"FILEDEF CARDS DISK CDTPLST EXEC *(APPEND*)
#
"FILEDEF CARDS DISK RMC84 EXEC *(APPEND*)
  DD DSN=PROCLIB(CRPDPOS),DISP=SHR
#
"FILEDEF CARDS DISK RMC86 DOC *(APPEND*)
"FILEDEF CARDS DISK COTNL01 EXEC *(APPEND*)
"FILEDEF CARDS DISK CP446 DOC *(APPEND*)
"FILEDEF CARDS DISK COTNL02 EXEC *(APPEND*)
#
"FILEDEF CARDS DISK CP345 DOC *(APPEND*)
"FILEDEF CARDS DISK CDVGLST EXEC *(APPEND*)
"FILEDEF CARDS DISK CP462 DOC *(APPEND*)
"FILEDEF CARDS DISK CVBUP1F EXEC *(APPEND*)
"FILEDEF CARDS DISK CP463 DOC *(APPEND*)
"FILEDEF CARDS DISK CVBUPHF EXEC *(APPEND*)
"FILEDEF CARDS DISK CP464 DOC *(APPEND*)
"FILEDEF CARDS DISK CVBUPTR EXEC *(APPEND*)
"FILEDEF CARDS DISK CP465 DOC *(APPEND*)
"FILEDEF CARDS DISK CVCHECK EXEC *(APPEND*)
"FILEDEF CARDS DISK CP466 DOC *(APPEND*)
"FILEDEF CARDS DISK CVEDEF EXEC *(APPEND*)
"FILEDEF CARDS DISK CP442 DOC *(APPEND*)
"FILEDEF CARDS DISK CVEXPORT EXEC *(APPEND*)
"FILEDEF CARDS DISK CP467 DOC *(APPEND*)
"FILEDEF CARDS DISK CVGEBCHA EXEC *(APPEND*)
"FILEDEF CARDS DISK CP468 DOC *(APPEND*)
"FILEDEF CARDS DISK CVGERIF EXEC *(APPEND*)
"FILEDEF CARDS DISK CP469 DOC *(APPEND*)
"FILEDEF CARDS DISK CVIMPORT EXEC *(APPEND*)
"FILEDEF CARDS DISK CP409 DOC *(APPEND*)
"FILEDEF CARDS DISK CVLISTIF EXEC *(APPEND*)
#*
"FILEDEF CARDS DISK CP410 DOC *(APPEND*)

VM/CMS ISIS (Mainframe Version)
"FILEDEF CARDS DISK CP426 DOC * (APPEND*)
   DD DSN=PROCLIB(CVUHL2),DISP=SHR
/
"FILEDEF CARDS DISK CP426 DOC * (APPEND*)
"FILEDEF CARDS DISK CVUHL3 EXEC * (APPEND*)
/
"FILEDEF CARDS DISK CP447 DOC * (APPEND*)
   DD DSN=PROCLIB(CVUHL20),DISP=SHR
/
"FILEDEF CARDS DISK CP438 DOC * (APPEND*)
   DD DSN=PROCLIB(CVUHL17),DISP=SHR
/
"FILEDEF CARDS DISK CP427 DOC * (APPEND*)
   DD DSN=PROCLIB(CVYSP00),DISP=SHR
/
"FILEDEF CARDS DISK CP419 DOC * (APPEND*)
   DD DSN=PROCLIB(CVYSP01),DISP=SHR
/
"FILEDEF CARDS DISK CP428 DOC * (APPEND*)
"FILEDEF CARDS DISK CVYSP01 EXEC * (APPEND*)
/
"FILEDEF CARDS DISK CP429 DOC * (APPEND*)
   DD DSN=PROCLIB(CVYSP015),DISP=SHR
/
"FILEDEF CARDS DISK CP430 DOC * (APPEND*)
   DD DSN=PROCLIB(CVYSP011),DISP=SHR
/
"FILEDEF CARDS DISK CP431 DOC * (APPEND*)
   DD DSN=PROCLIB(CVYSP02),DISP=SHR
/
"FILEDEF CARDS DISK CP432 DOC * (APPEND*)
"FILEDEF CARDS DISK CVYSP04 EXEC * (APPEND*)
/
"FILEDEF CARDS DISK CP433 DOC * (APPEND*)
   DD DSN=PROCLIB(CVYSP06),DISP=SHR
/
"FILEDEF CARDS DISK CP436 DOC * (APPEND*)
"FILEDEF CARDS DISK CVYSP06 EXEC * (APPEND*)
"FILEDEF CARDS DISK CP438 EXEC * (APPEND*)
"FILEDEF CARDS DISK CP439 EXEC * (APPEND*)
"FILEDEF CARDS DISK CP43A DOC * (APPEND*)
"FILEDEF CARDS DISK CVYSP10 EXEC * (APPEND*)

VM/CMS ISIS (Mainframe Version)
/* ISIS/CMS - DEFINE THE NAMES FOR 3 BACKUP TAPES */
ADDRESS COMMAND

ARG parameters

PARSE SOURCE . . execname .
tape. ="
tapes =""

PARSE VAR parameters fileclin tape.1 tape.2 tape.3 .

IF LENGTH(fileclin) = 6 THEN DO
  SAY "**** error in" execname "****"
  SAY "**** The filename of the BACKUP file must",
    "be exactly 6 characters long."
  RETURN 16
END

DO index=1 TO 3
  IF tape.index = "" THEN DO
    SAY "**** error in" execname "****"
    SAY "**** The BACKUP tape" index "is empty"
    RETURN 16
  END
END

tapes = tapes||tape.index"
END

"GLOBALY SELECT #EXEC SETLP" fileclin tapes

RETURN 0

---

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

/*
/* COD/ISIS - PRINT DATA MANIPULATION LANGUAGE MANUAL (RELEASE 4.6)
/*
/*
ADDRESS COMMAND

"FILEDEF CARDS DISK DML DOC "
"EXEC CDTUL82 (ULC=4,ADJ=8,SIDE=S,LL=88,LP=63,XEROX=N"
util2returncode = RC
IF util2returncode = 0 THEN EXIT util2returncode

"FILEDEF # CLEAR"
RETURN util2returncode

---

VM/CMS ISIS (Mainframe Version)
ADDRESS COMMAND

"FILEDEF = CLEAR"
"FILEDEF CARDS DISK NEWGDTIT DCC *"
"FILEDEF CARDS DISK NEWGDTOC DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDFOR DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDFHS DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDB81 DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDB82 DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDB83 DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDB84 DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDB85 DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDAPI DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDAPI DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDAPI DCC = (APPEND)*
"FILEDEF CARDS DISK NEWGDAPI DCC = (APPEND)*

**EXEC CUTOFF2 (ULC=L, ADJ=0, SIDE=S, LL=88, LP=53, XEROX=N)**

util@2returncode = RC
IF util@2returncode = 0 THEN DO
EXIT util@2returncode
END
"FILEDEF = CLEAR"
RETURN util@2returncode

VM/CMS ISIS (Mainframe Version)

ADDRESS COMMAND

ARG parameters
PARSE SOURCE . execname .
result = ""
PARSE YAR parameters operands "(" parameters
PARSE YAR operands caller operands
PARSE YAR parameters in fm .
IF LENGTH(fm) = 8 & LENGTH(fm) = 6 THEN DO
SAY "******** error in" execname "called from" caller "*******"
SAY "******** The filename of the BACKUP file must be 6 or 8 characters long."
result = 16
RETURN result
END
ft = "BACKUP"
IF fm = "" THEN fm = "*
IF LENGTH(fm) = 1 THEN DO
SAY "******** error in" execname "called from" caller "*******"
SAY "******** The filename of the BACKUP file must be 1 character."
RETURN 16
END
pre = SUBSTR(fm,1,2)
cin = SUBSTR(fm,3,4)
gen = SUBSTR(fm,7,2)
IF gen = "" THEN gen = "00"
IF DATATYPE(gen) = "NUM" THEN DO
SAY "******** error in" execname "called from" caller "*******"
SAY "******** The generation must be numeric."
result = 16
RETURN result
END
IF pre = "TR" & pre = "MF" & pre = "IF" THEN DO
VM/CMS ISIS (Mainframe Version)
REXX EXEC

SAY "****** error in" execname "called from" caller "******"
SAY "****** file" fn ft fm "must start with TR, MF, or IF."
result = 16
RETURN result
END
IF pre = "TR" THEN DO
maxtrfile = 10
number = gen + 1
IF number < 1 | number > maxtrfile THEN DO
SAY "****** error in" execname "called from" caller "******"
SAY "****** The generation must be 0 < gen < maxtrfile", is gen
result = 15
RETURN result
END
fn = pre||cin||gen
*ESTATE fn ft fm
rcstateln = RC
IF rclstateln = 0 THEN DO
SAY "****** error in" execname "called from" caller "******"
SAY "****** file" fn ft fm "not found."
result = rclstateln
RETURN result
END
IF caller == "" THEN DO
SAY "The backup pre||cin||gen has been found."
result = 0
RETURN result
END

/* defaults_supported */

"GLOBALY SELECT #EXEC GET pre||cin
number = gen + 1
IF number < 1 | number > 3 THEN DO
SAY "****** error in" execname "called from" caller "******"
SAY "****** The generation must be 0 < gen < 2, is gen
result = 16
RETURN result
END
VM/CMS ISIS (Mainframe Version)

REXX EXEC

tapes = "VALUE"(pre||cin)
IF tapes = "" THEN DO
SAY "****** error in" execname "called from" caller "******"
SAY "****** There are no tapes available for pre||cin"
result = 16
RETURN result
END
tape. = ""
PARSE VAR tapes "\"" tapes
ltape = 1
DO WHILE tapes ¥ ""
PARSE VAR tapes ltape "\"" tapes
ltape = ltape + 1
END
result = tape. number
IF caller = "" THEN DO
SAY pre||cin||gen "is on tape" result
result = 0
END
RETURN result
VM/CMS ISIS (Mainframe Version)
/*
/* VM/CMS ISIS - SYSTEM INSTALLATION MANUAL 
/*
/*
ADDRESS COMMAND

*FILEDEF = CLEAR
*FILEDEF CARDS DISK IMIT1 DOC = "
*FILEDEF CARDS DISK INSTALL DOC = (APPEND)*
/*
*FILEDEF CARDS DISK IMITOC DOC = (APPEND)*
*FILEDEF CARDS DISK IMINIT DOC = (APPEND)*
SORCLIB(CSVTXBR)
*FILEDEF CARDS DISK IMAGA DOC = (APPEND)*
SORCLIB(COSURGT1)
*FILEDEF CARDS DISK IMIN2 DOC = (APPEND)*
MACLIB(SMAP)
*FILEDEF CARDS DISK IMIN3 DOC = (APPEND)*
MACLIB(UCTRTAB)
*FILEDEF CARDS DISK IMIN4 DOC = (APPEND)*
MACLIB(SRRTTAB)
*FILEDEF CARDS DISK IMIN5 DOC = (APPEND)*
MACLIB(KGTRTAB)
*FILEDEF CARDS DISK IMIN8 DOC = (APPEND)*
MACLIB(KGRTRB3)
*FILEDEF CARDS DISK IMIN7 DOC = (APPEND)*
/*
*FILEDEF CARDS DISK IMAP1 DOC = (APPEND)*
*FILEDEF CARDS DISK IMAP2 DOC = (APPEND)*
*FILEDEF CARDS DISK GENTIS EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAG2 DOC = (APPEND)*
*FILEDEF CARDS DISK ASSEMBLE EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAG2C DOC = (APPEND)*
*FILEDEF CARDS DISK ASSENCOS EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAS7 DOC = (APPEND)*
*FILEDEF CARDS DISK ASSEMB EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAS2 EXEC = (APPEND)*
*FILEDEF CARDS DISK IMASST DOC = (APPEND)*
*FILEDEF CARDS DISK ASSC EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAS6 EXEC = (APPEND)*
*FILEDEF CARDS DISK ASSC EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAS5 EXEC = (APPEND)*
*FILEDEF CARDS DISK ASSC EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAS4 DOC = (APPEND)*
*FILEDEF CARDS DISK IMAS3 EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAS2 EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAS1 EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAS0 EXEC = (APPEND)*
*FILEDEF CARDS DISK COMALL EXEC = (APPEND)*
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS  
REXX EXECs

*FILEDEF CARDS DISK IMAPA DOC = (APPEND)*
*FILEDEF CARDS DISK MODUL EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAPA DOC = (APPEND)*
*FILEDEF CARDS DISK MODALL EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAP1 DOC = (APPEND)*
*FILEDEF CARDS DISK INIT1#2 EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAP2 DOC = (APPEND)*
*FILEDEF CARDS DISK INIT1#2 EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAP3 DOC = (APPEND)*
*FILEDEF CARDS DISK DFMASCAT EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAP4 DOC = (APPEND)*
*FILEDEF CARDS DISK DFMASCAT EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAP5 DOC = (APPEND)*
*FILEDEF CARDS DISK DFTEISIS ASENSRY = (APPEND)*
*FILEDEF CARDS DISK IMAP6 DOC = (APPEND)*
*FILEDEF CARDS DISK DFTEISIS ASENSRY = (APPEND)*
*FILEDEF CARDS DISK IMAP1 DOC = (APPEND)*
*FILEDEF CARDS DISK VMIS1 EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAP2 DOC = (APPEND)*
*FILEDEF CARDS DISK SERVRUN EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAP3 DOC = (APPEND)*
*FILEDEF CARDS DISK SERVEND EXEC = (APPEND)*
*FILEDEF CARDS DISK IMAP4 DOC = (APPEND)*
*FILEDEF CARDS DISK JVCLTAB CARDS = (APPEND)*
/*
*FILEDEF CARDS DISK IMAP8 DOC = (APPEND)*
/*
*EXEC DOUT#2 (ULC=4,ADJ=0,SIDE=6,LL=80,LP=53,XEROX=N)*
util@2*returncode = RC
IF util@2returncode <> 0 THEN DO
   EXIT util@2returncode
END

*FILEDEF = CLEAR
RETURN util@2returncode

VM/CMS ISIS (Mainframe Version)
Mainframe VM/CMS ISIS

REXX EXEC

/* ISIS/CMS - RECOVER EXTRAPARTITION DATA SETS */
/* ISIS/CMS - HANDLE PARAMETERS FOR CDIPL EXEC */

/* cat = vsm catalog name */
/* catowner = vsm catalog unit user id */
/* catunit = vsm catalog unit */
/* catunitpw = vsm catalog unit password */
/* dbcedps = dcb for extra partition data set */
/* dcsfls80 = dcb for cards */
/* func = function */
/* fx = prefix of data sets */
/* sortlib = sort txtlib */
/* sys = prefix of system files */
/* txtlib = isis txtlib */

ADDRESS COMMAND

ARC parameters

PARSE SOURCE . execname .

PARSE VAR parameters operands "f" options

"EXEC PROLOG" execname "f" options

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults supported */

"GLOBAL SELECT *EXEC GET PROLOG1"
"GLOBAL SELECT *EXEC GET PROLOG2"
"GLOBAL SELECT *EXEC GET PROLOG3"
"GLOBAL SELECT *EXEC GET PROLOG1"
"GLOBAL SELECT *EXEC GET PROLOG2"

nexecedef = "VALUE"(PROLOG*1) "VALUE"(PROLOG*2) "VALUE"(PROLOG*3)

pexecedef = "VALUE"(PROLOG*1) "VALUE"(PROLOG*2) "VALUE"(PROLOG*3)

PARSE VAR nexecedef "f" nexecedef
PARSE VAR pexecedef "f" pexecedef

DO WHILE nexecedef = "f" nexecedef
    PARSE VAR nexecedef name "f" nexecedef
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

PARSE VAR pexecedef parms "f" pexecedef
INTERPRET name = "parse"

END

fn. = ""
ft. = ""
fs. = ""

PARSE VAR operands fn.1 operands
IF fn.1 = "f" THEN fn.1 = "YSPP89"
IF LENGTH(fn.1) > 8 THEN DO
    SAY "****** error in" execname " ******"
    SAY "****** filename" fn.1 "has more than 8 characters."
    RETURN 16
END

ft.1 = "CARDS"
fs.1 = "s"

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib

"EXEC CIME" catowner catunit "s" catunitpw "IQ MR STACK MODE -P"

gimereturncode = RC
PULL filename .
IF gimereturncode > 4 THEN RETURN gimereturncode

"ESTATE" sys "INPO1 A"
imp@1state = RC

IF imp@1state = 0 THEN DO
    "FILEDEF = CLEAR"
    "FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
    "FILEDEF SYSPRINT DISK UTIL@1 LISTING A"
    "FILEDEF TPINP DISK" sys "INPO1 A"
    "FILEDEF TPOUT DUMMY"
    "FILEDEF TPSAVE DISK" sys "IPLO1 A ("dcbeds" DISP MOD"
    "FILEDEF CARDS DUMMY"
    "FILEDEF JCLTAB DUMMY"
    stack = "TPRUN\"-P\;
    PUSH stack
    "EXEC @U UTIL@4"
util@4returncode = RC
VM/CMS ISIS (Mainframe Version)
IF util4returncode -> 0 THEN DO
    RETURN util4returncode
END

"ERASE" sys "IN01 A"

END

"ESTATE" sys "IN02 A"
inp2state = RC

IF inp2state = 0 THEN DO
    "FILEDEF = CLEAR"
    "FILEDEF PLDUMP DISK PLDUMP LISTING A"
    "FILEDEF SYSPRINT DISK UTIL04R2 LISTING A"
    "FILEDEF TPINP DISK" sys "IN02 A"
    "FILEDEF TPOUT DUMMY"
    "FILEDEF TPSAVE DISK" sys "1PL02 A (dcbepds) DISP MOD"
    "FILEDEF CARDS DUMMY"
    "FILEDEF JCLTAB DUMMY"

    stack = "TVPRUN+R":"
    PUSH stack

    "EXECOS UTIL04"

    util4returncode = RC
    IF util4returncode -> 0 THEN DO
        RETURN util4returncode
    END

    "ERASE" sys "IN02 A"

END

"DLBL IJYSCT" filemode "DSN" cat
"DLBL BOOK" filemode "DSN" pfx "BOOK MASTER VSAM CAT IJYSCT"
"DLBL PERI" filemode "DSN" pfx "PERI MASTER VSAM CAT IJYSCT"
"DLBL PREP" filemode "DSN" pfx "PREP MASTER VSAM CAT IJYSCT"

"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK VSP08 LISTING A"
"FILEDEF CARDS DISK" fn.1 ft.1 fm.1
"FILEDEF PLDUMP DISK PLDUMP LISTING A"
"FILEDEF LOGMSG DISK VSP08 LOGMSG A"
"FILEDEF M8KUP DUMMY"

VM/CMS ISIS (Mainframe Version)

---

Mainframe VM/CMS ISIS

STACK = "CHKL":"
push stack

"EXECOS VSP08"

vsp08returncode = RC
IF vsp08returncode -> 0 THEN DO
    RETURN vsp08returncode
END

"SET CMSTYPE HT"
IF gisernreturncode = 0 THEN "EXEC DROP" filemode "10"
"FILEDEF = CLEAR"
"DLBL = CLEAR"
"SET CMSTYPE RT"

RETURN vsp08returncode

VM/CMS ISIS (Mainframe Version)
/* ISIS/CMS - PROCESS EXTRAPARTITION DATA SETS */
/* ISIS/CMS - HANDLE PARAMETERS FOR COPEP EXEC */
/* catunipsw = vsam catalog unit password */
/* dcbepds = dcb for extra partition data set */
/* dcbftp80 = dcb for cards */
/* sortlib = sort txtlib */
/* sortpgm = name of sort program */
/* sys = prefix of system files */
/* txtlib = isis txtlib */

ADDRESS COMMAND
ARG parameters

PARSE SOURCE . execname .

PARSE VAR parameters operands "(" options

*EXEC PROLOG* execname "("options

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

*/ defaults_supported */

"GLOBAL SELECT EXEC GET PROLOG1"
"GLOBAL SELECT EXEC GET PROLOG2"
"GLOBAL SELECT EXEC GET PROLOG3"
"GLOBAL SELECT EXEC GET PROLOG4"
"GLOBAL SELECT EXEC GET PROLOG5"

nexecedef = "VALUE"(PROLOG"N1") || "VALUE"(PROLOG"N2") || "VALUE"(PROLOG"N3")
pexecedef = "VALUE"(PROLOG"P1") || "VALUE"(PROLOG"P2") || "VALUE"(PROLOG"P3")

PARSE VAR nexecedef "\" nexecedef
PARSE VAR pexecedef "\" pexecedef

DO WHILE nexecedef = ""
   PARSE VAR nexecedef name "\" nexecedef
   PARSE VAR pexecedef para "\" pexecedef
   INTERPRET name = "para"
END

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

fn. = ""
ft. = ""
fn. = ""

"GLOBAL TXTLIB PLLIB CMLIB" txtlib sortlib

"ESTATE ONLINEU EXEC A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE ONLINEU EXEC A"

"ESTATE ONLINE EXEC A"
rctexted = RC
IF rctexted = 0 THEN "ERASE ONLINED EXEC A"

"ESTATE* sys *INP01 A"
inptstate = RC

IF inptstate = 0 THEN DO
   "ESTATE* sys *DUMMY A"
   listingstate = RC
   IF listingstate = 0 THEN "ERASE* sys *DUMMY A"
   "FILEDEF SYSOUT DISK "sortpgm" LISTING A"
   sortpgm sys *INP01 A sys *DUMMY A COSEPOS SORT *" asortreturncode = RC
   IF asortreturncode = 0 THEN DO
      RETURN asortreturncode
   END
   "ERASE* sys *INP01 A"
   "ESTATE* sortpgm LISTING A"
   listingstate = RC
   IF listingstate = 0 THEN "ERASE* sortpgm LISTING A"
   "FILEDEF * CLEAR*
   "FILEDEF PLDUMP DISK PLDUMP LISTING A"
   "FILEDEF SYSPRINT DISK UTIL801 LISTING A"
   "FILEDEF TPNP DISK" sys *DUMMY A"
   "FILEDEF TPRNT DUMMY"
   "FILEDEF TPSAVE DISK" sys *1PL801 A (*dcbepds "DISP MOD"
   "FILEDEF CARDS DUMMY"
   "FILEDEF JCLTAB DUMMY"

VM/CMS ISIS (Mainframe Version)
stack = "TYPRUN=R";

**EXECOS UTL04**

utl04returncode = RC
IF utl04returncode = 0 THEN DO
RETURN utl04returncode
END

"ERASE" sys "DUMMY A"

"ESTATE UTL04R1 LISTING A"
sysprintstate = RC
IF sysprintstate = 0 THEN "ERASE UTL04R1 LISTING A"

"ESTATE" sys "PROOF A"
proofstate = RC

"FILEDEF = CLEAR"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF SYSPRINT DISK UTL04U1 LISTING A"
"FILEDEF TPINP DISK" sys "IPL01 A"
IF proofstate = 0 THEN DO
"FILEDEF TPINP DISK" sys "PROOF A (APPEND"
END
"FILEDEF TPUT DUMMY"
"FILEDEF TPSAVE DUMMY"
"FILEDEF CARDS DISK ONLINEU EXEC A ("dcbfs08"
"FILEDEF JCLTAB DISK JCLTAB CARDS *"

stack = "TYPRUN=-U";

**EXECOS UTL04**

utl04returncode = RC
IF utl04returncode = 0 THEN DO
RETURN utl04returncode
END

"ESTATE UTL04U1 LISTING A"
sysprintstate = RC
IF sysprintstate = 0 THEN "ERASE UTL04U1 LISTING A"

"ESTATE" sys "INPUT A"
inputstate = RC

VM/CMS ISIS (Mainframe Version)

---

Mainframe VM/CMS ISIS

REXX EXECS

IF inputstate = 0 THEN "ERASE" sys "INPUT A"

"FILEDEF = CLEAR"
"FILEDEF SYSDUT DISK "sortpgm" LISTING A"

IF proofstate = 0 THEN DO
sortpgm sys "IPL01 A" sys "PROOF A" sys "INPUT A CDCTERM SORT *"
END
ELSE DO
sortpgm sys "IPL01 A" sys "INPUT A CDCTERM SORT *"
END

ssortreturncode = RC
IF ssortreturncode = 0 THEN DO
RETURN ssortreturncode
END

"ESTATE "sortpgm" LISTING A"
liststate = RC
IF liststate = 0 THEN "ERASE "sortpgm" LISTING A"

"ESTATE" sys "IPL01 A"
ipi01state = RC
IF pipi01state = 0 THEN "ERASE" sys "IPL01 A"

IF proofstate = 0 THEN DO
"ERASE" sys "PROOF A"
END

END

"ESTATE" sys "INP02 A"
inp02state = RC

IF inp02state = 0 THEN DO
"FILEDEF = CLEAR"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF SYSPRINT DISK UTL04R2 LISTING A"
"FILEDEF TPINP DISK" sys "INP02 A"
"FILEDEF TPUT DUMMY"
"FILEDEF TPSAVE DISK" sys "IPL02 A (*dcbps08 "DISP MOD"
"FILEDEF CARDS DUMMY"
"FILEDEF JCLTAB DUMMY"

stack = "TYPRUN= R";

**EXECOS UTL04**

utl04returncode = RC
IF utl04returncode = 0 THEN DO
RETURN utl04returncode
END

"PROOF A (APPEND"
END
"PROOF A"

VM/CMS ISIS (Mainframe Version)
"EXECOS UTL04"
util@returncode = RC
IF util@returncode = 0 THEN DO
  RETURN util@returncode
END
"ERASE *sys "INF02 A"
"ESTATE UTL04R2 LISTING A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE UTL04R2 LISTING A"
"ESTATE" *sys "QUERY A"
querystate = RC
"FILEDEF = CLEAR"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF SYSRPRINT DISK UTL04U2 LISTING A"
"FILEDEF TPINP DISK" *sys "IPL02 A"
IF querystate = 0 THEN DO
  "FILEDEF TPINP DISK" *sys "QUERY A (APPEND)"
END
"FILEDEF TPUT DUMMY"
"FILEDEF TPSAVE DISK" *sys "QUERY A (DISP MOD)"
"FILEDEF CARDS DISK ONLINED EXEC A ("dcbfb80"
"FILEDEF JCLTAB DISK JCLTAB CARDS *"
stack = "IYPRUN="'U'";
PUSH stack
"EXECOS UTL04"
util@returncode = RC
IF util@returncode = 0 THEN DO
  RETURN util@returncode
END
"ESTATE UTL04U2 LISTING A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE UTL04U2 LISTING A"
"ESTATE" *sys "IPL02 A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE" *sys "IPL02 A"

VM/CMS ISIS (Mainframe Version)

FILEDEF = CLEAR"
END
"ESTATE ONLINEU EXEC A"
rcsaved = RC
IF rcsaved = 0 THEN DO
  "EXEC ONLINEU" catunitpu
END
"SET CMSTYPE HT"
"FILEDEF = CLEAR"
"SET CMSTYPE RT"
RETURN 0

VM/CMS ISIS (Mainframe Version)
ADDRESS COMMAND

"FILEDEF = CLEAR"
"FILEDEF CARDS DISK PHO7T DOC * (APPEND)*
"FILEDEF CARDS DISK PHO7T DOC * (APPEND)*
"FILEDEF CARDS DISK GSND DOC * (APPEND)*
"FILEDEF CARDS DISK CENTE DOC * (APPEND)*
"FILEDEF CARDS DISK CREN DOC * (APPEND)*
"FILEDEF CARDS DISK COMU DOC * (APPEND)*
"FILEDEF CARDS DISK DEFCO DOC * (APPEND)*
"FILEDEF CARDS DISK DEPSC DOC * (APPEND)*
"FILEDEF CARDS DISK DOPT DOC * (APPEND)*
"FILEDEF CARDS DISK ESND DOC * (APPEND)*
"FILEDEF CARDS DISK FLMA DOC * (APPEND)*
"FILEDEF CARDS DISK FORC DOC * (APPEND)*
"FILEDEF CARDS DISK HEACU DOC * (APPEND)*
"FILEDEF CARDS DISK INDEX DOC * (APPEND)*
"FILEDEF CARDS DISK INHSP DOC * (APPEND)*
"FILEDEF CARDS DISK LOOKU DOC * (APPEND)*
"FILEDEF CARDS DISK MARGI DOC * (APPEND)*
"FILEDEF CARDS DISK MOOR DOC * (APPEND)*
"FILEDEF CARDS DISK NWCQ DOC * (APPEND)*
"FILEDEF CARDS DISK NEH1 DOC * (APPEND)*
"FILEDEF CARDS DISK NWP1A DOC * (APPEND)*
"FILEDEF CARDS DISK NOFAC DOC * (APPEND)*
"FILEDEF CARDS DISK NOTC DOC * (APPEND)*
"FILEDEF CARDS DISK POI7T DOC * (APPEND)*
"FILEDEF CARDS DISK PSIZE DOC * (APPEND)*
"FILEDEF CARDS DISK PUT DOC * (APPEND)*
"FILEDEF CARDS DISK QUAD2 DOC * (APPEND)*
"FILEDEF CARDS DISK REMPO DOC * (APPEND)*
"FILEDEF CARDS DISK REMP1A DOC * (APPEND)*
"FILEDEF CARDS DISK REPSA DOC * (APPEND)*
"FILEDEF CARDS DISK SEND DOC * (APPEND)*
"FILEDEF CARDS DISK SKPI DOC * (APPEND)*
"FILEDEF CARDS DISK TAB DOC * (APPEND)*
"FILEDEF CARDS DISK TABC DOC * (APPEND)*
"FILEDEF CARDS DISK TCHN DOC * (APPEND)*
"FILEDEF CARDS DISK TFACU DOC * (APPEND)*
"FILEDEF CARDS DISK TRS7 DOC * (APPEND)*
"FILEDEF CARDS DISK PHC08 DOC * (APPEND)*
"FILEDEF CARDS DISK PHC02 DOC * (APPEND)*

Mainframe VM/CMS ISIS

"FILEDEF CARDS DISK PHC03 DOC * (APPEND)*
"FILEDEF CARDS DISK PHC04 DOC * (APPEND)*
"FILEDEF CARDS DISK PHN1V DOC * (APPEND)*
"FILEDEF CARDS DISK PHAP5 DOC * (APPEND)*
"FILEDEF CARDS DISK PHAP6 DOC * (APPEND)*
"FILEDEF CARDS DISK PHAP7 DOC * (APPEND)*

*EXEC COUTL2 UFC=L, ADJ=#, SIDE=5, LL=88, LP=63, XERDX=N*
util2returncode = RC
IF util2returncode = 0 THEN DO
EXIT util2returncode
END

"FILEDEF = CLEAR"
RETURN util2returncode

VM/CMS ISIS (Mainframe Version)
/* ISIS/CMS - PUT A MEMBER OF A GENERATION DATA GROUP
*/
/* ISIS/CMS - HANDLE PARAMETERS FOR CPUDG EXEC */

ADDRESS COMMAND
ARG parameters
PARS VAR parameters operands "(" parameters
PARS VAR operands caller operands
PARS VAR parameters fn fm.

IF LENGTH(fn) == 8 & LENGTH(fm) == 6 THEN DO
  SAY "******** error in" execname "called from" caller "********"
  SAY "******** The filename of the BACKUP file must",
  "be 6 or 8 characters long."
  RESULT = 16
  RETURN result
END

FM = "BACKUP"
IF fm == "" THEN fm = "A"
IF LENGTH(fm) == 1 THEN DO
  SAY "******** error in" execname "called from" caller "********"
  SAY "******** The filemode of the BACKUP file must be 1 character."
  RETURN 16
END

pre = SUBSTR(fn,1,2)
cin = SUBSTR(fn,3,4)
gen = SUBSTR(fn,7,2)

IF DATATYPE(gen) == "NUM" THEN DO
  IF gen == "" THEN DO
    SAY "******** error in" execname "called from" caller "********"
    SAY "******** The generation must be numeric."
    RESULT = 16
    RETURN result
  END
ELSE DO
  number = 1
END

VM/CMS ISIS (Mainframe Version)

END ELSE DO
number = gen + 1
END

IF pre == "TR" & pre == "MF" & pre == "IF" THEN DO
  SAY "******** error in" execname "called from" caller "********"
  SAY "******** file" fn ft fm "must start with TR, MF, or IF."
  RESULT = 16
  RETURN result
END

fn = ""

IF pre == "TR" THEN DO
  maxtrfile = 18
  IF number < 1 | number > maxtrfile THEN DO
    SAY "******** error in" execname "called from" caller "********"
    SAY "******** The generation must be: 0 < gen < maxtrfile", is" gen
    RESULT = 16
    RETURN result
  END
ELSE DO
  IF gen == "" THEN DO
    IF caller == "" THEN DO
      SAY "New BACKUP of TRILE not allowed. No action."
      RETURN 0
    END
    DO index = 1 TO maxtrfile
      fn.index = pre||cin||RIGHT(index-1,2,"0")
      rcsatexfn.index = 0
    END
    DO index = 1 TO maxtrfile
      ESTATE fn.index ft fm
      rcsatexfn.index = RC
      IF rcsatexfn.index == 0 THEN DO
        curtrfile = index - 1
        LEAVE index
      END
    END
    IF SYMBOL("curtrfile") == "LIT" THEN DO
      ERASE fn.maxtrfile ft fm
      curtrfile = maxtrfile - 1
    END
  END
VM/CMS ISIS (Mainframe Version)
DO index = currfile BY -1 TO 1
    next = index + 1
    "RENAME" fn.index ft fs fn.next ft fs
    if rrenamefn.index = 0 THEN DO
        SAY "****** error in" execname "called from" caller "******"
        SAY "****** error in RENAME" fn.index ft fs fn.next ft fs
        result = 16
        RETURN result
    END
    END
    result = pre||cin||"BB"
    RETURN result
ELSE DO
    fn.1 = pre||cin||gen
    "ESTATE" fn.1 ft fm
    rstatefn1 = RC
    result = fn.1
    IF rstatefn1 = 0 THEN DO
        IF caller = "" THEN DO
            SAY fn.1 ft fm "does not exist."
            RETURN 0
        END
        RETURN result
    END
    ELSE DO
        IF caller = "" THEN DO
            SAY fn.1 ft fm "exists."
            RETURN 0
        END
        "ERASE" fn.1 ft fm
        RETURN result
    END
END

/* defaults_supported */
"GLOBAL SELECT $EXEC GET" pre||cin

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

IF number < 1 | number > 3 THEN DO
    SAY "****** error in" execname "called from" caller "******"
    SAY "****** The generation must be: 0 < gen < 2, ia" gen
    result = 16
    RETURN result
END
tapes = "VALUE"(pre||cin)
IF tapes = "" THEN DO
    SAY "****** error in" execname "called from" caller "******"
    SAY "****** There are no tapes available for" pre||cin
    result = 16
    RETURN result
END
tape = ""
PARSE VAR tapes "\"" tapes
   ITape = 1
   DO WHILE tapes = ""
       PARSE VAR tapes tape.1tape "\"" tapes
       tape.1tape = STRIP(tape.1tape)
       ITape = ITape + 1
   END
   IF gen = "" THEN DO
       tapes = "\"tape.3\"tape.1\"tape.2\"
   "GLOBAL SELECT $EXEC SETLP" pre||cin tapes
       result = tape.3
   IF caller = "" THEN DO
       SAY pre||cin||"BB is on tape" tape.3
       SAY pre||cin||"B1 is on tape" tape.1
       SAY pre||cin||"B2 is on tape" tape.2
       RETURN 0
   END
   RETURN result
END
IF gen = "" THEN DO
    result = tape.number
    IF caller = "" THEN DO
        SAY pre||cin||gen "is on tape" result
        RETURN 0
    END

VM/CMS ISIS (Mainframe Version)
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

RETURN result
END
RETURN result

 VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

/* ISIS/CMS - BACKUP AND LOG EXTRAPARTITION DATA SETS */
/* ISIS/CMS - HANDLE PARAMETERS FOR CDRCVREP EXEC */
/* dbn = data base name */
/* dcbsdps = dcb for extra partition data set */
/* sortlib = sort txtlib */
/* egs = prefix of system files */
/* txtlib = isis txtlib */

ADDRESS COMMAND

ARG parameters
PARSE SOURCE . . execname .
PARSE VAR parameters operands "i" options
"EXEC PROLOG" execname "(DBN=x" options

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults supported */
"GLOBALV SELECT EXEC GET PROLOG1"
"GLOBALV SELECT EXEC GET PROLOG2"
"GLOBALV SELECT EXEC GET PROLOG3"
"GLOBALV SELECT EXEC GET PROLOG4"
"GLOBALV SELECT EXEC GET PROLOG5"

nexecdef = "VALUE"(PROLOG"N1")|| "VALUE"(PROLOG"N2")|| "VALUE"(PROLOG"N3")

nexecdef = "VALUE"(PROLOG"P1")|| "VALUE"(PROLOG"P2")|| "VALUE"(PROLOG"P3")

PARSE VAR nexecdef pars "\" nexecdef
PARSE VAR nexexecf pars "\" nexexecf

DO WHILE nexexecf = ""
PARSE VAR nexexecf name pars "\" nexexecf
INTERPRET name = " pars"
END
"GLOBAL TXTLIB PLILIB CMSLIB* txtlib sortlib
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

/*
 */

ADDRESS COMMAND

"FILEDEF = CLEAR"
"FILEDEF CARDS DISK RMT11 DOC *"
"FILEDEF CARDS DISK RMT12 DOC (APPEND)"
"FILEDEF CARDS DISK RMT13 DOC (APPEND)"
"FILEDEF CARDS DISK RMT14 DOC (APPEND)"
"FILEDEF CARDS DISK RMT15 DOC (APPEND)"
"FILEDEF CARDS DISK RMT16 DOC (APPEND)"
"FILEDEF CARDS DISK RMT17 DOC (APPEND)"
"FILEDEF CARDS DISK RMT18 DOC (APPEND)"
"FILEDEF CARDS DISK RMT19 DOC (APPEND)"
"FILEDEF CARDS DISK RMT20 DOC (APPEND)"
"FILEDEF CARDS DISK RMT21 DOC (APPEND)"
"FILEDEF CARDS DISK RMT22 DOC (APPEND)"
"FILEDEF CARDS DISK RMT23 DOC (APPEND)"
"FILEDEF CARDS DISK RMT24 DOC (APPEND)"
"FILEDEF CARDS DISK RMT25 DOC (APPEND)"
"FILEDEF CARDS DISK RMT26 DOC (APPEND)"
"FILEDEF CARDS DISK RMT27 DOC (APPEND)"
"FILEDEF CARDS DISK RMT28 DOC (APPEND)"
"FILEDEF CARDS DISK RMT29 DOC (APPEND)"
"FILEDEF CARDS DISK RMT30 DOC (APPEND)"
"FILEDEF CARDS DISK RMT31 DOC (APPEND)"
"FILEDEF CARDS DISK RMT32 DOC (APPEND)"
"FILEDEF CARDS DISK RMT33 DOC (APPEND)"
"FILEDEF CARDS DISK RMT34 DOC (APPEND)"
"FILEDEF CARDS DISK RMT35 DOC (APPEND)"

VM/CMS ISIS (Mainframe Version)
"FILEDEF CARDS DISK RMP38 DDC * (APPEND)"
"FILEDEF CARDS DISK RMFL DDC * (APPEND)"
"FILEDEF CARDS DISK RMM1 DDC * (APPEND)"
"FILEDEF CARDS DISK PLICNF DDC * (APPEND)"
"FILEDEF CARDS DISK RMM2 DDC * (APPEND)"
"FILEDEF CARDS DISK VSMALL DDC * (APPEND)"
"FILEDEF CARDS DISK SMALL DDC * (APPEND)"

"EXEC COUTL02 (ULC=L, ADJ=8, SIDE=S, LL=88, LP=53, XEROX=N"

util02returncode = RC
IF util02returncode = 0 THEN DO
EXIT util02returncode
END

/
* THE OUTPUT BEING TOO LARGE, XEDIT WILL FAIL WITH
CMSX032S FILE 'UTL02 LISTING A1' TOO LARGE
THE SIZE OF THE VIRTUAL MACHINE HAS TO BE INCREASED
TO, SAY, 6M. PROCEED AS FOLLOWS:
* QUERY STORAGE (TO SEE WHAT YOU GOT)
* DEFINE STOR GN (INCREASE THE SIZE)
* IPL CMS (NEW SIZE WILL BE EFFECTIVE)
DO NOT FORGET TO RESET THE SIZE AFTERWARDS TO WHAT IT WAS BEFORE ! */
"FILEDEF = CLEAR"
RETURN util02returncode

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

/*
/* CMS/ISIS - PRINT THESAURUS SUBSYSTEM MANUAL (RELEASE 4.6) */
/*
ADDRESS COMMAND
"FILEDEF = CLEAR"
"FILEDEF CARDS DISK THES DDC *"
"EXEC COUTL02 (ULC=L, ADJ=8, SIDE=S, LL=88, LP=53, XEROX=N"

util02returncode = RC
IF util02returncode = 0 THEN DO
EXIT util02returncode
END
"FILEDEF = CLEAR"
RETURN util02returncode

VM/CMS ISIS (Mainframe Version)
ADDRESS COMMAND

"FILEDEF = CLEAR"
"FILEDEF CARDS DISK TPTIT DOC =" (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)

*EXEC COUTL01 (ULC=+, ADC=0, SIDE=S, LL=88, LP=63, XEROX=N)

util2returncode = RC
IF util2returncode ≠ 0 THEN DO
EXIT util2returncode
END

"FILEDEF = CLEAR"
RETURN util2returncode

### VM/CMS ISIS (Mainframe Version) ###

```

// Mainframe VM/CMS ISIS
// COTLIST 43

/\**
/\ CDS/ISIS - PRINT TERMINAL OPERATOR MANUAL (RELEASE 4.6)
/\*
/\*
/\*

ADDRESS COMMAND

"FILEDEF = CLEAR"
"FILEDEF CARDS DISK TPTIT DOC =" (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)
"FILEDEF CARDS DISK TPDOC DOC = (APPEND)

*EXEC COUTL01 (ULC=+, ADC=0, SIDE=S, LL=88, LP=63, XEROX=N)

util2returncode = RC
IF util2returncode ≠ 0 THEN DO
EXIT util2returncode
END

"FILEDEF = CLEAR"
RETURN util2returncode

```

```

// Mainframe VM/CMS ISIS
// COUTL01 44

/\* ISIS/CMZ - UTIL8 (LOAD FIELD SELECT TABLES AND PRINT FORMATS) *
/\*
/\*
/\*

ADDRESS COMMAND

ARG parameters
PARSE SOURCE . . execname .
PARSE VAR parameters operands "(" options
*EXEC PROLOG* execname "(*CLN=" options
prologreturncode = RC
IF prologreturncode ≠ 0 THEN RETURN prologreturncode

/\* defaults_supported *\/
"GLOBALV SELECT EXEC GET PROLOGM1"
"GLOBALV SELECT EXEC GET PROLOGM2"
"GLOBALV SELECT EXEC GET PROLOGM3"
"GLOBALV SELECT EXEC GET PROLOGP1"
"GLOBALV SELECT EXEC GET PROLOGP2"
"GLOBALV SELECT EXEC GET PROLOGP3"

nexecdef = "VALUE"(PROLOGM*N1") "VALUE"(PROLOGM*N2") "VALUE"(PROLOGM*N3")
pxexecdef = "VALUE"(PROLOGP*P1") "VALUE"(PROLOGP*P2") "VALUE"(PROLOGP*P3")

PARSE VAR nexecdef "\" nexecdef
PARSE VAR pxexecdef "\" pxexecdef

DO WHILE nexecdef ≠ ""
   PARSE VAR nexecdef name "\" nexecdef
   PARSE VAR pxexecdef para "\" pxexecdef
   INTERPRET name = "para"
END

fn. = ""
ft. = ""
fn. = ""

VM/CMS ISIS (Mainframe Version)
```
PARSE VAR operands fn.2 operands

fn.1 = dbn
ft.1 = "FSI"
fs.1 = "s"

fn.2 = dbn
ft.2 = "PARM"

IF fn.2 = "" THEN DO
  SAY "****** error in" execname " ******"
  SAY "****** filemode of parafile is missing, is mandatory." 
  RETURN 16
END

IF LENGTH(fn.2) > 1 THEN DO
  SAY "****** error in" execname " ******"
  SAY "****** filemode" fn.2 "has more than 1 character."
  RETURN 16
END

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib

"ESTATE" fn.1 ft.1 fm.1
rcsaved = RC
IF rcsaved = 0 THEN DO
  SAY "****** error in" execname " ******"
  SAY "****** file" fn.1 ft.1 fm.1 "not found."
  RETURN rcsaved
END

"ESTATE" fn.2 ft.2 fm.2
rcsaved = RC
IF rcsaved = 0 THEN "ERASE" fn.2 ft.2 fm.2

"FILEDEF = CLEAR"
"FILEDEF SORTPOS DISK" fn.2 ft.2 fm.2 "l"dcbvb4028
"FILEDEF CARDS DISK" fn.1 ft.1 fm.1
"FILEDEF PRINTER DISK UTIL01 LISTING A"

"EXECOS UTL01"
util01returncode = RC
IF util01returncode = 0 THEN DO
  RETURN util01returncode
END

VM/CMS ISIS (Mainframe Version)
ADDRESS COMMAND

ARG parameters

PARSE SOURCE execname

PARSE VAR parameters operands "(" options

"EXEC PROLOG execname "LP=63" options

prologreturncode = RC
IF prologreturncode ~ 0 THEN RETURN prologreturncode

"GLOBAL SELECT exec GET PROLOG1"
"GLOBAL SELECT exec GET PROLOG2"
"GLOBAL SELECT exec GET PROLOGN3"
"GLOBAL SELECT exec GET PROLOG1"
"GLOBAL SELECT exec GET PROLOGP2"
"GLOBAL SELECT exec GET PROLOGP3"

nexecdef = "VALUE" *(PROLOG*N1) | "VALUE"*(PROLOG*N2) | "VALUE"*(PROLOG*N3)

pxexecdef = "VALUE" *(PROLOG*P1) | "VALUE"*(PROLOG*P2) | "VALUE"*(PROLOG*P3)

PARSE VAR nexecdef "\" nexecdef
PARSE VAR pxexecdef "\" pxexecdef

DO WHILE nexecdef ~ ""
PARSE VAR nexecdef name "\" nexecdef
PARSE VAR pxexecdef parm "\" pxexecdef
INTERPRET name = \" parm"

VM/CMSISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

END

fn. = ""
ft. = ""
fa. = ""

PARSE VAR operands fn.1 operands
IF fn.1 ~ "" THEN DO
IF LENGTH(fn.1) > 8 THEN DO
SAY "\"filename fn.1 has more than 8 characters."
RETURN 16
END
ft.1 = "CARDS"
fn.1 = "\" END

"GLOBAL TXTLIB PLLIB CMSLIB" txtlib sortlib

"FILEDEF SYSPRINT DISK UTIL02 SYSPRINT A"
"FILEDEF PRINTER DISK UTIL02 LISTING A ("dcblist

/* THIS FILEDEF IS ONLY EXECUTED IF THE FILENAME IS EXPLICITLY
MENTIONED IN THE CALL. OTHERWISE THE FILEDEF IS SUPPOSED TO
BE IN THE CALLING PROGRAM

IF fn.1 ~ "" THEN DO
FILEDEF CARDS DISK* fn.1 ft.1 fn.1 END

stack = "UC="*uc"",LL="*ll",LP="*lp",SIDE="*side",ADJ="*adj",XEROX="*xerox";"
push stack

"EXECDS UTIL02"
util02returncode = RC
IF util02returncode ~ 0 THEN DO
RETURN util02returncode
END

"ESTATE UTIL02 SYSPRINT A"
rcsaved = RC
IF rsaved ~ 0 THEN "ERASE UTIL02 SYSPRINT A"

"FILEDEF = CLEAR"

VM/CMSISIS (Mainframe Version)
RETURN util@2returncode

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS  

REXX EXECs  

CDVSLST  $S$

/*
/* CDS/ISIS - SWITCHING TO VSAM (RELEASE 4.6)
/*
*/

ADDRESS COMMAND

"FILEDEF * CLEAR"
"FILEDEF CARDS DISK 40 DOC *"
"EXEC CDUTL@2 ULC=L,ADJ=0,SIDE=5,LL=80,LP=63,XEROX=N"
util@2returncode = RC
IF util@2returncode = 0 THEN DO
EXIT util@2returncode
END
"FILEDEF * CLEAR"
RETURN util@2returncode
/* ISIS/CMS - INVERTED FILE BACKUP UTILITY (DISK VERSION) */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVBKUPIF EXEC */
#/ cat = vsam catalog name
#/ catorner = vsam catalog unit user id
#/ catunit = vsam catalog unit
#/ cinn = cluster name
#/ dbcbvttape = dbcb for bkup tape
#/ func = function (here dump)
#/ igen = log generation
#/ pf = prefix of data sets
#/ sortlib = sort txtlib
#/ txtlib = isis txtlib

ADDRESS COMMAND

ARG parameters

PARSE SOURCE .. execname .

PARSE VAR parameters operands "*" options

"EXEC PROLOG" execname "<DBN>", "options "FUNC=BKUP"

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */

"GLOBAL SELECT "EXEC GET PROLOG1""
"GLOBAL SELECT "EXEC GET PROLOG2""
"GLOBAL SELECT "EXEC GET PROLOG3""
"GLOBAL SELECT "EXEC GET PROLOG4""

nextcmd = "VALUE"(PROLOG*N1)) | "VALUE"(PROLOG*N2) | "VALUE"(PROLOG*N3) | "VALUE"(PROLOG*N4)

pexecode = "VALUE"(PROLOG*P1) | "VALUE"(PROLOG*P2) | "VALUE"(PROLOG*P3) | "VALUE"(PROLOG*P4)

PARSE VAR nextcmd = "" nextcmd
PARSE VAR pexecode = "" pexecode

DO WHILE nextcmd = ""

PARSE VAR nextcmd name = "" nextcmd

VM/CMS ISIS (Mainframe Version)

INTERPRET name = "para"
END
fn. = ""
ft. = ""
fs. = ""

PARSE VAR operands fm.1 operands
IF fm.1 = "" THEN fm.1 = "A"
IF LENGTH(fm.1) > 1 THEN DO
SAY "******** error in" execname "********
SAY "******** filename" fm.1 "has more than 1 character."
RETURN 16
END
ft. = "BACKUP"
fn. = "IF"|cin|"80"

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib

"EXEC GIVE" catowner catunit "* G STRING MODE -P"
giserreturncode = RC
FULL filemode .
IF giserreturncode > 4 THEN RETURN giserreturncode

"DBLB JYSYCT" filemode "DSN" cat

"DBLB INVF" filemode "DSN" pf x cin "INVF (YSAM CAT JYSYCT)"
"DBLB IFLOG" filemode "DSN" pf x cin "IFLOG (YSAM CAT JYSYCT)"

"FILEDEF = CLEAR"
IF fn.1 = "" THEN "FILEDEF IFBKUP DISK" fn.1 ft.1 fm.1 "(dbcbvttape"
ELSE "FILEDEF IFBKUP DUMMY"

suffix = SUBSTR(func,1,1)|SUBSTR(cln,1,2)

"FILEDEF IFLIST DISK VSP35 LISTING A"
"FILEDEF SYSPRINT DISK VSP35"suffix "LISTING A"
"FILEDEF LOGMSG DISK VSP35"suffix "LOGMSG A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"

stack = "FUNC","func",CLN="cin",GEN="igen"
push stack

"EXECOS VSP35"

vsp35returncode = RC
IF vsp35returncode > 4 THEN DO

VM/CMS ISIS (Mainframe Version)
RETURN vsp35returncode
END
"SET CMSTYPE HT"
IF gimerreturncode = 0 THEN "EXEC DROP" filename "Q"
"FILEDEF CLEAR"
"DLBL CLEAR"
"SET CMSTYPE RT"
RETURN vsp35returncode

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS
/*
** ARG parameters
** PARSE SOURCE . . execname .
** PARSE VAR parameters operands "if" options
** "EXEC PROLOG" execname "::DBN" . . options "FUNC=BKUP"
** prologreturncode = RC
** IF prologreturncode = 0 THEN RETURN prologreturncode
/*
** defaults_supported */
"GLOBAL SELECT EXEC GET PROLOG1"
"GLOBAL SELECT EXEC GET PROLOG2"
"GLOBAL SELECT EXEC GET PROLOG3"
"GLOBAL SELECT EXEC GET PROLOGP1"
"GLOBAL SELECT EXEC GET PROLOGP2"
"GLOBAL SELECT EXEC GET PROLOGP3"
nextedef = "VALUE" (PROLOG*N1) || "VALUE" (PROLOG*N2) || "VALUE" (PROLOG*N3)
nextedef = "VALUE" (PROLOG*P1) || "VALUE" (PROLOG*P2) || "VALUE" (PROLOG*P3)
PARSE VAR nextedef "\\" nextedef
PARSE VAR nextedef "\\" nextedef

VM/CMS ISIS (Mainframe Version)
Mainframe VM/CSM ISIS

REXX EXEC

DO WHILE nexecdef = ""
    PARSE VAR nexecdef name " \" nexecdef
    PARSE VAR nexecdef parm " \" nexecdef
    INTERPRET name "=" parm
END

fn. = ""
ft. = ""
fs. = ""

PARSE VAR operands fn.1 operands

IF fn.1 = "" THEN fn.1 = "A"
IF LENGTH(fn.1) > 1 THEN DO
    SAY "***** error in" execname " *****"
    SAY "***** filemode" fn.1 "has more than 1 character."
    RETURN 16
END

ft.1 = "BACKUP"
fn.1 = "MF"||cln||"00"

*ESTATE* fn.1 ft.1 fn.1
rceaved = RC
IF rceaved = 0 THEN DO
    SAY "***** error in" execname " *****"
    SAY "***** backup file" fn.1 ft.1 fn.1 "exists already."
    RETURN rceaved
END

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib

"EXEC GIME" catowner catunit "= IQ STACK MODE -P"
qismreturncode = RC
PULL filemode .
IF qismreturncode = 4 THEN RETURN qismreturncode

"DLBL JYSYCT" filemode "OSH" cat

"FILEDEF * CLEAR"

"DLBL MASTER" filemode "DSN" pxn cln "MASTER (VSAM CAT JYSYCT)"
"DLBL MFLOG" filemode "DSN" pxn cln "MFLOG (VSAM CAT JYSYCT)"
"FILEDEF MFBUK DISK" fn.1 ft.1 fn.1 ":"(dcbvstop

suffix = SUBSTR(func,1,1)||SUBSTR(cln,1,2)

"FILEDEF VSPRINT DISK VSP99"suffix "LISTING A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"

VM/CM ISIS (Mainframe Version)

Mainframe VM/CSM ISIS

REXX EXEC

"FILEDEF LOGMSG DISK VSP99"suffix "LOGMSG A"
stack = "FUNC="func",\"CLN="cln,"
"FROM="from",\"TO="to",\"GEN="igen\"NEXTNO="nextno":"
push stack

"EXECOS VSP99"

vsp9returncode = RC
IF vsp9returncode = 4 THEN DO
    RETURN vsp9returncode
END

"SET CMSTYPE HT"
IF qismreturncode = 0 THEN "EXEC DROP" filemode "IQ"
"FILEDEF * CLEAR"
"DLBL * CLEAR"
"SET CMSTYPE RT"

RETURN vsp9returncode

VM/CM ISIS (Mainframe Version)
/* ISIS/CMS - TRANSACTION FILE BACKUP */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVBKUPTR EXEC */

/* cat */
/* catowner */
/* catunit */
/* dcbvbtape */
/* func */
/* pfix */
/* scin */
/* sortlib */
/* txlib */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE .. execname .

PARSE VAR parameters operands "(" options

"EXEC PROLOG" execname ")" options "FUNC-BKUP"

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */

*/
/*GLOBAL SELECT \xecc EXEC GET PROLOGN1* */
/*GLOBAL SELECT \xecc EXEC GET PROLOGN2* */
/*GLOBAL SELECT \xecc EXEC GET PROLOGN3* */
/*GLOBAL SELECT \xecc EXEC GET PROLOGF1* */
/*GLOBAL SELECT \xecc EXEC GET PROLOGF2* */
/*GLOBAL SELECT \xecc EXEC GET PROLOGF3* */

nexedef = "VALUE"(PROLOGN1) | "VALUE"(PROLOGN2) | "VALUE"(PROLOGN3)

pexedef = "VALUE"(PROLOGF1) | "VALUE"(PROLOGF2) | "VALUE"(PROLOGF3)

PARSE VAR nexedef ")" nexedef
PARSE VAR pexedef ")" pexedef

DO WHILE nexedef = ")"
PARSE VAR nexedef name ")" nexedef
PARSE VAR pexedef name ")" pexedef
INTERPRET name = "para"

VM/CMS ISIS (Mainframe Version)

END

bkuptrfile = "TRFILE"
bkupmode = "A"

CALL CPUTDG(C execname ")bkuptrfile bkupmode"

IF DATATYPE(result) = "NUM" THEN DO
  SAY "***** error in "execname" *****"
  SAY "***** file" bkuptrfile "not found."
  RETURN result
END

bkuptrfile = result

"GLOBAL TRLIB PLLIB CMSLIB" txllib sortlib

"EXEC GIME" catowner catunit "$ IQ STACK MODE -P"
gimereturncode = rc

EJECT filemode .
IF gimereturncode > 4 THEN RETURN gimereturncode

"DLIB I.JSYSCT" filemode "DSN" cat
"DLIB MASTERS" filemode "DSN" pfix scin "TRFILE (VSAM CAT I.JSYSCT)"

"FILEDEF = CLEAR"
suffix = SUBSTR(func,1,1)||SUBSTR(scin,1,2)

"FILEDEF VSPB9 DISK VSPB9 suffix "LISTING A"

"FILEDEF PIBUPP DISK PIBUPP LISTING A"

"FILEDEF IBKUPP DISK bkuptrfile "BACKUP" bkupmode ")dcbvbtape"

stack = "FUNC"("func","CLN","scin","TF="V;"

push stack

"EXECMS VSPB9"

vep@returncode = RC
IF vep@returncode = 0 THEN DO
  RETURN vep@returncode
END

"ESTATE VSPB9"suffix "LISTING A"
rcsav = RC
IF rcsav = 0 THEN "ERASE VSPB9"suffix "LISTING A"

VM/CMS ISIS (Mainframe Version)
"SET CMSTYPE HT"
IF gisereturncode = 0 THEN "EXEC DROP" filemode "IQ"
"FILEDEF = CLEAR"
"DLBL = CLEAR"
"SET CMSTYPE RI"
RETURN vsp@Sreturncode

VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS REXX EXCS

/* ISICS/CMS - GLOBAL VALIDATION OF TRANSACTION FILE (DATA BASE) */
/* ISICS/CMS - HANDLE PARAMETERS FOR CYCHECK EXEC */
/* acin = alternate cluster name */
/* cat = vsam catalog name */
/* catowner = vsam catalog unit userid */
/* catunit = vsam catalog unit */
/* catunitpw = vsam catalog unit password */
/* cin = cluster name */
/* dbn = data base name */
/* dbblink = dcb for link file */
/* inpfrat = vsp@ input format */
/* pf = prefix of data sets */
/* scin = trfile/fdvs cluster name */
/* seq = check sequencing of input */
/* sortlib = sort txtlib */
/* sortpgm = name of sort program */
/* tf = trfile required */
/* txtlib = isis txtlib */
/* uic = upper / lower case printer */

ADDRESS COMMAND
ARG parameters
PARSE SOURCE . execname .
PARSE VAR parameters operands "+" options
"EXEC PROLOG* execname "+" options "INPFRT=CHECK"
prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode
/* defaults_supported */
"GLOBAL SELECT exec GET PROLOGN1"
"GLOBAL SELECT exec GET PROLOGN2"
"GLOBAL SELECT exec GET PROLOGP1"
"GLOBAL SELECT exec GET PROLOGP2"
"GLOBAL SELECT exec GET PROLOGP3"
nextexecdef = "VALUE*(PROLOGN1*)" "VALUE*(PROLOGN2*)" "VALUE*(PROLOGN3*)""VALUE*(PROLOGP1*)" "VALUE*(PROLOGP2*)" "VALUE*(PROLOGP3*)"
PARSE VAR nexedef \"\" nexedef
PARSE VAR pexedef \"\" pexedef
DO WHILE nexedef = ""
    PARSE VAR nexedef name \"\" nexedef
    PARSE VAR pexedef para \"\" pexedef
    INTERPRET name "= para"
END
fn. = ""
ft. = ""
f.. = ""

PARSE VAR operands fa.1 operands
"ESTATE" dbn "PARM A"
rcsavct = RC
IF rcsvact = 0 THEN DO
    SAY "***** error in" execname "*****"
    SAY "***** file" dbn "PARM A not found."
END rcsvact
END
"ESTATE" dbn "VSP00 A"

vsp@stateternetcode = RC
IF vsp@stateternetcode = 0 THEN DO
    fn.1 = dbn
    ft.1 = "LINK"
    IF fa.1 = "" THEN fa.1 = "A"
    IF LENGTH(fa.1) > 1 THEN DO
        SAY "***** error in" execname "*****"
        SAY "***** filemode" fa.1 "has more than 1 character."
    RETURN 16
END
"EXEC CVODELF VERIFYF (FILE=TRFILE CATUNIT PW=CATUNITPW CUNITPW"
"EXEC CVODELF VERIFYF (FILE=MASTER CLN="CLN" CUNITPW=CUNITPW"
"EXEC CVODELF VERIFYF (FILE=MFLOG CLN="CLN" CUNITPW=CUNITPW"
"EXEC CVODELF VERIFYF (FILE=INVF CLN="CLN" CUNITPW=CUNITPW"
"EXEC CVODELF VERIFYF (FILE=MFLOG CLN="CLN" CUNITPW=CUNITPW"
"EXEC CVODELF VERIFYF (FILE=MFLOG CLN="CLN" CUNITPW=CUNITPW"
"EXEC CVODELF VERIFYF (FILE=MFLOG CLN="CLN" CUNITPW=CUNITPW"
"EXEC CVODELF VERIFYF (FILE=MFLOG CLN="CLN" CUNITPW=CUNITPW"

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib

VM/CMS ISIS (Mainframe Version)

EXEC GIME catowner catunit "A" catunitpw "IO MR STACK MODE -P"
ginereturncode = RC
FULL filemode.
IF ginereturncode > 4 THEN RETURN ginereturncode
"DLOB JSYSCT" filemode "DSN" cat
"DLOB TRFILE" filemode "DSN" px cln "TRFILE (YSAM CAT IJSYSCT"
"DLOB MASTER" filemode "DSN" px cln "MASTER (YSAM CAT IJSYSCT"
"DLOB MATER" filemode "DSN" px cln "MASTER (YSAM CAT IJSYSCT"
"DLOB MFLOG" filemode "DSN" px cln "MFLOG (YSAM CAT IJSYSCT"
"DLOB FDT" filemode "DSN" px cln "FDTYS (YSAM CAT IJSYSCT"
IF cln = "PREP" THEN DO
    "DLOB 1PFERI" filemode "DSN" px "PERI INVF (YSAM CAT IJSYSCT"
END
suffix = "U" SUBSTR(dbn,1,2)
FILEDEF = CLEAR
FILEDEF SYSPRINT DISK VSP00 suffix "SYSPRINT A"
FILEDEF LOGMSG DISK VSP00 suffix "LOGMSG A"
FILEDEF PRINTER DISK VSP00 suffix "LISTING A"
FILEDEF PLDUMP DISK PLDUMP LISTING A"
FILEDEF SRTPOS DISK dbn "PARR A"
FILEDEF LINK DISK dbn "VSP00 A" (*dcblink
stack = DBN="dbn",IPFRMT="ipfrmt",UIC="uic",SEQ="seq",TF="tf";
push stack
EXECVS VSP00

vsp@returncode = RC
IF vsp@returncode = 8 THEN DO
    RETURN vsp@returncode
END

"ESTATE VSP00 A suffix "SYSPRINT A"
rcsavct = RC
IF rcsavct = 0 THEN "ERASE VSP00 A suffix "SYSPRINT A"
"ESTATE VSP00 A suffix "LOGMSG A"
rcsavct = RC
IF rcsavct = 0 THEN "ERASE VSP00 A suffix "LOGMSG A"
IF vsp@returncode = 8 THEN DO
    SAY "no link records written"
    RETURN vsp@returncode

VM/CMS ISIS (Mainframe Version)
END
"ESTATE" dbn "VSP00 A"
rcsaved = RC
IF rcsaved = 0 THEN RETURN vsp08returncode

"ESTATE" fn.1 ft.1 fm.1
rcsaved = RC
IF rcsaved = 0 THEN "ERASE" fn.1 ft.1 fm.1

"FILEDEF = CLEAR"
"FILEDEF SYSOUT DISK "sortpgm" LISTING A"

sortpgm dbn "VSP00 A" fn.1 ft.1 fm.1 "CDCSYL NK SORT A"
sortreturncode = RC
IF sortreturncode = 0 THEN DO
RETURN sortreturncode
END

"ESTATE" dbn "VSP00 A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE" dbn "VSP00 A"

"ESTATE" sortpgm "LISTING A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE" sortpgm "LISTING A"

"DLBL MASTER" filemode "DSN" px fcln "MASTER (VSAM CAT JJSYSCT"
"DLBL MLOG" filemode "DSN" px fcln "MLOG (VSAM CAT JJSYSCT"
"DLBL INVS" filemode "DSN" px fcln "INVS (VSAM CAT JJSYSCT"
"DLBL JLOG" filemode "DSN" px fcln "JLOG (VSAM CAT JJSYSCT"

"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK VSP05"suffix "LISTING A"
"FILEDEF LOGMSG DISK VSP05"suffix "LOGMSG A"
"FILEDEF PLDUMP DISK PLDUMP LISTING A"
"FILEDEF LINK DISK" fn.1 ft.1 fm.1

stack = "UPD"
push stack

"EXECVS VSP05"

vm@rsreturncode = RC
IF vm@rsreturncode > 8 THEN DO
VM/CMS ISIS (Mainframe Version)
/* ISIS/CMS - DELETE/DEFINE VSAM FILE (INVOKE AMSESRV) */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVDELEDEF EXEC */

/* anycin = anyfile cluster name */
/* cat = vsam catalog name */
/* catowner = vsam catalog unit userid */
/* catunit = vsam catalog unit */
/* catunitpw = vsam catalog unit password */
/* cin = cluster name */
/* file = vsam file name */
/* ikcin = lookup cluster name */
/* pfx = prefix of data sets */
/* scin = trfile/fdtvs cluster name */
/* sortlib = sort txtlib */
/* txtlib = isis txtlib */

/* Note, that the convention for DEL/DEF's is as follows: */
/* the first two characters are: */
/* 'GF' for fdtvs, or */
/* 'TE' for temps, or */
/* 'MF' for master/trfile, or */
/* 'AL' for alog, or */
/* 'IF' for infv, or */
/* 'IL' for iflog, or */
/* 'AN' for any, or */
/* 'LK' for lookup, and */
/* the next four characters are: */
/* cin for trfile/fdtvs temps, or */
/* anycin for the anyfile, or */
/* ikcin for the lookup file */

ADDRESS COMMAND
ARG parameters

PARSE SOURCE . execname .

PARSE VAR parameters operands "(" options
*EXEC PROLOG* execname "(DBN= .FILE=FDTVS" options
prologreturncode = RC
IF prologreturncode -> 8 THEN RETURN prologreturncode

VM/CMS ISIS (Mainframe Version)

/* defaults_supported */

"GLOBALV SELECT #EXEC GET PROLOGN1"
"GLOBALV SELECT #EXEC GET PROLOGN2"
"GLOBALV SELECT #EXEC GET PROLOGN3"
"GLOBALV SELECT #EXEC GET PROLOGP1"
"GLOBALV SELECT #EXEC GET PROLOGP2"
"GLOBALV SELECT #EXEC GET PROLOGP3"

nexedef = "VALUE" (PROLOG*N1) || "VALUE" (PROLOG*N2) || "VALUE" (PROLOG*N3)
pexedef = "VALUE" (PROLOG*P1) || "VALUE" (PROLOG*P2) || "VALUE" (PROLOG*P3)

PARSE VAR nexedef "\" nexedef
PARSE VAR pexedef "\" pexedef

DO WHILE nexedef ~ "
PARSE VAR nexedef pars "\" nexedef
PARSE VAR pexedef pars "\" pexedef

INTERPRET name "= pars"
END

fn. = ""
ft. = ""
fm. = ""

PARSE VAR operands fn.1 operands
IF fn.1 ~ "" THEN DO
IF LENGTH(fn.1) > 8 THEN DO
SAY "**** error in" execname "****
SAY "**** filename" fn.1 "has more than 8 characters."
RETURN 16
END
END

PARSE VAR operands fn.2 operands
IF length(fn.2) > 1 THEN DO
SAY "**** error in" execname "****
SAY "**** filemode" fn.2 "has more than 1 character."
RETURN 16
END

defaultcin = ""

DO WHILE defaultcin = ""
IF file = "TEMPS" THEN DO
defaultcin = scin
VM/CMS ISIS (Mainframe Version)
Mainframe VM/CMS ISIS

REXX EXEC

IF fn.1 = ** THEN fn.1 = "OFTE\|\|deledefcin"
LEAVE
END
IF file = "TRFILE" THEN DO
deledefcin = scin
IF fn.1 = ** THEN fn.1 = "OFMF\|\|deledefcin"
LEAVE
END
IF file = "FDTYS" THEN DO
deledefcin = scin
IF fn.1 = ** THEN fn.1 = "DFOT\|\|deledefcin"
LEAVE
END
IF file = "MASTER" THEN DO
deledefcin = scin
IF fn.1 = ** THEN fn.1 = "DFMF\|\|deledefcin"
LEAVE
END
IF file = "MFLG" THEN DO
deledefcin = scin
IF fn.1 = ** THEN fn.1 = "DFWL\|\|deledefcin"
LEAVE
END
IF file = "INVF" THEN DO
deledefcin = scin
IF fn.1 = ** THEN fn.1 = "DFIF\|\|deledefcin"
LEAVE
END
IF file = "IFLG" THEN DO
deledefcin = scin
IF fn.1 = ** THEN fn.1 = "DFIL\|\|deledefcin"
LEAVE
END
IF file = "LOOKUP" THEN DO
deledefcin = lkcin
IF fn.1 = ** THEN fn.1 = "DFLK\|\|deledefcin"
LEAVE
END
IF file = "ANY" THEN DO
deledefcin = scin
IF fn.1 = ** THEN fn.1 = "DFAN\|\|deledefcin"
LEAVE
END
IF fn.1 = ** THEN DO
deledefcin = scin
END
ELSE DO
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

SAY "***** error in execname "*****"
SAY "***** for non-standard files."
"the DELEDEF must be given as an operand"
RETURN 16
END
ft.1 = "AMESERV"
fa.1 = "*"

"ESTATE" fn.1 ft.1 fa.1
rsavexp = RC
IF rsavexp = 0 THEN DO
SAY "***** error in execname "*****"
SAY "***** file fn.1 ft.1 fa.1 not found."
RETURN rsavexp
END

"GLOBAL TXTLIB PLLIB CMSLIB" txtlib sortlib
IF catuntpu = "**" THEN DO
"EXEC GIME" catuntpu catuntpu "*Q MODE -P"
gimreturncode = RC
PULL filemode ,
IF gimreturncode = 4 THEN RETURN gimreturncode
END
ELSE DO
"EXEC GIME" catuntpu catuntpu "*Q MR MODE -P"
gimreturncode = RC
PULL filemode ,
IF gimreturncode = 4 THEN RETURN gimreturncode
END

"DLBL IJSYSC" filemode "DSN" cat
"DLBL" file filemode "DSN" pxl deledefcin file "(YSGAM CAT IJSYSC"
IF fn.2 = ** THEN DO
"DLBL S|||file file.2 CMS deledefcin file
END

"EXECDS AMSERV" fn.1
asservreturncode = RC
IF asservreturncode = 0 THEN DO
RETURN asservreturncode
END

"SET CMSTYPE HT"
VM/CMS ISIS (Mainframe Version)
IF gimereturncode = 0 THEN EXEC DROP "filecode " ID "QLBL = CLEAR" "SET CNSTYPE RT"
RETURN aservreturncode

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

/* IS/CMS - IS/IS/ISO FORMAT CONVERSION (DISK ONLY) */
/* IS/CMS - HANDLE PARAMETERS FOR CVEXPORT EXEC */
/* cat */
/* catowner */
/* catunit */
/* cc */
/* ctn */
/* dbn */
/* dcfvbtape */
/* dcfv4828 */
/* elem */
/* from */
/* func */
/* idl */
/* int */
/* ipi */
/* ifi */
/* ikln */
/* ipk */
/* newl */
/* pxk */
/* sortlib */
/* to */
/* txtlib */
/* Address Command */

ARG parameters

PARSE SOURCE . . execname .

PARSE VAR parameters operands "I" options

"EXEC PROLOG" execname "(CC=E) options "FUNC=EXPORT"

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* default_supported */

"GLOBAL SELECT EXEC GET PROLOGN1" "GLOBAL SELECT EXEC GET PROLOGN2"
"GLOBAL SELECT EXEC GET PROLOGN3" "GLOBAL SELECT EXEC GET PROLOGP1"

VM/CMS ISIS (Mainframe Version)
"GLOBALY SELECT ##EXEC GET PROLOGP2"
"GLOBALY SELECT ##EXEC GET PROLOGP3"

nexedcf = "VALUE"(PROLOG*N1) || "VALUE"(PROLOG*N2) || "VALUE"(PROLOG*N3)
pexedcf = "VALUE"(PROLOG*P1) || "VALUE"(PROLOG*P2) || "VALUE"(PROLOG*P3)

PARSE VAR nexedcf "\"" nexedcf
PARSE VAR pexedcf "\"" pexedcf

DO WHILE nexedcf = "˘"
   PARSE VAR nexedcf name "\"" nexedcf
   PARSE VAR pexedcf params "\"" pexedcf
   INTERPRET name "~ pare"
END

fn. = "" 
ft. = 
fs. = ""

PARSE VAR operators fn.1 operands

IF fn.1 = "" THEN fn.1 = "VSP10"
IF LENGTH(fn.1) > 8 THEN DO
   SAY "** error in" execname "*" 
   SAY "** filename" fn.1 "has more than 8 characters."
   RETURN 16
IF LENGTH(stack) > 100 THEN DO
   SAY "** error in" execname "*" 
   SAY "** length of pararm string exceeds 100 characters."
   RETURN 16
END

VM/CMS ISIS (Mainframe Version)
REXX EXECs

/* ISIS/CMS - UPDATE MASTER FILE (ISIS INPUT FORMAT) */

/* ISIS/CMS - HANDLE PARAMETERS FOR CVGBLCHA EXEC */

# acin - alternate cluster name
# cat - vsam catalog name
# catowner - vsam catalog unit userid
# cattunit - vsam catalog unit
# catunitpw - vsam catalog unit password
# cin - cluster name
# dbn - data base name
# dbcblink - dcb for link file
# dcev4w28 - dcb for sort file
# dle - data delimiter
# from - from af range
# hit - input on hit file
# inpfmt - vspp08 input format
# px - prefix of data sets
# prfs - print all proofs
# sclin - trfile/dtvs cluster name
# skp - skip proof copies
# sortlib - sort textlib
# sortpgm - name of sort program
# sys - prefix of system data sets
# tf - trfile required
# to - to af range
# txtlib - ialis txtlib
# ulc - upper / lowercase printer

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . execname .

PARSE VAR parameters operands "(" options

"EXEC PROLOG" execname "(SKF=1)" options "INPFRM=GBL TF+N"

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

GLOBALLY SELECT execname GET PROLOG1"
GLOBALLY SELECT execname GET PROLOG2"
GLOBALLY SELECT execname GET PROLOG3"

hexedef = "VALUE"(PROLOG"N") "VALUE"(PROLOG"N2") "VALUE"(PROLOG"N3")

hexedef = "VALUE"(PROLOG"P") "VALUE"(PROLOG"P2") "VALUE"(PROLOG"P3")

PARSE VAR hexedef " " hexedef
PARSE VAR hexedef " " hexedef

DO WHILE hexedef = ""
PARSE VAR hexedef name " " hexedef
PARSE VAR hexedef para " " hexedef

INTERPRET name " " para

END

fn. = ""
ft. = ""
fm. = ""

PARSE VAR operands fn.1 operands

IF fn.1 = "" THEN fn.1 = "VSP08"
IF LENGTH(fn.1) > 8 THEN DO
SA" "error in" execname " "
SA" "filename" fn.1 "has more than 8 characters."
RETURN 16
END

fn.1 = "CARDS"
fn.1 = ""

"ESTATE" fn.1 ft.1 fm.1
rcsavd = RC
IF rcsavd = 0 THEN DO
SA" "error in" execname " 
SA" "file" fn.1 ft.1 fm.1 "not found."
RETURN rcsavd
END

"ESTATE" dbn "PARAM "
rcsavd = RC
IF rcsavd = 0 THEN DO
SA" "error in" execname " 
SA" "file" dbn "PARAM not found."
RETURN rcsavd
END

VM/CMS ISIS (Mainframe Version)
"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib
"EXEC GINE" catowner catunit "*" catunittpw "IQ MR STACK MODE -P"
GILERETURNCODE = RC
PULL filenode .
IF gilereturncode > 4 THEN DO
SAY "GISE " gilereturncode
RETURN gilereturncode
END
"SET CMSTYPE HT"
"DLBL = CLEAR"
"SET CMSTYPE RT"
"DLBL IJSYST" filenode "DSN" cat
"DLBL MASTER" filenode "DSN" px cln "MASTER (VSMAN CAT IJSYST)"
"DLBL MASTERA" filenode "DSN" px cln "MASTER (VSMAN CAT IJSYST)"
"DLBL MFLDC" filenode "DSN" px cln "MFLDC (VSMAN CAT IJSYST)"
"DLBL FDT" filenode "DSN" px cln "FDTVS (VSMAN CAT IJSYST)"
IF cln = "PREP" THEN DO
"DLBL IFPERI" filenode "DSN" px "PERI INFY (VSMAN CAT IJSYST)"
END
"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK VSP# SYSPRINT A"
"FILEDEF PRINTER DISK VSP# LISTING A"
"FILEDEF LOGMSG DISK VSP# LOGMSG A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF LINKF DISK" dbn "LINKF A l"dcblink
"FILEDEF SORTPOS DISK "dbn "FARM a"
"FILEDEF INPUT DUMMY"
"FILEDEF UPDATE DUMMY"
"FILEDEF CARDS DISK" fn.1 ft.1 fm.1
stack = "DSN"="dbn", "INPUTF"="inputf", "
stack = stack "DLN="die", "PRES=" "pres", "
stack = stack "FROM="="from", "TO="="to", "T="="t", "
stack = stack "ULC="="ulc", "HIT="="hit", "SKP="="skp", "1"
push stack
SAY "EXECVS VSP#"
"EXECVS VSP#"
vsp@returncode = RC
IF vsp@returncode > 8 THEN DO
VM/CMS ISIS (Mainframe Version)
Mainframe VM/CMS ISIS
REXX EXECs
SAY "VSP# :" vsp@returncode
RETURN vsp@returncode
END
"ESTATE VSP# SYSPRINT A"
rcsave = RC
IF rcsave = 0 THEN "ERASE VSP# SYSPRINT A"
"ESTATE VSP# LOGMSG A"
rcsave = RC
IF rcsave = 0 THEN "ERASE VSP# LOGMSG A"
IF vsp@returncode = 8 THEN DO
SAY "no link records written"
RETURN vsp@returncode
END
"ESTATE" dbn "LINK A"
linkstate = RC
IF linkstate = 0 THEN "ERASE" dbn "LINK A"
"FILEDEF = CLEAR"
"FILEDEF SYSOUT DISK" sortpg "LISTING A"
SAY sortpg dbn "LINKF A" dbn "LINK A COCSYLNK SORT x"
sortpg dbn "LINKF A" dbn "LINK A COCSYLNK SORT x"
sortreturncode = RC
IF sortreturncode = 0 THEN DO
SAY "SORTs" sortreturncode
RETURN sortreturncode
END
"ESTATE" dbn "LINKF A"
linkstate = RC
IF linkstate = 0 THEN "ERASE" dbn "LINKF A"
"SET CMSTYPE HT"
"DLBL = CLEAR"
"SET CMSTYPE RT"
"DLBL IJSYST" filenode "DSN" cat
"DLBL INVY" filenode "DSN" px cln "INVY (VSMAN CAT IJSYST)"
"DLBL IFPERI" filenode "DSN" px cln "IFPERI (VSMAN CAT IJSYST)"
"DLBL MASTER" filenode "DSN" px cln "MASTER (VSMAN CAT IJSYST)"
"DLBL MFLDC" filenode "DSN" px cln "MFLDC (VSMAN CAT IJSYST)"
VM/CMS ISIS (Mainframe Version)
"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK VSP05 LISTING A"
"FILEDEF LOGMSG DISK VSP05 LOGMSG A"
"FILEDEF PLDUMP DISK PLDUMP LISTING A"
"FILEDEF LINK DISK" dbn "LINK A"

stack = "UPD"
push stack

SAY *EXECOS VSP05*
"EXECOS VSP05"

vsp05returncode = RC
IF vsp05returncode = 8 THEN DO
  SAY "VSP05 end" vsp05returncode
RETURN vsp05returncode
END

"ESTATE VSP05 LISTING A"
rsaved = RC
IF rsaved = 8 THEN "ERASE VSP05 LISTING A"

"ESTATE VSP05 LOGMSG A"
rsaved = RC
IF rsaved = 8 THEN "ERASE VSP05 LOGMSG A"

"ESTATE" dbn "LINK A"
lkstate = RC
IF lkstate = 8 THEN "ERASE" dbn "LINK A"

"SET CMSTYPE HT"
IF gisパーテコード = 8 THEN "EXEC DROP" filemode "(O"
"FILEDEF = CLEAR"
"DBL" = CLEAR"
"SET CMSTYPE RT"

RETURN vsp05returncode

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

/* ISICMS - CREATE INVERTED FILE (DATA BASE) */
/* ISICMS - HANDLE PARAMETERS FOR CVGENIF EXEC */
/* acin */ alternate cluster name
/* cat */ vsam catalog name
/* catowner */ vsam catalog unit user id
/* catunit */ vsam catalog unit
/* catunitpass */ vsam catalog unit password
/* cin */ cluster name
/* create */ vs05 function
/* dbn */ data base name
/* dcbforlink */ ddo for link file
/* from */ from record
/* inputformat */ vs05 input format
/* prefix for data sets */
/* seq */ check sequencing of input
/* sortlib */ sort library
/* sortparams */ name of sort program
/* to */ to record
/* isfile */ file library
/* ulc */ upper/lower case printer

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . execname .

PARSE VAR parameters operands "(" options

*EXEC PROLOG* execname "," options "INPFRMT=GENIF"

prologreturncode = RC
IF prologreturncode = 8 THEN RETURN prologreturncode

/* defaults_supported */

*GLOBAL SELECT aEXEC GET PROLOG1*
*GLOBAL SELECT aEXEC GET PROLOG2*
*GLOBAL SELECT aEXEC GET PROLOG3*
*GLOBAL SELECT aEXEC GET PROLOGP1*
*GLOBAL SELECT aEXEC GET PROLOGP2*
*GLOBAL SELECT aEXEC GET PROLOGP3*

nextdef = "VALUE" (PROLOG*N1*) || "VALUE" (PROLOG*N2*) || "VALUE" (PROLOG*N3*)

VM/CMS ISIS (Mainframe Version)
pexecdef = "VALUE" (PROLOG*P1* || "VALUE" (PROLOG*P2* || "VALUE" (PROLOG*P3*))
PARSE VAR pexecdef "\\" pexecdef
PARSE VAR pexecdef "\\" pexecdef
DO WHILE pexecdef == "\\" pexecdef
PARSE VAR pexecdef name (\\) pexecdef
PARSE VAR pexecdef para "\\" pexecdef
PARSE VAR name "= para"
END
fn. = "="
ft. = "=
fa. = "=

PARSE VAR operands fn.1 operands
fn.1 = dbn
ft.1 = "LINK"
IF fn.1 = "=" THEN fn.1 = "A"

IF LENGTH(fn.1) > 1 THEN DO
  SAY "**** error in" execname " ****"
  SAY "**** filemode" fn.1 "has more than 1 character."
  RETURN 16
END

"ESTATE" dbn "PARN #"
rsaved = RC
IF rsaved == 0 THEN DO
  SAY "**** error in" execname " ****"
  SAY "**** file" dbn "PARN # not found."
  RETURN rsaved
END

"ESTATE" dbn "VSP# #"
vsp@statereturncode = RC
IF vsp@statereturncode == 0 THEN DO
  "GLOBAL TLXLIB PLLIB CMLSLIB" tlxlib sortlib
  "EXEC GME" catowner catunit "*" catunitpw "G MR STACK MODE -P"
  gimereturncode = RC
  IF gimereturncode > 4 THEN RETURN gimereturncode
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

*DLBL IJSYST filemode "DSN" cat
*DLBL MASTER filemode "DSN" pxr cin "MASTER (VSAM CAT IJSYST)"
*DLBL MASTERA filemode "DSN" pxr cin "MASTER (VSAM CAT IJSYST)"
*DLBL NFLOG filemode "DSN" pxr cin "NFLOG (VSAM CAT IJSYST)"

"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK VSP# SYSPRINT A"
"FILEDEF LOGMSG DISK VSP# LOGMSG A"
"FILEDEF PRINTER DISK VSP# LISTING A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF SORTPOS DISK" dbn "PARN #"
"FILEDEF LINKD DISK" dbn "VSP# A "dcnc1nk
stack = "DBN=":"dbn" ,INPFHT = "infhret" ,ULC="uic" ,SEQ="seq"
stack = stack||"",FROM="from",TO="to"
push stack

*EXECOS VSP#"
vsp@returncode = RC
IF vsp@returncode == 0 THEN DO
  RETURN vsp@returncode
END

"ESTATE" dbn "VSP# #"
rsaved = RC
IF rsaved == 0 THEN RETURN rsaved

"FILEDEF = CLEAR"
"FILEDEF SYSDUMP DISK sortpge "LISTING A"

"ESTATE" fn.1 ft.1 fa.1
rsaved = RC
IF rsaved == 0 THEN "ERASE" fn.1 ft.1 fa.1
sortpge dbn "VSP# A" fn.1 ft.1 fa.1 "DCPSYNK SORT #"
sortreturncode = RC
IF sortreturncode == 0 THEN DO
  RETURN sortreturncode
END

"ESTATE" dbn "VSP# A"
rsaved = RC
IF rsaved == 0 THEN "ERASE" dbn "VSP# A"
VM/CMS ISIS (Mainframe Version)
**VM/CMS ISIS (Mainframe Version)**

```
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

/* ISIS/CM - ISO/ISIS FORMAT CONVERSION (DISK ONLY) */
/* ISIS/CM - HANDLE PARAMETERS FOR CVIMPORT EXEC */
/* cat    = vaae catalog name */
/* caowner = vaae catalog unit userid */
/* catunit = vaae catalog unit */
/* caunittpu = vaae catalog unit password */
/* cc       = iso tape code */
/* cin      = cluster name */
/* dbn      = data base name */
/* dcbvtape = dcb for bkup tape */
/* dcbv4828 = dcb for hit file */
/* func     = function import/export */
/* newl     = fnf renumbering indicator */
/* out      = isis output file (master/sequential) */
/* px       = prefix of data sets */
/* sortlib  = sort tvtlib */
/* tvtlib   = isis tvtlib */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . . execname .

PARSE VAR parameters operands "/" options

"EXEC PROLOG" execname ""ICC=E"" options "FUNC=IMPORT"

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */

"GLOBAL SELECT eEXEC GET PROLOG1"
"GLOBAL SELECT eEXEC GET PROLOG2"
"GLOBAL SELECT eEXEC GET PROLOG3"
"GLOBAL SELECT eEXEC GET PROLOGP1"
"GLOBAL SELECT eEXEC GET PROLOGP2"
"GLOBAL SELECT eEXEC GET PROLOGP3"

nexecdef = "VALUE prolog*n1"|"VALUE prolog*n2"|"VALUE prolog*" prolognP2"
|"VALUE prolog*P1"

PARSE VAR nexecdef "/" nexecdef

VM/CMS ISIS (Mainframe Version)
PARSE VAR pexecdef "\" pexecdef

DO WHILE nexecdef = ""
  PARSE VAR nexecdef name "\" nexecdef
  PARSE VAR pexecdef para "\" pexecdef
  INTERPRET name = para
END

fn. = ""
ft. = ""
fs. = ""

PARSE VAR operands fn.1 operands

IF fn.1 = "" THEN fn.1 = "VSP10"
IF LENGTH(fn.1) > 8 THEN DO
  SAY "***** error in\" exeename \" *****"
  SAY "***** filename\" fn.1 "has more than 8 characters."
  RETURN 16
END
ft.1 = "CARDS"
fs.1 = "a"

"ESTATE" fn.1 ft.1 fs.1
CARD returncode = RC

fn.2 = dbn
ft.2 = "ISO"
fs.2 = "A"

stack = "CC=\"cc\", DBN=\"dbn\", FUNC=\"func\", NEWL=\"new\",
stack = stack || OUT="out";

IF LENGTH(stack) > 100 THEN DO
  SAY "***** error in\" exeename \" *****"
  SAY "***** length of parameter string exceeds 100 characters"
  RETURN 16
END

*GLOBAL TXTLIB PLLIB CMSLIB" txtlib sortlib

"EXEC GIME" catowner catunit "m" catunitpu "(Q MR STACK MODE -P"
gimreturncode = RC
PULL filemode .
IF gimreturncode > 4 THEN RETURN gimreturncode

VM/CMS ISIS (Mainframe Version)

*DLBL JISYSCT* filemode "DSN" cat
*DLBL MASTER* filemode "DSN" prefix "MASTER (VSAM CAT JISYSCT"

"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK VSP10 LISTING A"
"FILEDEF MASTER DISK" fn.2 ft.2 fs.2 "(dcvbtape
"FILEDEF PLIMDUMP DISK PLIMDUMP LISTING A"
"FILEDEF TAPE DUMMY"

IF cardereturncode = 0 THEN "FILEDEF CARDS DISK" fn.1 ft.1 fs.1

push stack

*EXECOS VSP10*

vsp18returncode = RC
IF vsp18returncode = 0 THEN DO
  RETURN vsp18returncode
END

"SET CMSTYPE HT"
IF gimreturncode = 0 THEN "EXEC DROP" filemode "(Q"
"FILEDEF = CLEAR"
"DLBL = CLEAR"
"SET CMSTYPE RT"

RETURN vsp18returncode

VM/CMS ISIS (Mainframe Version)
/* ISIS/CMS - LIST/DUMP INVERTED FILE */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVLISTIF EXEC */
/* cat */
/* catowner */
/* catunit */
/* cln */
/* dbn */
/* func */
/* from */
/* pfx */
/* sortlib */
/* styp */
/* to */
/* txtlib */
/* utyp */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . execlname .

PARSE VAR parameters operands "(" options

"EXEC PROLOG" execlname "(FUNC=LIST, FROM= , TO= , DBN= ," options

prologreturncode = RC

IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */

"GLOBALV SELECT #EXEC GET PROLOG1"
"GLOBALV SELECT #EXEC GET PROLOG2"
"GLOBALV SELECT #EXEC GET PROLOG3"
"GLOBALV SELECT #EXEC GET PROLOG4"
"GLOBALV SELECT #EXEC GET PROLOG5"
"GLOBALV SELECT #EXEC GET PROLOG6"

nexedef = "VALUE" (PROLOG"N1") | "VALUE" (PROLOG"N2") | "VALUE" (PROLOG"N3")

pexedef = "VALUE" (PROLOG"P1") | "VALUE" (PROLOG"P2") | "VALUE" (PROLOG"P3")

PARSE VAR nexedef "\" nexedef

PARSE VAR pexedef "\" pexedef

VM/CMS ISIS (Mainframe Version)

DO WHILE nexedef = ""

PARSE VAR nexedef name "\" nexedef

PARSE VAR pexedef para "\" pexedef

INTERPRET name = "para"

END

fn. = ""
ft. = ""
fm. = ""

PARSE VAR operands ft.1 operands

IF ft.1 = "SELECT" | ft.1 = "IGNORE" THEN DO

IF LENGTH(ft.1) > 8 THEN DO

SAY "****** error in" execlname " ******"

SAY "****** filetype" ft.1 "has more than 8 characters."

RETURN 16

END

fn.1 = "VSP35"
fn.1 = "s"

END

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib

"EXEC GIME" catowner catunit "s IQ STACK MODE -P"

gimereturncode = RC

PULL filemode .

IF gimereturncode > 4 THEN RETURN gimereturncode

"DL8L IJSYSC" filename "DSN" cat

"DL8L INV" filename "DSN" pfx ctn "INV" ISAM CAT IJSYSC"

"DL8L IFLOG" filename "DSN" pfx ctn "IFLOG" ISAM CAT IJSYSC"

FILEDEF = CLEAR

IF fn.1 = "" THEN "FILEDEF" "ft.1" DISK "fn.1 ft.1 fm.1"

"FILEDEF IFLIST DISK VSP35 LISTING A"

"FILEDEF SYSPRINT DISK VSP35 SYSPRINT A"

"FILEDEF FLIDUMP DISK FLIDUMP LISTING A"

stack = "FUNC" "func", "DBN" "dbn", "CLN" "cln", "fn."

stack = stack | "STYPE" "styp", "UTYPE" "utyp", "FROM" "from", "TO" "to"

push stack

"EXECOS VSP35"

vsp35returncode = RC

VM/CMS ISIS (Mainframe Version)
MAINFRAME VM/CMS ISIS

REXX EXECS

IF vsp35returncode = 0 THEN DO
  RETURN vsp35returncode
END

"ESTATE VSP35 SYSPRINT A"
rcsSaved = RC
IF rcsSaved = 0 THEN "ERASE VSP35 SYSPRINT A"

"SET CNSTYPE HT"
IF gisReturncode = 0 THEN "EXEC DROP" filemode *(G)
"FILEDEF = CLEAR"
"OLDL = CLEAR"
"SET CNSTYPE RT"
RETURN vsp35returncode

VM/CMS ISIS (Mainframe Version)

MAINFRAME VM/CMS ISIS

REXX EXECS

"/s ISIS/CMS - RESTORE TRANSACTION FILE (CLUSTER)"

"/s ISIS/CMS - HANDLE PARAMETERS FOR CVLOADTR EXEC"

"/s cat = vsam catalog name"
"/s catowner = vsam catalog unit user-id"
"/s catunit = vsam catalog unit"
"/s catunitpw = vsam catalog unit password"
"/s dcbbtape = dcbb of load tape"
"/s func = function"
"/s px = prefix of data sets"
"/s scin = tfile/fdvs cluster name"
"/s sortlib = sort txtlib"
"/s txtlib = isis txtlib"

"/s NOTE, THAT THE EXECUTION OF THIS EXEC MUST BE PRECEDED BY:
  "EXEC DELDEF (FILE=TRFILE CATUNITpw=\"password\"

"/s ADDRESS COMMAND"

ARG parameters

PARSE SOURCE . . execname .

PARSE VAR parameters operands "(" options

"EXEC PROLOG" execname "(" options "FUNC=LOAD"

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

"/s defaults_supported w/

"GLOBAL SELECT #EXEC GET PROLOG1"
"GLOBAL SELECT #EXEC GET PROLOG2"
"GLOBAL SELECT #EXEC GET PROLOG3"

nexecdef = "VALUE" (PROLOG"N1") | "VALUE" (PROLOG"N2") | "VALUE" (PROLOG"N3")
pxexecdef = "VALUE" (PROLOG"P1") | "VALUE" (PROLOG"P2") | "VALUE" (PROLOG"P3")

PARSE VAR nexecdef "]" nexecdef
PARSE VAR pxexecdef "]" pxexecdef

VM/CMS ISIS (Mainframe Version)
DO WHILE nxexecdef = ""
  PARSE VAR nxexecdef name "\"" nxexecdef
  PARSE VAR nxexecdef pare "\"" nxexecdef
  INTERPRET name = "pare"
END

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib

"EXEC GIME" catowner catunit "*" catunittpw "*Q MR STACK MODE -P"
gimereturncode = RC
FULL filecode .
IF gimereturncode > 4 THEN RETURN gimereturncode

"DLBL 1JSYSC" filemode "DSN" cat
"DLBL MASTER" filemode "DSN" px scin "TRFILE IYSAM CAT 1JSYSC"
bkuptrfile = "TRFILE"
CALL COGETGDD execname "bkuptrfile

IF DATATYPE(result) = "NUM" THEN DO
  SAY "***** error in execname " *****"
  SAY "****** file " bkuptrfile " not found."
RETURN result
END
bkuptrfile = bkuptrfile || "00"
  suffix = SUBSTR(func,1,1)||SUBSTR(scin,1,2)

"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK VSP89"suffix "LISTING A"
"FILEDEF PLDUMP DISK PLDUMP LISTING A"
"FILEDEF LOGRSG DISK VSP89"suffix "LOGRSG A"
"FILEDEF RESKUP DISK bkuptrfile "BACKUP = "dcbvbtape"
stack = "FUN"="func",CLN="scin",TF="Y";
push stack

"EXECOS VSP89"
vsp89returncode = RC
IF vsp89returncode = 0 THEN DO
  RETURN vsp89returncode
END

"ESTATE VSP89"suffix "LISTING A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE VSP89"suffix "LISTING A"

VM/CMS ISIS (Mainframe Version)
VM/CMS ISIS

REXX EXECs

/* ISIS/CMS PHOTOCOMPOSITION (VM/CMS) */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVPHOTOC EXEC */
/* acin - alternate cluster name */
/* anycin - anyfile cluster name */
/* cat - vmai catalog name */
/* catowner - vmai catalog unit userid */
/* catunit - vmai catalog unit */
/* dcbphcli - dcb for phcline */
/* dcbphoto - dcb for phdphoto */
/* dbphsav - dcb for phcsave */
/* dbpthex - dcb for phctext */
/* clin - cluster name */
/* ikcin - lookup cluster name */
/* px - prefix of data sets */
/* sortlib - sort twllib */
/* twllib - isis twllib */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE ".. execname .."

PARSE VAR parameters operands "*" options

"EXEC PROLOG" execname "DBN=", "options"

prologreturncode = RC

IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */

"GLOBAL SELECT execl GET PROLOG1"
"GLOBAL SELECT execl GET PROLOG2"
"GLOBAL SELECT execl GET PROLOG3"
"GLOBAL SELECT execl GET PROLOG4"
"GLOBAL SELECT execl GET PROLOG5"

nexedef = "VALUE"(PROLOG*N1) "VALUE"(PROLOG*N2) "VALUE"(PROLOG*N3)

nexedef = "VALUE"(PROLOG*P1) "VALUE"(PROLOG*P2) "VALUE"(PROLOG*P3)

PARSE VAR nexedef 

PARSE VAR nexedef 

VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS

REXX EXECs

DO WHILE nexedef = ""

PARSE VAR nexedef name "" nexedef

PARSE VAR nexedef pare "" nexedef

INTERPRET name = "par"

END

fn. = **
ft. = **
fs. = **

PARSE VAR operands fn.1 operands

PARSE VAR operands fn.2 operands

IF fn.1 = "" THEN fn.1 = "PHC01"

IF LENGTH(fn.1) > 8 THEN DO

SAY "******** error in' execname ".

SAY "******** filename" fn.1 "has more than 8 characters."

RETURN 16

END

ft.1 = "CARDS"
fs.1 = "*"

"ESTATE fn.1 ft.1 fs.1

rcsav = RC

IF rcsav = 0 THEN DO

SAY "******** error in' execname ".

SAY "******** file" fn.1 ft.1 fs.1 "not found."

RETURN rcsav

END

IF fn.2 = "" THEN fn.2 = "PHC02"

IF LENGTH(fn.2) > 8 THEN DO

SAY "******** error in' execname ".

SAY "******** filename" fn.2 "has more than 8 characters."

RETURN 16

END

ft.2 = "CARDS"
fs.2 = "*"

"ESTATE fn.2 ft.2 fs.2

rcsav = RC

IF rcsav = 0 THEN DO

SAY "******** error in' execname ".

SAY "******** file" fn.2 ft.2 fs.2 "not found."

RETURN rcsav

END

VM/CMS ISIS (Mainframe Version)
"ESTATE SORT HITSORT ="
saved = RC
IF saved = 0 THEN DO
  SAY "#### error in" execname = "####"
  SAY "#### file SORT HITSORT = not found."
  RETURN saved
END
"GLOBAL TLIB PLILIB CMSLIB" txtlib sortlib
"EXEC GIME" catowner catunit = "Q" STACK MODE =P"
gimereturncode = RC
PULL filemode .
IF gimereturncode > 4 THEN RETURN gimereturncode
"DLBL JYSCST" filemode = "DSN" cat
"DLBL JYVCST" filemode = "DSN" pxv clh = "INVF (VSAM CAT IJYSCST)
"DLBL JYPSST" filemode = "DSN" pxv clh = "MASTER (VSAM CAT IJYSCST)
"DLBL JYPSST" filemode = "DSN" pxv clh = "MASTER (VSAM CAT IJYSCST)
"DLBL JYPSST" filemode = "DSN" pxv clh = "ANY (VSAM CAT IJYSCST)
"DLBL JYPSST" filemode = "DSN" pxv clh = "ANY (VSAM CAT IJYSCST)
"FILEDEF = CLEAR"
"FILEDEF CARDS DISK" fl = vl . fm 1
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF SYSPRINT DISK PHC02 SYSPRINT A"
"FILEDEF COSFOTO DISK PHC02 PHOT0IN A (dcbphoto
"FILEDEF HITSORT DISK SORT HITSORT ="
"EXECOS PHC01"
phc01returncode = RC
IF phc01returncode = 0 THEN DO
  RETURN phc01returncode
END
"ESTATE PHC01 SYSPRINT A"
saved = RC
IF saved = 0 THEN "ERASE PHC01 SYSPRINT A"
"ESTATE SORT HITSORT ="
saved = RC
IF saved = 0 THEN "ERASE SORT HITSORT ="
"FILEDEF = CLEAR"
"FILEDEF CARDS DISK" fl = vl . fm 2
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

"FILEDEF PHCFILE DISK PHC02 PHOT0IN A"
"FILEDEF PHCFILE DISK PHC02 PHCFILE A ("dbcbphcf"
"FILEDEF PHCTEXT DISK PHC02 LISTING A ("dcdbphctex
"FILEDEF PHCMESS DISK PHC02 PHCMESS A"
"FILEDEF PHCTDCN DISK PHC02 TABCON A ("dcdbphoto
"FILEDEF PHCSAVE DISK PHC02 PHCSAVE A ("dcdbphsav
"FILEDEF SYSPRINT DISK PHC02 SYSPRINT A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
stack = "*"
push stack
"EXECOS PHC02"
phc02returncode = RC
IF phc02returncode = 0 THEN DO
  RETURN phc02returncode
END
"ESTATE PHC02 SYSPRINT A"
saved = RC
IF saved = 0 THEN "ERASE PHC02 SYSPRINT A"
"ESTATE PHC02 PHOT0IN A"
saved = RC
IF saved = 0 THEN "ERASE PHC02 PHOT0IN A"
"ESTATE PHC02 PHCFILE A"
saved = RC
IF saved = 0 THEN "ERASE PHC02 PHCFILE A"
"ESTATE PHC02 PHCMESS A"
saved = RC
IF saved = 0 THEN "ERASE PHC02 PHCMESS A"
"ESTATE PHC02 TABCON A"
saved = RC
IF saved = 0 THEN "ERASE PHC02 TABCON A"
"ESTATE PHC02 PHCSAVE A"
saved = RC
IF saved = 0 THEN "ERASE PHC02 PHCSAVE A"
"SET CMSTYPE HT"
IF gimereturncode = 0 THEN "EXEC DROP" filemode = "Q"
"FILEDEF = CLEAR"
"DLBL = CLEAR"
VM/CMS ISIS (Mainframe Version)
"SET CMSTYPE RT"
RETURN phoB2returncode

VM/CMS ISIS (Mainframe Version)

/ * ISICS/CMS - PROOF COPY PRINT OF ALL RECORDS IN ERROR (DATA BASE) */
/ * ISICS/CMS - HANDLE PARAMETERS FOR CVPROOF EXEC */
/ acln          = alternate cluster name
/ cat           = vspam catalog name
/ catowner      = vspam catalog unit userid
/ catunit       = vspam catalog unit
/ catunitpw     = vspam catalog unit password
/ cin           = cluster name
/ dbn           = data base name
/ dccbts88      = dcb for wrkf
/ dcblink       = dcb for link file
/ infret        = vspam input format
/ fx            = prefix of data sets
/ prfs          = print all proofs
/ scin          = tfiile/ftvs cluster name
/ seq           = check sequencing of input
/ skp           = skip proof copies
/ sortlib       = sort txtlib
/ sortpga       = name of sort program
/ tf             = tfiile required
/ txtlib        = isis txtlib
/ uic           = upper / lower case printer

ADDRESS COMMAND

ARG parameters
PARSE SOURCE . . execname .
PARSE YAR parameters operands "f" options
*EXEC PROLOG* execname *(SKP=1* options "INPFMT=PROOF"
prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode
/ * defaults_supported */
"GLOBALV SELECT #EXEC GET PROLOG1"
"GLOBALV SELECT #EXEC GET PROLOG2"
"GLOBALV SELECT #EXEC GET PROLOG3"
"GLOBALV SELECT #EXEC GET PROLOGP1"
"GLOBALV SELECT #EXEC GET PROLOGP2"
"GLOBALV SELECT #EXEC GET PROLOGP3"

VM/CMS ISIS (Mainframe Version)
nexeckdef = "VALUE"(PRLOG=*N1*) | "VALUE"(PRLOG=*N2*) | "VALUE"(PRLOG=*N3*)
prexckdef = "VALUE"(PRLOG=*P1*) | "VALUE"(PRLOG=*P2*) | "VALUE"(PRLOG=*P3*)

PARSE VAR nexeckdef "\" nexeckdef
PARSE VAR prexckdef "\" prexckdef

DO WHILE nexeckdef = "**"
  PARSE VAR nexeckdef name "\" nexeckdef
  PARSE VAR prexckdef parm "\" prexckdef
  INTERPRET name = "parm"
END

fn. = **
ft. = **
fn. = **

PARSE VAR operands fn.1 operands

"ESTATE" dbn "PARM #"
rcsavd = RC
IF rcsavd = 0 THEN DO
  SAY "***** error in" execname "*****
  SAY "***** file" dbn "PARM # not found."
RETURN rcsavd
END

fn.1 = dbn
ft.1 = "LINK"
IF fn.1 = "" THEN fn.1 = "A"

IF LENGTH(ft.1) > 1 THEN DO
  SAY "***** error in" execname "*****
  SAY "***** filemode" ft.1 has more than 1 character." RETURN 16
END

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib
"ESTATE" fn.1 ft.1 fn.1
rcsavd = RC
IF rcsavd = 0 THEN "ERASE" fn.1 ft.1 fn.1

"ESTATE WRKF SCRATCH A"
rcsavd = RC
IF rcsavd = 0 THEN "ERASE WRKF SCRATCH A"

VM/CMS ISIS (Mainframe Version)

*EXEC GIME* catowner catunit "#" catunitTips "IQ MR STACK MODE -P"
gsmreturncode = RC
Pull filename.
IF gsmreturncode > 4 THEN RETURN gsmreturncode

"DLSL J1JSYCT" filemode "DSN# cat"
"DLSL TFILE" filemode "DSN# pxn clin "TRFILE VYSAM CAT J1JSYCT"
"DLSL MASTER" filemode "DSN# pxn clin "MASTER VYSAM CAT J1JSYCT"
"DLSL MASTERA" filemode "DSN# pxn clin "MASTER VYSAM CAT J1JSYCT";
"DLSL MLOG" filemode "DSN# pxn clin "MLOG VYSAM CAT J1JSYCT"
"DLSL FDT" filemode "DSN# pxn clin "FDTYS VYSAM CAT J1JSYCT"
IF clin = "PREP" THEN DO
  "DLSL IFPERI" filemode "DSN# pxn "PERI INYF (VYSAM CAT J1JSYCT"
END

"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK VSP# SYSPRINT A"
"FILEDEF LOGMSG DISK VSP# LOGMSG A"
"FILEDEF PRINTER DISK VSP# LISTING A"
"FILEDEF PLOGDISK PLOGDISK LISTING A"
"FILEDEF SORTPOS DISK" dbn "PARM #"
"FILEDEF LINKF DISK" fn.1 ft.1 fn.1 "(dcblink"
"FILEDEF WRKF DISK WRKF SCRATCH A "dcbfnbb"
"FILEDEF CARDS DUMMY"

stack = "DBN":"dbn","NPFRMT="npfrfmt","ULC="ulc","SEQ="seq
stack = stack||".SKP="skp","PHFS="phfs","T="tf"
push stack

"EXECDOS VSP#"
"ESTATE WRKF SCRATCH A"
rcsavd = RC
IF rcsavd = 0 THEN "ERASE WRKF SCRATCH A"

vsp@returncode = RC
IF vsp@returncode > 0 THEN DO
RETURN vsp@returncode
END

"ESTATE VSP# SYSPRINT A"
rcsavd = RC
IF rcsavd = 0 THEN "ERASE VSP# SYSPRINT A"

"ESTATE VSP# LOGMSG A"
rcsavd = RC
IF rcsavd = 0 THEN "ERASE VSP# LOGMSG A"

VM/CMS ISIS (Mainframe Version)
IF vsp$RETURNCODE = 8 THEN DO
  SAY "no link records written"
  RETURN vsp$RETURNCODE
END

"ESTATE" fn.1 ft.1 fs.1
rcsaved = RC
IF rcsaved = 0 THEN RETURN vsp$RETURNCODE

"FILEDEF = CLEAR"
"FILEDEF SYSOUT DISK*" sortpgm "LISTING A"
sortpgm fn.1 ft.1 fs.1 "* * * CDCSYLNK SORT a"

ssortreturncode = RC
IF ssortreturncode = 0 THEN DO
  RETURN ssortreturncode
END

"ESTATE" sortpgm "LISTING A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE" sortpgm "LISTING A"

"DLBL MASTER" filetype "DSN" pfx cln "MASTER (VSAM CAT IJSYSGT)"
"DLBL MLOG" filetype "DSN" pfx cln "MLOG (VSAM CAT IJSYSGT)"
"DLBL INVF" filetype "DSN" pfx cln "INVF (VSAM CAT IJSYSGT)"
"DLBL IFLOG" filetype "DSN" pfx cln "IFLOG (VSAM CAT IJSYSGT)"

"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK VSPPBS LISTING A"
"FILEDEF LOGMSG DISK VSPPBS LOGMSG A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF LINK DISK*" fn.1 ft.1 fs.1

stack = "UPD"
push stack

"EXECDS VSPPBS"

vsp$RETURNCODE = RC
IF vsp$RETURNCODE = 8 THEN DO
  RETURN vsp$RETURNCODE
END

"ESTATE" fn.1 ft.1 fs.1
rcsaved = RC
VM/CMS ISIS (Mainframe Version)
Mainframe VM/CMS ISIS

REXX EXEC

/* ISIS/CMS - QUERY (BATCH RETRIEVAL, UNSORTED OUTPUT) */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVQUERY EXEC */

/* acin  - alternate cluster name */
/* angcin - angfile cluster name */
/* cat  - vsm catalog name */
/* catowner - vsm catalog unit userid */
/* catunit - vsm catalog unit */
/* cin  - cluster name */
/* dbn  - database name */
/* dbvbf4028 - dbv for tape file */
/* lkcin - lookup cluster name */
/* pfx  - prefix of data sets */
/* sortlib - sort txtlib */
/* txtlib - isis txtlib */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE .. execname ..

PARSE VAR parameters operands "(" options

"EXEC PROLOG" execname "(" options

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode
/* defaults_supported */

"GLOBALV SELECT WEXEC GET PROLOGN1" "GLOBALV SELECT WEXEC GET PROLOGN2" "GLOBALV SELECT WEXEC GET PROLOGP1" "GLOBALV SELECT WEXEC GET PROLOGP2" "GLOBALV SELECT WEXEC GET PROLOGP3" nexecdef = "VALUE"("PROLOGN1") | "VALUE"("PROLOGN2") | "VALUE"("PROLOGP1") | "VALUE"("PROLOGP2") | "VALUE"("PROLOGP3") pexecdef = "VALUE"("PROLOGP1") | "VALUE"("PROLOGP2") | "VALUE"("PROLOGP3")

PARSE VAR nexecdef ")" nexecdef
PARSE VAR pexecdef ")" pexecdef

DO WHILE nexecdef = ""

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

/* ISIS/CMS - QUERY (BATCH RETRIEVAL, UNSORTED OUTPUT) */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVQUERY EXEC */

/* acin  - alternate cluster name */
/* angcin - angfile cluster name */
/* cat  - vsm catalog name */
/* catowner - vsm catalog unit userid */
/* catunit - vsm catalog unit */
/* cin  - cluster name */
/* dbn  - database name */
/* dbvbf4028 - dbv for tape file */
/* lkcin - lookup cluster name */
/* pfx  - prefix of data sets */
/* sortlib - sort txtlib */
/* txtlib - isis txtlib */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE .. execname ..

PARSE VAR parameters operands "(" options

"EXEC PROLOG" execname "(" options

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode
/* defaults_supported */

"GLOBALV SELECT WEXEC GET PROLOGN1" "GLOBALV SELECT WEXEC GET PROLOGN2" "GLOBALV SELECT WEXEC GET PROLOGP1" "GLOBALV SELECT WEXEC GET PROLOGP2" "GLOBALV SELECT WEXEC GET PROLOGP3" nexecdef = "VALUE"("PROLOGN1") | "VALUE"("PROLOGN2") | "VALUE"("PROLOGP1") | "VALUE"("PROLOGP2") | "VALUE"("PROLOGP3") pexecdef = "VALUE"("PROLOGP1") | "VALUE"("PROLOGP2") | "VALUE"("PROLOGP3")

PARSE VAR nexecdef ")" nexecdef
PARSE VAR pexecdef ")" pexecdef

DO WHILE nexecdef = ""

VM/CMS ISIS (Mainframe Version)
"DLBL MASTER*" filemode "DSN*" px x acln "MASTER (VSM CAT IJSYSCT)"
"DLBL ANYFILE*" filemode "DSN*" px x acln "ANY (VSM CAT IJSYSCT)"
"DLBL LOOKUP*" filemode "DSN*" px x acln "LOOKUP (VSM CAT IJSYSCT)"

"FILEDEF = CLEAR"
"FILEDEF SORTDISK*" dbn "PARM s*"
"FILEDEF PRINTER DISK VSP01 LISTING A*"
"FILEDEF TAPE DISK VSP01 TAPE A* (*dcbb4828"
"FILEDEF CARDS DISK*" fn.1 ft.1 fn.1

arguments = "N" dbn

"EXECOS VSP01*" arguments

vsp01returncode = RC
IF vsp01returncode > 4 THEN DO
RETURN vsp01returncode
END

"ESTATE VSP01 TAPE A*"
rcsavcd = RC
IF rsavcd = 8 THEN "ERASE VSP01 TAPE A*"

"SET CMSTYPE HT*
"FILEDEF = CLEAR"
"DLBL = CLEAR"
IF gserrorcount = 8 THEN "EXEC DROP" filemode "Q"
"SET CMSTYPE RT*

RETURN vsp01returncode

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

/* ISCSI/CMS BATCH RETRIEVAL */
/* ISCSI/CMS - HANDLE PARAMETERS FOR CVQUERY EXEC */
/* acln = alternate cluster name */
/* act = alternate cluster name */
/* cat = vsam catalog name */
/* catowner = vsam catalog unit userid */
/* catunit = vsam catalog unit */
/* cc = force control */
/* cl = cluster name */
/* cnt = code for repeat headings (global) */
/* cu = column width */
/* dbn = data base name */
/* dcbv4828 = dcb for hitsort */
/* ind = indentation of headings */
/* lkln = lookup cluster name */
/* ip = lines per page */
/* lw = line width */
/* mode = display mode (Q/P/H) */
/* nc = number of columns */
/* nh = number of headings */
/* nt = number of totals */
/* pc = pagination code */
/* prefix of data sets */
/* pn = starting page number */
/* pt = print train (upper/lower case) */
/* rv = recto-verso */
/* skp = skip lines after heading */
/* skpl = level for which headings are to be repeated */
/* skpl = code for repeat headings (level) */
/* sort = sort required */
/* sortlib = sort textlib */
/* tol = lines available to edit record */
/* txtlib = textlib */
/* zt = zeroes in total statements */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . execname .

PARSE ARG parameters operands "(" options

VM/CMS ISIS (Mainframe Version)
"EXEC PROLOG" execname "l'options"
prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */
"GLOBAL SELECT EXEC GET PROLOGN1"
"GLOBAL SELECT EXEC GET PROLOGN2"
"GLOBAL SELECT EXEC GET PROLOGN3"
"GLOBAL SELECT EXEC GET PROLOGP1"
"GLOBAL SELECT EXEC GET PROLOGP2"
"GLOBAL SELECT EXEC GET PROLOGP3"

execdef = "VALUE"("PROLOG\*N1") | "VALUE"("PROLOG\*N2") | "VALUE"("PROLOG\*N3")
execdef = "VALUE"("PROLOG\*P1") | "VALUE"("PROLOG\*P2") | "VALUE"("PROLOG\*P3")

PARSE VAR execdef "\"" execdef
PARSE VAR execdef "\"" execdef
DO WHILE execdef = ""
PARSE VAR execdef name "\"" execdef
PARSE VAR execdef parm "\"" execdef
INTERPRET name = "par"
END

fn. = **
ft. = **
fu. = **

PARSE VAR operands fn.1 operands
IF fn.1 = "" THEN fn.1 = "VSPLY"
IF LENGTH(fn.1) > 8 THEN DO
SAY "****** error in" execname " ******"
SAY "****** filename" fn.1 "has more than 8 characters."
RETURN 16
END
ft.1 = "CARDS"
fn.1 = "\""

"ESTATE" fn.1 ft.1 fn.1
rcsave = RC
IF rcsave = 0 THEN DO
SAY "****** error in" execname " ******"
SAY "****** file" fn.1 ft.1 fn.1 "not found."
RETURN rcsave
END

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

"ESTATE" dbn "PARM \""
rcsave = RC
IF rcsave = 0 THEN DO
SAY "****** error in" execname " ******"
SAY "****** file" dbn "PARM \" not found."
RETURN rcsave
END

PARSE VAR operands fn.2 operands
IF fn.2 = "" THEN fn.2 = "VSPLY"
ft.2 = "CARDS"
fu.2 = "\""

"GLOBAL TXTLIB PLLIB CMSLIB" txtlib sortlib
EXEC GIME catowner catunit * IQ STACK MODE -P"
 Gimreturncode = RC
 fullerode .
IF fullerode > 4 THEN RETURN fullerode

"DBL 1 JYSCT" filenode "DSN" cat
"DBL INY1" filenode "DSN" px cld "INVF (VSM CAT IJYSCT)"
"DBL INY1" filenode "DSN" px cld "INVF (VSM CAT IJYSCT)"
"DBL MASTER" filenode "DSN" px cld "MASTER (VSM CAT IJYSCT)"
"DBL MASTER" filenode "DSN" px cld "MASTER (VSM CAT IJYSCT)"
"DBL ANYFILE" filenode "DSN" px cld "ANY (VSM CAT IJYSCT)"
"DBL LOOKUP" filenode "DSN" px cld "LOOKUP (VSM CAT IJYSCT)"

"FILEDEF = CLEAR"
"FILEDEF SORTPOS DISK" dbn "PARM \""
"FILEDEF PRINTER DISK VSPLY LISTING A"
"FILEDEF TAPE DISK VSPLY TAPE A ("dcbvb4828"
"FILEDEF CARDS DISK" fn.1 ft.1 fu.1
"FILEDEF HITSORT DISK VSPLY HITSORT A ("dcbvb4828"

arguments = sort dbn
"EXECVS VSPLY" arguments
vsplreturncode = RC
IF vsplreturncode = 0 THEN DO
RETURN vsplreturncode
END

VM/CMS ISIS (Mainframe Version)
FILEDEF *CLEAR*
FILEDEF SORTRGS DISK" dnn "*PARN*"*A*
FILEDEF CARDS DISK" fn.2 ft.2 fn.2
FILEDEF HITSORT DISK VSP01 HITSORT A*
FILEDEF PRINTER DISK VSP04 LISTING A*
FILEDEF SYSPRINT DISK VSP04 SYSPRINT A*
FILEDEF PLODUMP DISK PLODUMP LISTING A*

stack = "iu", "cu", "lp", "nc", "nh", "pt", "mode",
stack = stack ["cc", "pc", "pn", "skp", "ind", "to", "",
stack = stack ["skpl", "cnt", "skpl", "nt", "zt", "rv",
push stack

EXECOS VSP04*

vsp@4returncode = RC
IF vsp@4returncode = @ THEN DO
RETURN vsp@4returncode
END

"ESTATE VSP04 SYSPRINT A*
rcased = RC
IF rcased = @ THEN "ERASE VSP04 SYSPRINT A"

"ESTATE VSP01 TAPE A*
rcased = RC
IF rcased = @ THEN "ERASE VSP01 TAPE A"

"ESTATE VSP01 HITSORT A*
rcased = RC
IF rcased = @ THEN "ERASE VSP01 HITSORT A"

"SET CMSTYPE HIT*
"FILEDEF = CLEAR"
"DBL = CLEAR"
IF giga@returncode = @ THEN "EXEC DROP" filename "(Q"
"SET CMSTYPE RT*
RETURN vsp@4returncode
PARSE VAR pexecdef pare "\" pexecdef
INTERPRET name "= pare"
END

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib

"EXEC GIME" catowner catunit "*" catunitp "*Q MR STACK MODE -P"
gimereturncode = RC
PULL filemode.
IF gimereturncode > 4 THEN RETURN gimereturncode

"DLBL IJSYSC" filemode "DSN" cat
"DLBL MASTER" filemode "DSN" px scln "TRFILE (YSAM CAT 1J5YSCT"

bkuptrfile = "TRFILE"
CALL CDGETGOG execname "("bkuptrfile

IF DATATYPE(result) = "NUM" THEN DO
  SAY "******** error in" execname " ********"
  SAY "******** file" bkuptrfile "not found."
  RETURN result
END
bkuptrfile = bkuptrfile || "00"
  suffix = SUBSTR(func,1,1)||SUBSTR(sc1n,1,2)

  *FILEDEF = CLEAR*
  *FILEDEF SYSPRINT DISK VSP09*suffix "LISTING A"
  *FILEDEF PLDUMP DISK PLDUMP LISTING A*
  *FILEDEF LOGMSG DISK VSP09*suffix "LOGMSG A"
  *FILEDEF MFBKUP DISK* bkuptrfile "BACKUP = ("dcbvbtape
  stack = "FUNC="func",DBN="dbn",CLN="scln",TF="Y";"
  push stack

  *EXEC VSP09*
  vsp9returncode = RC
  IF vsp9returncode = 0 THEN DO
    RETURN vsp9returncode
  END

  "ESTATE VSP09*suffix "LISTING A"
rceaved = RC
  IF rceaved = 0 THEN "ERASE VSP09*suffix "LISTING A"
  "SET CMSTYPE HT"
VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS (Mainframe Version)
/* ISIS/CMS - LOAD LOOKUP FILE */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVRTY36 EXEC */
/*
/cat = vsam catalog name
/catowner = vsam catalog unit userid
/catunit = vsam catalog unit
/catunitpw = vsam catalog unit password
/lkcln = lookup cluster name
/dl = data length
/dp = data position
/kl = key length
/kp = key position
/pfx = prefix of data sets
/sortlib = sort txtlib
/txtlib = isis txtlib
*/
/*
NOTE, THAT THE EXECUTION OF THIS MUST BE PRECEDED BY:
*EXEC DEDDEF (FILE=LOOKUP CATUNITPW=password
*/

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . . execname .

PARSE VAR parameters operands "(" options

*EXEC PROLOG" execname ")"options

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */

*GLOBAL SELECT #EXEC GET PROLOGN1*
*GLOBAL SELECT #EXEC GET PROLOGN2*
*GLOBAL SELECT #EXEC GET PROLOGN3*
*GLOBAL SELECT #EXEC GET PROLOGP1*
*GLOBAL SELECT #EXEC GET PROLOGP2*
*GLOBAL SELECT #EXEC GET PROLOGP3*

nexecdef = "VALUE" (PROLOG*N1) "VALUE" (PROLOG*N2) "VALUE" (PROLOG*N3)
nexecdef = "VALUE" (PROLOG*P1) "VALUE" (PROLOG*P2) "VALUE" (PROLOG*P3)

PARSE VAR nexecdef "\" nexecdef

VM/CMS ISIS (Mainframe Version)
 Mainframe VM/CMS ISIS

REXX EXEC

stack = "KP="kp", KL="kl", DP="dp", DL="dl";"
push stack

"EXECOS RTRY36"
rtv36returncode = RC
IF rtv36returncode = 0 THEN DD
RETURN rtv36returncode
END

"SET CMSTYPE HT"
IF ginerrtno = 0 THEN "EXEC DROP" filecode "IQ"
"FILEDEF = CLEAR"
"DLBL = CLEAR"
"SET CMSTYPE RT"
RETURN rtv36returncode

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

/* ISICMS/CMS - PROCESS ENTRY PROCESSOR EXTRAPARTITION DATA SET */
/* WITH A LIST OF MN/DBN SORTED BY TERMINAL */
/* (TP/HIP PRODUCED BY CDEP / CDEPUT04 ) */
/* ISICMS/CMS - HANDLE PARAMETERS FOR CYTERMINL EXEC */
*/
/* acin = alternate cluster name */
/* cat = vsm catalog name */
/* catowner = vsm catalog unit userid */
/* catunit = vsm catalog unit */
/* catunitp = vsm catalog unit password */
/* cln = cluster name */
/* dbn = data base name */
/* dcblink = dcb for link file */
/* dcbsv=482 = dcb for sort file */
/* dtd = field definition table */
/* dcb = (if different from dbn) */
/* inpt = vspa input format */
/* pfx = prefix of data sets */
/* pte = list of proof copies sorted by term. */
/* sclin = trfile/dtvs cluster name */
/* seq = check sequencing of input */
/* skp = skip proof copies */
/* sortlib = sort txtlib */
/* sortpgm = name of sort program */
/* sys = prefix of system data sets */
/* txtlib = isis txtlib */
/* ulc = upper / lower case printer */

ADDRESS COMMAND
ARG parameters
PARLE SOURCE . . execname .
PARLE VAR parameters operands "(* options"
EXEC PROLOC execname "(SKP=1)" options
prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode
/* defaults_supported */
"GLOBAL SELECT wEXEC GET PROLOGN1"
"GLOBAL SELECT wEXEC GET PROLOGN2"

VM/CMS ISIS (Mainframe Version)
"GLOBALY SELECT #EXEC GET PROLOGN3"
"GLOBALY SELECT #EXEC GET PROLOGP1"
"GLOBALY SELECT #EXEC GET PROLOGP2"
"GLOBALY SELECT #EXEC GET PROLOGP3"

nexedc = "VALUE"(PROLOGN1)"VALUE"(PROLOGP1)"VALUE"(PROLOGP2)"VALUE"(PROLOGP3)
pexedc = "VALUE"(PROLOGP1)"VALUE"(PROLOGP2)"VALUE"(PROLOGP3)

PARSE VAR nexedc \"\" nexedc
PARSE VAR pexedc \"\" pexedc

DO WHILE nexedc = "" PARSE VAR nexedc name \"\" nexedc
PARSE VAR pexedc name \"\" pexedc

INTERPRET name = "parm"
END

IF rdt = "" THEN rdt = dbn
suffix = "U"|SUBSTR(rdt,1,2)

"ESTATE" dbn "PARAM "
rcsaved = RC
IF rcsaved = 0 THEN DO
SAY "****** error in" execname = "******"
SAY "****** file" dbn "PARAM not found."
RETURN rcsaved
END

"ESTATE" dbn"UTLIB "
rcsaved = RC
IF rcsaved = 0 THEN DO
SAY "****** error in" execname = "******"
SAY "****** UTLIB failed in previous run."
RETURN rcsaved
END

"ESTATE" dbn"SORT "
rcsaved = RC
IF rcsaved = 0 THEN DO
SAY "****** error in" execname = "******"
SAY "****** SORT failed in previous run."
RETURN rcsaved
END

"ESTATE" dbn"LINK "
linkstatereturncode = RC

VM/CMS ISIS (Mainframe Version)
*FILEDEF PRINTER DISK UTL00* suffix "LISTING A"
*FILEDEF SYSPRINT DISK UTL00* suffix "SYSPRINT A"
*FILEDEF PLDUMP DISK PLDUMP LISTING A"
*FILEDEF INPUT DUMMY"
*FILEDEF SORTPOS DISK* dbn "PARN *"
*FILEDEF JKFILE DISK UTL00 JKFILE A ('dcvb48028"
IF psa = 'N' THEN DO
*FILEDEF TPINP DISK* dbn "INPUT A"
*FILEDEF OUT DISK* dbn "SORT A ('dcvb48028"
END ELSE DO
*FILEDEF TPINP DISK* sys "INPUT A"
*FILEDEF OUT DISK* dbn "UTL00 A ('dcvb48028"
END

arguments = fdt dbn
SAY "EXECOS UTL00" arguments
*EXECOS UTL00* arguments
utl00-returncode = RC
IF utl00-returncode >= 2000 THEN DO
SAY "UTL00 : utl00-returncode"
RETURN utl00-returncode
END

*ESTATE UTL00* suffix "LISTING A"
sortstate = RC
IF sortstate = 0 THEN "ERASE UTL00* suffix "LISTING A"

*ESTATE UTL00* suffix "SYSPRINT A"
sortstate = RC
IF sortstate = 0 THEN "ERASE UTL00* suffix "SYSPRINT A"
IF psa = 'N'. THEN DO
*FILEDEF = CLEAR"
SAY sortpsm dbn "UTL00 A" dbn "SORT A CCDSNP SORT *"
sortpsm dbn "UTL00 A" dbn "SORT A CCDSNP SORT *"
ESORTRETURNCODE = RC
IF ESORTRETURNCODE = 0 THEN DO
SAY "SORT : " ESORTRETURNCODE
RETURN ESORTRETURNCODE
END

VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS

*ESTATE* dbn "UTL00 A"
sortstate = RC
IF sortstate = 0 THEN "ERASE" dbn "UTL00 A"
END

*SET CMSTYPE HT*
*DLBL = CLEAR"
*SET CMSTYPE RT*

suffix = SUBSTR(infnum,1,1))|SUBSTR(fdt,1,2)

*DLBL JYSYST* filemode "DSN" cat
*DLBL MASTER* filemode "DSN" pfx cin "MASTER (YSAM CAT JYSYST)"
*DLBL MASTERA* filemode "DSN" pfx acin "MASTER (YSAM CAT JYSYST)"
*DLBL MFLDC* filemode "DSN" pfx cin "MFLDC (YSAM CAT JYSYST)"
*DLBL TRFILE* filemode "DSN" pfx cin "TRFILE (YSAM CAT JYSYST)"
*DLBL FDT* filemode "DSN" pfx cin "FDTYS (YSAM CAT JYSYST)"

IF cin = "PREP" THEN DO
*DLBL 1PERI* filemode "DSN" pfx "PERI INVP (YSAM CAT JYSYST)"
END

*FILEDEF = CLEAR"
*FILEDEF SYSPRINT DISK YSP00* suffix "SYSPRINT A"
*FILEDEF PRINTER DISK YSP00* suffix "LISTING A"
*FILEDEF LOGMSG DISK YSP00* suffix "LOGMSG A"
*FILEDEF PLDUMP DISK PLDUMP LISTING A"
*FILEDEF LINKF DISK* dbn "YSP00 A ('dcvblink"
*FILEDEF SORTPOS DISK* dbn "PARN *"
*FILEDEF INPUT DISK* dbn "SORT A"
*FILEDEF UPDATE DUMMY"
*FILEDEF CARDS DUMMY"

stack = "DBN=""fdt"",INFRMT=""infnum"","
push stack
SAY "EXECOS YSP00"
*EXECOS YSP00* vsp00-returncode = RC
IF vsp00-returncode > 16 THEN DO
SAY "YSP00 : vsp00-returncode"
RETURN vsp00-returncode
END

VM/CMS ISIS (Mainframe Version)
**Mainframe VM/CMS ISIS**

**REXX EXEC**

```
*ESTATE* VSP00"suffix "SYSPRINT A"
rcsaeved = RC
IF rcsaved = 0 THEN "ERASE VSP00"suffix "SYSPRINT A"

*ESTATE* VSP00"suffix "LOGMSG A"
rcsaeved = RC
IF rcsaved = 0 THEN "ERASE VSP00"suffix "LOGMSG A"

*ESTATE* dbn "SORT A"
rcsaeved = RC
IF rcsaved = 0 THEN "ERASE" dbn "SORT A"

IF pste = "N" THEN DO

*ESTATE* sys "ACCT A"
acctreturncode = RC

*ESTATE* sys "ISAC A"
isacreturncode = RC

IF acctreturncode = 0 & isacreturncode = 0 THEN DO

   SAY COPYF sys "ACCT A" sys "ISAC A" dbn "USER A (REP"
   COPYF sys "ACCT A" sys "ISAC A" dbn "USER A (REP"

   copyreturncode = RC
   IF copyreturncode = 0 THEN DO
      SAY "COPYF:" copyreturncode
      RETURN copyreturncode
   END

   SAY sortpgm dbn "USER A" dbn "USER A CCDSISAC SORT 1 OUT REP"
   say sortpgm dbn "USER A" dbn "USER A CCDSISAC SORT 1 OUT REP"

   asortreturncode = RC
   IF asortreturncode = 0 THEN DO
      SAY "SORT:" asortreturncode
      RETURN asortreturncode
   END

END

"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK UTILS"suffix "LISTING A"
"FILEDEF PIDUMP DISK PIDUMP LISTING A"
"FILEDEF INPUT DISK" dbn "INPUT A"
```

VM/CMS ISIS (Mainframe Version)

**Mainframe VM/CMS ISIS**

**REXX EXEC**

```
*ESTATE* dbn "USER A"
rcsaeved = RC
IF rcsaved = 0 THEN "FILEDEF USER DISK" dbn "USER A"
ELSE "FILEDEF USER DUMMY"

stack = ftd
push stack

SAY "EXEC EXEC UTILS"
"EXEC EXEC UTILS"

utilisreturncode = RC
IF utilisreturncode = 2000 THEN DO
   SAY "UTILIS:" utilisreturncode
   RETURN utilisreturncode
END

*ESTATE* dbn "USER A"
rcsaeved = RC
IF rcsaved = 0 THEN "ERASE" dbn "USER A"

*ESTATE* dbn "INPUT A"
rcsaeved = RC
IF rcsaved = 0 THEN "ERASE" dbn "INPUT A"

SAY COPYF VSP00"suffix "LISTING A" UTILIS"suffix "LISTING A",
   VSP00"suffix "LISTING A (REP",
   COPYF VSP00"suffix "LISTING A UTILIS"suffix "LISTING A",
   VSP00"suffix "LISTING A (REP"

copyreturncode = RC
IF copyreturncode = 0 THEN DO
   SAY "COPYF:" copyreturncode
   RETURN copyreturncode
END

*ESTATE UTILIS"suffix "LISTING A"
rcsaeved = RC
IF rcsaved = 0 THEN "ERASE UTILIS"suffix "LISTING A"
```

VM/CMS ISIS (Mainframe Version)
"DLBL * CLEAR"
"SET CMSTYPE RT"
RETURN vspp0@returncode
END

"FILEDEF * CLEAR"

say sortpgs ddb演唱会 "LINK A CDCSYLK SORT a"
sortpgs ddb演唱会 "LINK A CDCSYLK SORT a"

ssortreturncode = RC
if ssortreturncode = 0 then do
say "SSORT: ssortreturncode"
return ssortreturncode
end

"ESTATE" ddb演唱会 "VSP80 A"
rcsavaed = RC
if rcsavaed = 0 then "ERASE" ddb演唱会 "VSP80 A"
end

"SET CMSTYPE HT"
"DLBL * CLEAR"
"SET CMSTYPE RT"
suffix = "U"||SUBSTR(fdt1,1,2)

"DLBL IJSYSC" filemode "DSN" cat
"DLBL INVF" filemode "DSN" px c ln "INVF (YSAM CAT IJSYSC)"
"DLBL IF" filemode "DSN" px c ln "IFLOG (YSAM CAT IJSYSC)"
"DLBL MASTER" filemode "DSN" px c ln "MASTER (YSAM CAT IJSYSC)"
"DLBL MFLDG" filemode "DSN" px c ln "MFLDG (YSAM CAT IJSYSC)"

"FILEDEF * CLEAR"
"FILEDEF SYSPRT DISK VSP80suffix LISTING A"
"FILEDEF LOGMSG DISK VSP80suffix LOGMSG A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF LINK DISK" ddb演唱会 "LINK A"

stack = "UPD"
push stack
say "EXECAP VSP80"
"EXECAP VSP80"

VM/CMS ISIS (Mainframe Version)
/* ISIS/CMS - TRANSACTION FILE - RESTORE DATA BASE */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVTRINIT EXEC */

/* cat = veam catalog name */
/* catowner = veam catalog unit userid */
/* catunit = veam catalog unit */
/* catunitpu = veam catalog unit password */
/* dbn = data base name */
/* func = function */
/* nextno = next af number */
/* px = prefix of data sets */
/* scin = trfile/fdsvc cluster name */
/* sortlib = sort txtlib */
/* txtlib = isis txtlib */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . , execname .

PARSE VAR parameters operands "(" options

"EXEC PROLOC" execname "("options "FUNC=INIT"

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */

"GLOBAL SELECT .EXEC GET PROLOGN1"
"GLOBAL SELECT .EXEC GET PROLOGN2"
"GLOBAL SELECT .EXEC GET PROLOGN3"
"GLOBAL SELECT .EXEC GET PROLOGP1"
"GLOBAL SELECT .EXEC GET PROLOGP2"
"GLOBAL SELECT .EXEC GET PROLOGP3"

nexthexdef = "VALUE" (PROLOG"N1") | "VALUE" (PROLOG"N2") | "VALUE" (PROLOG"N3")
nexthexdef = "VALUE" (PROLOG"P1") | "VALUE" (PROLOG"P2") | "VALUE" (PROLOG"P3")

PARSE VAR nexthexdef "\" nexthexdef
PARSE VAR nexthexdef "\" hexdef

DO WHILE nexthexdef = ""
PARSE VAR nexthexdef name "\" nexthexdef

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

PARSE VAR pexecdef pars "\" pexecdef
INTERPRET name "= pars"
END

"GLOBAL TXTLIB PLLIB CMSLIB" txtlib sortlib

"EXEC GNE" catowner catunit "*" catunitpu "IQ MR STACK MODE =P"
ginerreturncode = RC
FULL filemode .
IF ginerreturncode > 4 THEN RETURN ginerreturncode

"DLBL IJSYCT" filemode "DSN" cat
"DLBL MASTER" filemode "DSN" pxw scin "TRFILE (YSAM CAT IJSYCT"

suffix = SUBSTR(func,1,1) || SUBSTR(scin,1,2)

"FILEDEF = CLEAR"
"FILEDEF SYSPRINT DISK VSP09"suffix "LISTING A"
"FILEDEF PLDUMP DISK PLDUMP LISTING A"

stack = "FUNC="func",DBN="dbn",CLN="scin",NEXTNO="nextno";
push stack

"EXECDS VSP09"

vsp09returncode = RC
IF vsp09returncode = 0 THEN DO
RETURN vsp09returncode
END

"ESTATE VSP09"suffix "LISTING A"
rsaved = RC
IF rsaved = 0 THEN "ERASE VSP09"suffix "LISTING A"

"SET CMSTYPE HT"
IF ginerreturncode = 0 THEN "EXEC DROP" filemode "IQ"
"FILEDEF = CLEAR"
"DLBL = CLEAR"
"SET CMSTYPE RT"

RETURN vsp09returncode

VM/CMS ISIS (Mainframe Version)
Mainframe VM/CMIS ISIS
REXX EXEC

IF rcsaved ^= 0 THEN DO
  SAY "***** error in execname " execname " found."
  RETURN rcsaved
END

*GLOBAL TXTLIB PLILIB CMSLIB* txtlib sortlib
EXEC GIME catowner catunit *"m" catunitpu "IQ MR STACK MODE -P"
gimereturncode = RC
FULL filenode,.
IF gimereturncode > 4 THEN DO
  SAY "GIME ; gimereturncode"
  RETURN gimereturncode
END

*SET CSTYPE HT*
*DLBL = CLEAR*
*SET CSTYPE R1*

"DLBL JJSYSC* filenode "DSN" cat
"DLBL MASTER" filenode "DSN" pxcln "MASTER (VSAM CAT JJSYSC)"
"DLBL MASTERA" filenode "DSN" px cln "MASTER (VSAM CAT JJSYSC)"
"DLBL MFLOG" filenode "DSN" px cln "MFLOG (VSAM CAT JJSYSC)"
"DLBL FDT" filenode "DSN" px cln "FTV5S (VSAM CAT JJSYSC)"
IF cln = "PREP" THEN DO
  "DLBL IFPERI" filenode "DSN" pxl "PERI INVF (VSAM CAT JJSYSC)"
END

„FILEDEF = CLEAR“
„FILEDEF SYSPRINT DISK VSP00 SYSPRINT A“
„FILEDEF PRINTER DISK VSP00 LISTING A“
„FILEDEF LOGMSG DISK VSP00 LOGMSG A“
„FILEDEF PLIDUMP DISK PLIDUMP LISTING A“
„FILEDEF LINKF DISK" dbn "LINKF A (" dcblink
„FILEDEF SORTPOS DISK" dbn "P ARM A“
„FILEDEF INPUT DUMMY“
IF updatedelay = 0 THEN "FILEDEF CARDS DISK" fn.1 ft.1 fm.1
ELSE "FILEDEF CARDS DUMMY“
IF updatedelay = 0 THEN "FILEDEF UPDATE DISK" fn.2 ft.2 fm.2
ELSE "FILEDEF UPDATE DUMMY“

stack = "DBN=""dn""", "INFMT=""inffmt"",.
stack = stack [ "DLN=""dln"", "PRFS=""prf"",.
stack = stack [ "TF=""tf"",.
stack = stack [ "ULC=""ulc"", "SKP=""skp"",.
push stack

Mainframe VM/CMIS ISIS
REXX EXEC

SAY "EXECDS VSP00"
"EXECDS VSP00" vsp00returncode = RC
IF vsp00returncode ^= 0 THEN DO
  SAY "VSP00 ; vsp00returncode"
  RETURN vsp00returncode
END

"ESTATE VSP00 SYSPRINT A" rcsaved = RC
IF rcsaved ^= 0 THEN "ERASE VSP00 SYSPRINT A"

"ESTATE VSP00 LOGMSG A" rcsaved = RC
IF rcsaved ^= 0 THEN "ERASE VSP00 LOGMSG A"

IF vsp00returncode ^= 0 THEN DO
  SAY "no link records written"
  *SET CSTYPE HT*
  "FILEDEF = CLEAR"
  "DLBL = CLEAR"
  *SET CSTYPE R1*
  RETURN vsp00returncode
END

IF vsp00returncode ^= -4 THEN DO
  "ESTATE" dbn "LINK A"
  linkstate = RC
  IF linkstate ^= 0 THEN "ERASE" dbn "LINK A"
  "FILEDEF = CLEAR"
  "FILEDEF SYSOUT DISK "sortpm" LISTING A"
  SAY sortpm dbn "LINKF A" dbn "LINK A CCDSYLNK SORT *"
  sortpm dbn "LINKF A" dbn "LINK A CCDSYLNK SORT *"
  asortreturncode = RC
  IF asortreturncode ^= 0 THEN DO
    SAY "ASORT ; asortreturncode"
    RETURN asortreturncode
  END
  "ESTATE" dbn "LINKF A"

VM/CMIS ISIS (Mainframe Version)
linkstate = RC
IF linkstate = 8 THEN "ERASE" dbn "LINKF A"

*SET CMSTYPE HT*
*DBL = CLEAR*
*SET CMSTYPE RT*

*DBL IJSYST "filemode" DSN "cat*
*DBL INVF "filemode" DSN "px cin " INVF (VSAM CAT IJSYST)*
*DBL IFLOG "filemode" DSN "px cin " IFLOG (VSAM CAT IJSYST)*
*DBL MASTER "filemode" DSN "px cin " MASTER (VSAM CAT IJSYST)*
*DBL MFILE "filemode" DSN "px cin " MFILE (VSAM CAT IJSYST)*

*FILEDEF = CLEAR*
*FILEDEF SYSPRINT DISK VSP05 LISTING A*
*FILEDEF LOGMSG DISK VSP05 LOGMSG A*
*FILEDEF PLIDUMP DISK PLIDUMP LISTING A*
*FILEDEF LINK DISK" dbn "LINK A"

stack = "UPD"
push stack

SAY "EXECOS VSP05"
"EXECOS VSP05"

vsp05returncode = RC
IF vsp05returncode > 8 THEN DO
SAY "VSP05: vesp05returncode"
RETURN vsp05returncode
END

*ESTATE VSP05 LISTING A*
rceaved = RC
IF rceaved = 8 THEN "ERASE VSP05 LISTING A"

*ESTATE VSP05 LOGMSG A*
rceaved = RC
IF rceaved = 8 THEN "ERASE VSP05 LOGMSG A"

*ESTATE dbn "LINK A"
linkstate = RC
IF linkstate = 8 THEN "ERASE" dbn "LINK A"

END

*SET CMSTYPE HT*
IF gimreturncode = 0 THEN "EXEC DROP" filemode "Q"

VM/CMS ISIS (Mainframe Version)
/* ISIS/CMS - UPDATE MASTER & INVERTED FILES (CDS INPUT FORMAT) */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVUPDATE EXEC */
/* acln = alternate cluster name */
/* cat = vsm catalog name */
/* catowner = vsm catalog unit userid */
/* catunit = vsm catalog unit */
/* catunitpw = vsm catalog unit password */
/* cln = cluster name */
/* dbn = data base name */
/* dcbffile = dcb for link file */
/* dcbv0828 = dcb for sort file */
/* ddbname = vsp0 input format */
/* dffn = prefix of data sets */
/* dcnm = tfi5e/in6e cluster name */
/* dseq = check sequencing of input */
/* dskp = skip proof copies */
/* dsortib = sort tfi5lib */
/* dsortpqe = name of sort program */
/* dsys = prefix of system data sets */
/* dffr = tfi5lib required */
/* dtlib = isis tfi5lib */
/* duc = upper / lower case printer */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . . . execname . .

PARSE VAR parameters operands "(" options

"EXEC PROLOG" execname "(SKP--1" options "INPFRMT=CDS TF-W"

prologreturncode = RC

IF prologreturncode = 0 THEN RETURN prologreturncode

/* default_supported */
"GLOBALY SELECT <EXEC GET PROLOG1>
"GLOBALY SELECT <EXEC GET PROLOG2>
"GLOBALY SELECT <EXEC GET PROLOG3>
"GLOBALY SELECT <EXEC GET PROLOG4>
"GLOBALY SELECT <EXEC GET PROLOG5>
"GLOBALY SELECT <EXEC GET PROLOG6>

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXECs

nexecdf = "VALUE" (PROLOG"N1") | "VALUE" (PROLOG"N2") | "VALUE" (PROLOG"N3")
pexecdf = "VALUE" (PROLOG"P1") | "VALUE" (PROLOG"P2") | "VALUE" (PROLOG"P3")

PARSE VAR nexecdf "" nexecdf
PARSE VAR pexecdf "" pexecdf

DO WHILE nexecdf = ""

PARSE VAR nexecdf name "" nexecdf
PARSE VAR pexecdf para "" pexecdf

INTERPRET name = "para"

END

fn. = ""
ft. = ""
fs. = ""

PARSE VAR operands fn.1 operands

IF fn.1 = "" THEN fn.1 = "UTL00"

IF LENGTH(fn.1) > 8 THEN DO

SAY "#### error in" execname "####

SAY "#### filename" fn.1 "has more than 8 characters."

RETURN 16

END

ft.1 = "INPUT"
fs.1 = "*

*ESTIMATE* fn.1 ft.1 fs.1
rcsaved = RC

IF rcsaved = 0 THEN DO

SAY "#### error in" execname "####

SAY "#### file" fn.1 ft.1 fs.1 "not found."

RETURN rcsaved

END

*ESTIMATE* dbn "PARM *"
rcsaved = RC

IF rcsaved = 0 THEN DO

SAY "#### error in" execname "####

SAY "#### file" dbn "PARM * not found."

RETURN rcsaved

END

*ESTIMATE* dbn "UTL00 *"
rcsaved = RC

IF rcsaved = 0 THEN DO

VM/CMS ISIS (Mainframe Version)
SAY "****** error in" execname " ******"
SAY "****** UTL00 failed in previous run."
RETURN rcsaved
END

"ESTATE" dbn "SORT *"
rcsaved = RC
IF rcsaved = 0 THEN DO
SAY "****** error in" execname " ******"
SAY "****** SORT failed in previous run."
RETURN rcsaved
END

"ESTATE" dbn "LINK *
linkstatusreturncode = RC
IF linkstatusreturncode = 0 THEN DO
SAY "****** error in" execname " ******"
SAY "****** VSP05 failed in previous run."
RETURN 16
END

"ESTATE" dbn "VSP05 *
vsp05statusreturncode = RC
IF vsp05statusreturncode = 0 THEN DO
SAY "****** error in" execname " ******"
SAY "****** LINK file SORT failed in previous run."
RETURN 16
END

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib
"EXEC GIME" catnumr callunit "*" catunitpp "IQ MR STACK MODE -P"
gimreturncode = RC
PULL filename.
IF gimreturncode > 4 THEN DO
SAY "GIME *" gimreturncode
RETURN gimreturncode
END
suffix = "U"||SUBSTR(dbn,1,2)
"DBL JJSYSTCT" filemode "DSN" cat
"DBL INF*" filemode "DSN" pfx cin "INF (VSYAM CAT IJSYSTC"
"DBL IFILE*" filemode "DSN" pfx cin "IFLOG (VSYAM CAT IJSYSTC"
"FILEDEF = CLEAR*

---

Mainframe VM/CMS ISIS

REXX EXEC

"FILEDEF PRINTER DISK UTL00*suffix "LISTING A"
"FILEDEF SYSPRINT DISK UTL00*suffix "SYSPRINT A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF INPUT DISK fn.1 fn.1 fn.1
"FILEDEF JFILE DISK UTL00 JFILE A (*dcvsb4028
"FILEDEF TPINP DUMMY"
"FILEDEF OUT DISK" dbn "UTL00 A (*dcvsb4028
arguments = ",","|dbn"

"EXECUS UTL00* arguments
ut100returncode = RC
IF ut100returncode = 2000 THEN DO
SAY "UTL00 *" ut100returncode
RETURN ut100returncode
END

"ESTATE UTL00*suffix "LISTING A"
sortstate = RC
IF sortstate = 0 THEN "ERASE UTL00*suffix "LISTING A"

"ESTATE UTL00*suffix "SYSPRINT A"
sortstate = RC
IF sortstate = 0 THEN "ERASE UTL00*suffix "SYSPRINT A"

"FILEDEF = CLEAR"
sortpas dbn "UTL00 A" dbn "SORT A COCSINP SORT *"

ssortreturncode = RC
IF ssortreturncode = 0 THEN DO
SAY "SSORT *" ssortreturncode
RETURN ssortreturncode
END

"ESTATE" dbn "UTL00 A"
ssortate = RC
IF sortstate = 0 THEN "ERASE" dbn "UTL00 A"

"SET CRMSTYPE HR"
"DBL = CLEAR"
"SET CRMSTYPE RT"
suffix = SUBSTR(infrat,1,1)||SUBSTR(dbn,1,2)

"DBL JJSYSTCT" filemode "DSN" cat

VM/CMS ISIS (Mainframe Version)
"DBL MASTER" filemode "DSN" pfx cln "MASTER (VSAM CAT I.JSYSCT)
"DBL MASTER" filemode "DSN" pfx cln "MASTER (VSAM CAT I.JSYSCT)
"DBL MFLOG" filemode "DSN" pfx cln "MFLOG (VSAM CAT I.JSYSCT)
"DBL FD" filemode "DSN" pfx cln "FDYS (VSAM CAT I.JSYSCT)

IF cln = "PREP" THEN DO
  "DBL IFPERI" filemode "DSN" pfx "IFPERI INVF (VSAM CAT I.JSYSCT)"
END

"FILEDEF = CLEAR"
"FILEDEF VSP00 suffix "SYSPRINT A"
"FILEDEF Printer Disk VSP00 suffix "LISTING A"
"FILEDEF LOGMSG DISK VSP00 suffix "LOGMSG A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF LINK DISK" dnb "VSP00 A ("dbolink"
"FILEDEF SORTPOS DISK" dnb "PARR A"
"FILEDEF INPUT DISK" dnb "SORT A"
"FILEDEF UPDATE DUMMY"
"FILEDEF CARDS DUMMY"

stack = "DN=n "dn", INFFRMAT=' "infimat",
stack = stack | "TF=" tf","
stack = stack | "ULC=" ulc", SEQ=' "seq", SXP=' "sxp"
push stack

"EXECDS VSP00"

vsp00returncode = RC
IF vsp00returncode > 16 THEN DO
  SAY "VSP00 " vsp00returncode
RETURN vsp00returncode
END

"ESTATE VSP00 suffix "SYSPRINT A"
rmsaved = RC
IF rmsaved = 0 THEN "ERASE VSP00 suffix "SYSPRINT A"

"ESTATE VSP00 suffix "LOGMSG A"
rmsaved = RC
IF rmsaved = 0 THEN "ERASE VSP00 suffix "LOGMSG A"

"ESTATE" dnn "SORT A"
rmsaved = RC
IF rmsaved = 0 THEN "ERASE" dnn "SORT A"

IF vsp00returncode = > 8 THEN DO
  SAY "no link records written"
VM/CMS ISIS (Mainframe Version)

"SET CMSTYPE HT"
IF gammareturncode = 0 THEN "EXEC DROP "filemode" (Q"
"FILEDEF = CLEAR"
"DBL = CLEAR"
"SET CMSTYPE RT"
RETURN gammareturncode
END

FILEDEF = CLEAR
sortpma dnn "VSP00 A" dnn "LINK A DCBSYLNK SORT A"
ssortreturncode = RC
IF ssortreturncode = 0 THEN DO
  SAY "SSORT " ssortreturncode
RETURN ssortreturncode
END

"ESTATE" dnn "VSP00 A"
rmsaved = RC
IF rmsaved = 0 THEN "ERASE" dnn "VSP00 A"

"SET CMSTYPE HT"
"DBL = CLEAR"
"SET CMSTYPE RT"

suffix = "U" | | SUBSTR(dnn,1,2)

"DBL J.ISYSCT filemode "DSN" cat
"DBL INV" filemode "DSN" pfx cln "INVF (VSAM CAT I.JSYSCT)
"DBL MFLOG" filemode "DSN" pfx cln "MFLOG (VSAM CAT I.JSYSCT)
"DBL MASTER" filemode "DSN" pfx cln "MASTER (VSAM CAT I.JSYSCT)
"DBL MFLOG" filemode "DSN" pfx cln "MFLOG (VSAM CAT I.JSYSCT)

"FILEDEF = CLEAR"
"FILEDEF VSP00 suffix "SYSPRINT A"
"FILEDEF LOGMSG DISK VSP00 suffix "LOGMSG A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF LINK DISK" dnn "LINK A"

stack = "UPD"
push stack

"EXECDS VSP00"

vsp00returncode = RC
IF vsp00returncode = > 8 THEN DO
VM/CMS ISIS (Mainframe Version)
Mainframe VM/CMS ISIS

REXX EXEC

SAV "VSP05 \&" vsp05returncode
RETURN vsp05returncode

END

"ESTATE VSP05" suffix "LISTING A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE VSP05" suffix "LISTING A"

"ESTATE VSP05" suffix "LOGMSG A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE VSP05" suffix "LOGMSG A"

"ESTATE" dbn "LINK A"
linkstate = RC
IF linkstate = 0 THEN "ERASE" dbn "LINK A"

"SET CMSTYPE HT"
IF gisareturncode = 0 THEN "EXEC DROP" filemode "(O"

"FILEDEF = CLEAR"
"DLBL = CLEAR"
"SET CMSTYPE RT"

RETURN 0

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

/* ISICS/CMS - TRANSLATE HL/FILE HEADINGS */
/* ISICS/CMS - HANDLE PARAMETERS FOR CVUTL1 EXEC */

argcln = anyfile cluster name
argcat = vasm catalog name
argcatunit = vasm catalog unit
argcln = cluster name
argdb = data base name
argdcbv4828 = dcs for sort files
argkey = heading (to be translated) number
argnotfrm = translated output file
argprefix = prefix of data sets
argtrf = tsf file/ftvcs cluster name
argsort = sort required
argsortlib = sort txtlib
argsortpgm = name of sort program
argtxtlib = isis txtlib

ADDRESS COMMAND ARG parameters

PARSE SOURCE execname.

PARSE VAR parameters operands (f) options

"EXEC PROLOG" execname "(DBN=HEAD CLN=PERI "options

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* default_supported */

"GLOBAL SELECT EXEC GET PROLOGN1"
"GLOBAL SELECT EXEC GET PROLOGN2"
"GLOBAL SELECT EXEC GET PROLOGP1"
"GLOBAL SELECT EXEC GET PROLOGP2"
"GLOBAL SELECT EXEC GET PROLOGP3"

nextcdef = "VALUE" (PROLOG*P1) "VALUE" (PROLOG*P2) "VALUE" (PROLOG*P3)

PARSE VAR nextcdef "\".nextcdef

VM/CMS ISIS (Mainframe Version)
PARSE VAR pexedef "" pexedef
DO WHILE pexedef ne ""
    PARSE VAR pexedef name "" pexedef
    PARSE VAR pexedef para "" pexedef
    INTERPRET name "= para"
END

fn. = ""
ft. = ""
fs. = ""

PARSE VAR operands fn.1 operands
IF fn.1 = "" THEN fn.1 = "UTL11"
IF LENGTH(fn.1) > 8 THEN DO
    SAY "***** error in" execname "****"
    SAY "***** filename" fn.1 "has more than 8 characters."
    RETURN 16
END
ft.1 = "CARDS"
fs.1 = "*"

"ESTATE fn.1 ft.1 fm.1"
rcsavaed = RC
IF rcsavaed = 0 THEN DO
    SAY "***** error in" execname "****"
    SAY "***** file" fn.1 ft.1 fm.1 "not found."
    RETURN rcsavaed
END

"ESTATE" dbn "PARM ="
rcsavaed = RC
IF rcsavaed = 0 THEN DO
    SAY "***** error in" execname "****"
    SAY "***** file" dbn "PARM = not found."
    RETURN rcsavaed
END

"ESTATE SORT HITSORT ="
rcsavaed = RC
IF rcsavaed = 0 THEN DO
    SAY "***** error in" execname "****"
    SAY "***** file SORT HITSORT = not found."
    RETURN rcsavaed
END

VM/CMS ISIS (Mainframe Version)
*FILEDEF HITSORT DISK SORT HITSORT A (*dcbv4028
*FILEDEF XHITIN DISK VSPB0 XHITIN A (*dcbv4028
*FILEDEF XHITOUT DISK UTL11 XHITOUT A (*dcbv4028

stack = key" notran" 
push stack
"EXECOS UTL11"

util11returncode = RC
IF util11returncode = 0 THEN DO
RETURN util11returncode
END

*ESTATE UTL11 SYSPRT A
rcsaved = RC
IF rcsaved = 0 THEN "ERASE UTL11 SYSPRT A"

*ESTATE HITSORT *
rcsaved = RC
IF rcsaved = 0 THEN "ERASE SORT HITSORT *

*ESTATE VSPB0 XHITIN A
rcsaved = RC
IF rcsaved = 0 THEN "ERASE VSPB0 XHITIN A"

*ESTATE SORT HITMERGE A
rcsaved = RC
IF rcsaved = 0 THEN "ERASE SORT HITMERGE A"

"FILEDEF = CLEAR" 
"FILEDEF SYSPRT DISK" sortpgm "LISTING A"
sortpgm "UTL11 XHITOUT A SORT HITSORT A CDSCLEA SORT A *
sassortreturncode = RC
IF assortreturncode = 0 THEN DO
RETURN assortreturncode
END

*ESTATE* sortpgm "LISTING A"
rcsaved = RC
IF rcsaved = 0 THEN "ERASE* sortpgm "LISTING A"

*ESTATE UTL11 XHITOUT A
rcsaved = RC
IF rcsaved = 0 THEN "ERASE UTL11 XHITOUT A"

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

"SET CMSTYPE HT"
IF pUSERreturncode = 0 THEN "EXEC DROP" filemode "IO"
"FILEDEF = CLEAR"
"DLBL = CLEAR"
"SET CMSTYPE RT"

RETURN util11returncode
/* ISIS/CMS - MASTER FILE EXTRACTION AND RE-FORMATTING (DISK ONLY) */

/* ISIS/CMS - HANDLE PARAMETERS FOR CVUTL13 EXEC */

/ * acin = alternate cluster name */
/ * cat = vSAM catalog name */
/ * catowner = vSAM catalog unit userid */
/ * catunit = vSAM catalog unit */
/ * cin = cluster name */
/ * dbn = data base name */
/ * dcbvbetape = dcb for backup tape */
/ * eles = elec hit file indicator */
/ * frme = from sf number */
/ * lkc = lookup cluster name */
/ * newl = snfr renumbering indicator */
/ * px = prefix of data sets */
/ * sortlib = sort txtlib */
/ * to = to sf number */
/ * txtlib = isis txtlib */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE .. execname ..

PARSE VAR parameters operands "() options

*EXEC PROLOG execname "() options

prologreturncode = RC
IF prologreturncode => 8 THEN RETURN prologreturncode

/* defaults supported */

/*GLOBAL SELECT *EXEC GET PROLOG1*/
/*GLOBAL SELECT *EXEC GET PROLOG2*/
/*GLOBAL SELECT *EXEC GET PROLOG3*/
/*GLOBAL SELECT *EXEC GET PROLOGP1*/
/*GLOBAL SELECT *EXEC GET PROLOGP2*/
/*GLOBAL SELECT *EXEC GET PROLOGP3*/

nexedef = "VALUE*(PROLOG\*N1)" | "VALUE*(PROLOG\*N2)" | "VALUE*(PROLOG\*N3)"
pexedef = "VALUE*(PROLOG\*P1)" | "VALUE*(PROLOG\*P2)" | "VALUE*(PROLOG\*P3)"

PARSE VAR nexedef "\" nexedef
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

PARSE VAR pexedef "\" pexedef

DO WHILE pexedef == ""
PARSE VAR pexedef name "\" pexedef
PARSE VAR pexedef para "\" pexedef
INTERPRET name = para
END

fn. = ""
ft. = ""
f. = ""

PARSE VAR operands fn.1 operands

IF fn.1 = "" THEN fn.1 = "UTL13"
IF LENGTH(fn.1) > 8 THEN DO
SAY "****** error in" execname " ******"
SAY "file name" fn.1 "has more than 8 characters."
RETURN 16
END

rt.1 = "CARDS"
f.1 = "f"

fn.2 = "F\|dbn\|"f8"
ft.2 = "BACKUP"
f.2 = "A"

IF eles == "N" THEN DO
**ESTATE SORT HITSORT **
rcsorted = RC
IF rcsorted => 8 THEN DO
SAY "****** error in" execname " ******"
SAY "file sort HITSORT not found."
RETURN rcsorted
END
END

*GLOBAL TXTLIB PLILIB CMSLIB* txtlib sortlib
*EXEC GIME* catowner catunit *G STACK MODE -P*
gimereturncode = RC
FULL filename .
IF gimereturncode > 4 THEN RETURN gimereturncode

*DLBL IJSYST* filename *DSN* cat
*DLBL MASTER* filename *DSN* px cin MASTER (VSM CAT IJSYST)*
VM/CMS ISIS (Mainframe Version)
/* DLBL MASTER. filemode "DSN" px acin "MASTER (VSM CAT IJYSCT)" */
/* DLBL LOOKUP filemode "DSN" px lcin "LOOKUP (VSM CAT IJYSCT)" */

/* FILEDEF = CLEAR */
/* FILEDEF SYSPRINT DISK UTIL13 LISTING A */
/* FILEDEF CARDS DISK* fn.1 ft.1 fm.1
FILEDEF MAST DISK* fn.2 ft.2 fm.2 */
/* FILEDEF PLIOMPA DISK PLIOMPA LISTING A */
IF elem = "N" THEN DO
   /* FILEDEF HITSORT DISK SORT HITSORT */
END

stack = "ELEM="+elem",DBN="+dbn",FROM="+from",TO="+to",NEW1="+new1;";
push stack

"EXECDS UTIL13"
util13returncode = RC
IF util13returncode > 4 THEN DO
   RETURN util13returncode
END
IF elem = "N" THEN DO
   "ESTATE SORT HITSORT"
   rcsaved = RC
   IF rcsaved = 0 THEN "ERASE SORT HITSORT"
END

*SET CMSTYPE HT*
IF gimereturncode = 0 THEN *EXEC DROP* filemode "I0"
*FILEDEF = CLEAR*
*DLBL = CLEAR*
*SET CMSTYPE RT*

RETURN util13returncode

VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS (Mainframe Version)

/* IS/IS - BATCH RETRIEVAL (SORTED OUTPUT ON HITSORT) */

/* IS/IS - HANDLE PARAMETERS FOR CVVPDB10 EXEC */

/* acin = alternate cluster name */
/* acgin = anyfile cluster name */
/* cat = vsm catalog name */
/* catowner = vsm catalog unit userid */
/* catunit = vsm catalog unit */
/* cin = cluster name */
/* dbn = data base name */
/* dcbvbr28 = dcb for sort files */
/* lcin = lookup cluster name */
/* px = prefix of data sets */
/* sort = sort required */
/* sortlib = sort txtlib */
/* txtlib = isis txtlib */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . execname 

PARSE VAR parameters operands "f" options

*EXEC PROLOC* execname "("options

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */

"GLOBALY SELECT #EXEC GET PROLOG*1"
"GLOBALY SELECT #EXEC GET PROLOG*2"
"GLOBALY SELECT #EXEC GET PROLOG*3"
"GLOBALY SELECT #EXEC GET PROLOG*4"
"GLOBALY SELECT #EXEC GET PROLOG*5"

nexedef = "VALUE" (PROLOG*N1) | "VALUE" (PROLOG*N2) | "VALUE" (PROLOG*N3)

nedef = "VALUE" (PROLOG*P1) | "VALUE" (PROLOG*P2) | "VALUE" (PROLOG*P3)

PARSE VAR nedefdef
PARSE VAR nedefdef "n"
DO WHILE nexedef = ""
  PARSE VAR nexedef name "\" nexedef
  PARSE VAR pexedef parm "\" pexedef
  INTERPRET name "= parm"
END

fn. = ""
ft. = ""
fs. = ""

PARSE VAR operands fn.1 operands

IF fn.1 = "" THEN fn.1 = "YSP01"
IF LENGTH(fn.1) > 8 THEN DO
  SAY "***** error in" execname "*****"
  SAY "***** filename" fn.1 "has more than 8 characters."
  RETURN 16
END

ft.1 = "CARDS"
fs.1 = "*

"ESTATE" fn.1 ft.1 fs.1
rcsaved = RC
IF rcsaved = 0 THEN DO
  SAY "***** error in" execname "*****"
  SAY "***** file" fn.1 ft.1 fs.1 "not found."
  RETURN rcsaved
END

"ESTATE" dbn "PARM m"
rcsaved = RC
IF rcsaved = 0 THEN DO
  SAY "***** error in" execname "*****"
  SAY "***** file" dbn "PARM m not found."
  RETURN rcsaved
END

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib

"EXEC GIME" catowner catunit "# I O STACK MODE -P"
gsierrtnocode = RC
PULL filemode .
IF gsierrtnocode > 4 THEN RETURN gsierrtnocode

"DLBL IJYSCT" filemode "DSN1" cat
"DLBL INVF" filemode "DSN1" pfx cin "INVF (YSAM CAT IJYSCT)"
"DLBL INV1" filemode "DSN1" pfx cln "INVF (YSAM CAT IJYSCT)"

VM/CMS ISIS (Mainframe Version)
/* ISIS/CMS - PRINT SORTED HIT FILE (INPUT ON SORT HITSORT) */

/* ISIS/CMS - HANDLE PARAMETERS FOR CVVSP04 EXEC */

/* acin */
/* ancin */
/* cat */
/* catowner */
/* catunit */
/* cc */
/* cim */
/* cnt */
/* cu */
/* dbn */
/* ind */
/* lckln */
/* lp */
/* lw */
/* mode */
/* nc */
/* nh */
/* nt */
/* pc */
/* pf */
/* pn */
/* pt */
/* rv */
/* skp */
/* skpl */
/* skpl */
/* sortlib */
/* sol */
/* txtlib */
/* zt */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE .. execname ..

PARSE VAR parameters operands "f" options

"EXEC PROLOC" execname "(f" options

VM/CMS ISIS (Mainframe Version)
"ESTATE SORT HITSORT *"
rcsav = RC
IF rcsav = 0 THEN DO
   SAY "***** error in" execname ' ****
   SAY "***** file SORT HITSORT * not found."
RETURN rcsav
END

"GLOBAL TXTLIB PLILIB CMSLIB* txtlib sortlib"

"EXEC GMEXEC catowner catunit * "IQ STACK MODE -P"
gimereturncode = RC
FULL filemode .
IF gimereturncode > 4 THEN RETURN gimereturncode

"DLBL IJSYSCT*" filemode "DSN" cat
"DLBL INVF*" filemode "DSN" px cin "INVF (VSAM CAT IJSYSCT"
"DLBL INVFI*" filemode "DSN" px cix "INVF (VSAM CAT IJSYSCT"
"DLBL MASTER*" filemode "DSN" px cin "MASTER (VSAM CAT IJSYSCT"
"DLBL ANYFILE*" filemode "DSN" px anycin "ANY (VSAM CAT IJSYSCT"
"DLBL LOOKUP*" filemode "DSN" px icin "LOOKUP (VSAM CAT IJSYSCT"

"FILEDEF = CLEAR"
"FILEDEF SORTPOS DISK" ddn "PARN a"
"FILEDEF CARDS DISK" fn.1 ft.1 fm.1
"FILEDEF HITSORT DISK SORT HITSORT *"
"FILEDEF PRINTER DISK VSP04 LISTING A"
"FILEDEF SYSPRINT DISK VSP04 SYSPRINT A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"

stack = iw, "cu", "ip", "nc", "nh", "pt", "mode", "
stack = stack "cc", "pc", "pp", "skp", "ind", "to"
stack = stack "skp1", "cnt", "skp1", "nt", "zt"
stack = stack "rv"
push stack

"EXECOS VSP04"

vsp04returncode = RC
IF vsp04returncode = 0 THEN DO
   RETURN vsp04returncode
END

"ESTATE VSP04 SYSPRINT A"
rcsav = RC
IF rcsav = 0 THEN "ERASE VSP04 SYSPRINT A"

VM/CMS ISIS (Mainframe Version)
/* ISICS - GENERATE ANY FILE (VM/CMS) */
/*
NOTE, THAT THE EXECUTION OF THIS MUST BE PRECEDED BY:
"EXEC DELDEF FILE=ANY CATUNIT=PW=password"
*/
/* ISICS - HANDLE PARAMETERS FOR CVSP06 EXEC */
/*
anycin = anyfile cluster name
/vsac catalog name
/vsac catalog unit userid
/vsac catalog unit
/vsac catalog unit password
/inv,c cluster name
/pfx prefix of data sets
/sortlib = sort library
/textlib = text library
*/
ADDRESS COMMAND
ARG parameters
PARSE SOURCE .. execname ..
PARSE VAR parameters operands /* options
"EXEC PROLOG" execname "/* options
prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode
*/
/* defaults supported */
"GLOBALLY SELECT "EXEC GET PROLOGN1"
"GLOBALLY SELECT "EXEC GET PROLOGN2"
"GLOBALLY SELECT "EXEC GET PROLOGP1"
"GLOBALLY SELECT "EXEC GET PROLOGP2"
"GLOBALLY SELECT "EXEC GET PROLOGP3"
newexecdef = "VALUE"(PROLOGN1) | "VALUE"(PROLOGN2) | "VALUE"(PROLOGN3)
newexecdef = "VALUE"(PROLOGP1) | "VALUE"(PROLOGP2) | "VALUE"(PROLOGP3)
PARSE VAR newexecdef "\"" newexecdef
PARSE VAR newexecdef "\"" newexecdef
DO WHILE newexecdef = ""
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

/* EXEC PROLOG */
/* execname "/* options
prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode
*/
/* defaults supported */
"GLOBALLY SELECT "EXEC GET PROLOGN1"
"GLOBALLY SELECT "EXEC GET PROLOGN2"
"GLOBALLY SELECT "EXEC GET PROLOGP1"
"GLOBALLY SELECT "EXEC GET PROLOGP2"
"GLOBALLY SELECT "EXEC GET PROLOGP3"
newexecdef = "VALUE"(PROLOGN1) | "VALUE"(PROLOGN2) | "VALUE"(PROLOGN3)
newexecdef = "VALUE"(PROLOGP1) | "VALUE"(PROLOGP2) | "VALUE"(PROLOGP3)
PARSE VAR newexecdef "\"" newexecdef
PARSE VAR newexecdef "\"" newexecdef
DO WHILE newexecdef = ""
VM/CMS ISIS (Mainframe Version)
"FILEDEF = CLEAR"
if ft.t = "CARDS" then "FILEDEF CARDS DISK" fn.t ft.t fn.t
else "FILEDEF HITFILE DISK" fn.t ft.t fn.t
"FILEDEF SYSPRT DISK VSP06 LISTING A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"

stack = *
push stack

"EXECOS VSP06"

vsp06-returncode = RC
if vsp06-returncode = 0 then do
    return vsp06-returncode
end

"SET CMSTYPE HT"
if gtmreturncode = 0 then "EXEC DROP" filecode "Q"
"FILEDEF = CLEAR"
"DUBL = CLEAR"
"SET CMSTYPE RI"

VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS

/* ISIS/CMS - MASTER FILE BACKUP/RESTORE UTILIY (DISK ONLY) */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVYSP06 EXEC */

#define cat _ vasm catalog name
#define catowner _ vasm catalog unit user id
#define cattrun _ vasm catalog unit
#define cattrunpw _ vasm catalog unit password
#define cilm _ cluster name
#define dbn _ data base name
#define dcbvtape _ dcb for bkup tape
#define from _ from mf number
#define func _ function
#define igen _ log generation
#define nextno _ next mf number
#define pfx _ prefix of data sets
#define sortlib _ sort txtlib
#define to _ to mf number
#define txtlib _ isis txtlib

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . execname .
PARSE VAR parameters operands "f" options

"EXEC PROLOG execname "(FUNC=UNLK" options
prologreturncode = RC
if prologreturncode = 0 then return prologreturncode

/* defaults_supported */

"GLOBALY SELECT #EXEC GET PROLOG1"
"GLOBALY SELECT #EXEC GET PROLOG2"
"GLOBALY SELECT #EXEC GET PROLOG3"
"GLOBALY SELECT #EXEC GET PROLOGP1"
"GLOBALY SELECT #EXEC GET PROLOGP2"
"GLOBALY SELECT #EXEC GET PROLOGP3"
nexedef = "VALUE" (PROLOG"N1") | "VALUE" (PROLOG"N2") | "VALUE" (PROLOG"N3")
pexedef = "VALUE" (PROLOG"P1") | "VALUE" (PROLOG"P2") | "VALUE" (PROLOG"P3")

PARSE VAR nexedef "f" pexedef

VM/CMS ISIS (Mainframe Version)
PARSE VAR pexecdef "\" pexecdef
DO WHILE pexecdef == ""
    PARSE VAR pexecdef name "\" pexecdef
    PARSE VAR pexecdef para "\" pexecdef
    INTERPRET name = para
END
fn. = ""
ft. = ""
fms. = ""

PARSE VAR operands fa.1 operands

IF func = "JHKL" THEN DO
    IF fa.1 == "" THEN DO
        ft.1 = "BACKUP"
        IF fn.1 = "MF"||cln."88"
            ELSE fn.1 = "MF"||dnbn."88"
    END
END ELSE DO
gen.1 = "YSPQ8"
    ft.1 = "CARDS"
    IF fa.1 == "" THEN fa.1 = "s"

END

IF fa.1 == "" THEN DO
    IF LENGTH(fn.1) > 1 THEN DO
        SAY "****** error in" execname "******"
        SAY "****** filename" fn.1="has more than 1 character."
        RETURN 16
    END

    "ESTATE" fn.1 ft.1 fa.1
    rcsaved = RC
    IF rcsaved == 0 THEN DO
        SAY "****** error in" execname "******"
        SAY "****** file" fn.1 ft.1 fa.1 "not found."
        RETURN rcsaved
    END
END

"GLOBAL TXTLIB PLIB CMSLIB" txtlib sortlib

IF catunitpw = "" THEN DO
    "EXEC GIME" catowner catunit "* IQ STACK MODE -P"
    gimereturncode = RC
END

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXECs

PULL filemode.
IF gimereturncode > 4 THEN RETURN gimereturncode
END ELSE DO
    "EXEC GIME" catowner catunit "*" catunitpw "* IQ MR STACK MODE -P"
    gimereturncode = RC
    PULL filemode.
    IF gimereturncode > 4 THEN RETURN gimereturncode
END

"DBLJ LJSYCT" filemode "DSN" cat
"FILEDEF = CLEAR"

IF func = "CHKL" THEN DO
    "DBLJ MASTER" filemode "DSN" pfx cln "MASTER" (YSAM CAT LJSYCT)
    "DBLJ MFLOG" filemode "DSN" pfx cln "MFLOG" (YSAM CAT LJSYCT)
    IF fn.1 == "" THEN "FILEDEF MFBUKUP DISK" fn.1 ft.1 fa.1 *(dcdbvbtape
    ELSE "FILEDEF MFBUKUP DUMMY"
END ELSE DO
    "DBLJ BOOK" filemode "DSN" pfx "BOOK MASTER" (YSAM CAT LJSYCT)
    "DBLJ PERI" filemode "DSN" pfx "PERI MASTER" (YSAM CAT LJSYCT)
    "DBLJ PREP" filemode "DSN" pfx "PREP MASTER" (YSAM CAT LJSYCT)
    "FILEDEF CARDS DISK" fn.1 ft.1 fa.1 *
    "FILEDEF MFBUKUP DUMMY"
END

IF dbn. == "" THEN suffix = SUBSTR(func,1,1)||SUBSTR(cln,1,2)
ELSE suffix = SUBSTR(func,1,1)||SUBSTR(dbn.1,2)

"FILEDEF SYSPRINT DISK VSPQR suffix "LISTING A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
"FILEDEF LOGMSG DISK VSPQR suffix "LOGMSG A"

stack = "FUNC","FUN","Dnbn","Dnbn","Cln."
       "Frp","Frp","Dnbn","Dnbn","Dnbn."
       push stack

"EXEC" VSPQR

IF vspqr=RETURNcode = RC
    RETURN vspqr=RETURNcode
END

"SET CMSTYPE HT"
IF gimereturncode = 0 THEN "EXEC DROP" filemode "IQ"

VM/CMS ISIS (Mainframe Version)
"FILEDEF * CLEAR"
"DGL = CLEAR"
"SET CMSTYPE RT"

RETURN vspb8returncode

/* ISIS/CMS - ISIS/ISO - ISO/ISIS FORMAT CONVERSION (DISK ONLY) */

/* ISIS/CMS - HANDLE PARAMETERS FOR CVVSP18 EXEC */

/* cat = vsam catalog name */
/* catowner = vsam catalog unit userid */
/* catunit = vsam catalog unit */
/* catunitpw = vsam catalog unit password */
/* cc = iso tape code */
/* clin = cluster name */
/* dbn = data base name */
/* dcbvbtape = dcb for bkp tape */
/* dcbvbtape2 = dcb for htp file */
/* elms = elms/hit file indicator */
/* from = from af number */
/* func = function import/export */
/* id = length of subfield ident in leader */
/* ini = length of field ident in leader */
/* ipl = length of implicit field in directory */
/* ifi = length of length field in directory */
/* icln = lookup cluster name */
/* ipl = length of offset field in directory */
/* m = mfn renumbering indicator */
/* out = isis output file (master/sequential) */
/* pf = prefix of data sets */
/* sortlib = sort txtlib */
/* to = to af number */
/* txtlib = isis txtlib */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . . execname .

PARSE YAR parameters operands "(" options

"EXEC PROLOG" execname "CC-E FUNC=EXPORT" options

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */

"GLOBALLY SELECT #EXEC GET PROLOG1"*
"GLOBALLY SELECT #EXEC GET PROLOG2"

VM/CMS ISIS (Mainframe Version)
"GLOBAL SELECT #EXEC GET PROLOG1P"
"GLOBAL SELECT #EXEC GET PROLOG2P"
"GLOBAL SELECT #EXEC GET PROLOG3P"

nexecdef = "VALUE" (PROLOG"1") | "VALUE" (PROLOG"2") | "VALUE" (PROLOG"3")
pexecdef = "VALUE" (PROLOG"P1") | "VALUE" (PROLOG"P2") | "VALUE" (PROLOG"P3")

PARSE VAR nexecdef "\"" nexecdef
PARSE VAR pexecdef "\"" pexecdef

DO WHILE nexecdef = ""
    PARSE VAR nexecdef name "\"" nexecdef
    PARSE VAR pexecdef para "\"" pexecdef
    INTERPRET name "= para"
END

fn. = **
ft. = **
fs. = **

PARSE VAR operands fn.1 operands

IF fn.1 = "" THEN fn.1 = "VSP1B"
IF LENGTH(fn.1) > 8 THEN DO
    SAY "***** error in" execname "*****"
    SAY "***** filename" fn.1 "has more than 8 characters."
    RETURN 16
END

ft.1 = "CARDS"
fn.1 = "*

Estate" fn.1 ft.1 fn.1
cardreturncode = RC

fn.2 = dbn
ft.2 = "ISO"
fs.2 = "A"

stack = "CC="cc",DBN="dbn",FUNC="func",NEW="new1,"

SELECT
    WHEN func = "IMPORT" THEN DO
        stack = stack["OUT="out""]
    END
    WHEN func = "EXPORT" THEN DO
        stack = stack["ELEM="elem",FROM="from",TO="to",""
        stack = stack["IPL="ipl",LPL="lpl"]
    END
END

VM/CMS ISIS (Mainframe Version)

otherwise
    SAY "***** error in" execname "*****"
    SAY "***** unsupported function " func
    RETURN 16
END

IF LENGTH(stack) > 100 THEN DO
    SAY "***** error in" execname "*****"
    SAY "***** length of parameter string exceeds 100 characters"
    RETURN 16
END

"GLOBAL TTB LIB CMSLIB" txtlib sortlib

IF catunitpu = "" THEN DO
    "EXEC GIME" catowner catunitpu "# IQ STACK MODE -P"
    gimereturncode = RC
    PULL filename.
    IF gimereturncode > 4 THEN RETURN gimereturncode
END
ELSE DO
    "EXEC GIME" catowner catunitpu "# IQ MR STACK MODE -P"
    gimereturncode = RC
    PULL filename.
    IF gimereturncode > 4 THEN RETURN gimereturncode
END

"DLBL IJSYSCT" filename "DSN" cat
"DLBL MASTER" filename "DSN" pxf cln "MASTER (YSAM CAT IJSYSCT)"
"DLBL LOOKUP" filename "DSN" pxf cln "LOOKUP (YSAM CAT IJSYSCT)"

"FILEDEF = CLEAR"
"FILEDEF SPRINT DISK VSP1B LISTING A"
"FILEDEF MAST DISK" fn.2 ft.2 fs.2 "("dcbvotape"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"
    IF else = "N" THEN "FILEDEF TAPE DISK SORT HITSORT = "dcbvb4R028"
    ELSE "FILEDEF TAPE DUMMY"
IF cardreturncode = 0 THEN "FILEDEF CARDS DISK" fn.1 ft.1 fn.1

push stack

"EXECUTS VSP1B"

vsp10=returncode = RC
    IF vsp10returncode = 0 THEN DO
        RETURN vsp10returncode
END

VM/CMS ISIS (Mainframe Version)
IF elen = "N" THEN DO
   "ESTATE SORT HITSORT *"
   rcsaved = RC
   IF rcsaved = 0 THEN "ERASE SORT HITSORT *"
END

"SET CMSTYPE HT"
IF gimereturncode = 0 THEN "EXEC DROP" filemode "(Q"
"FILEDEF = CLEAR"
"DLBL = CLEAR"
"SET CMSTYPE RT"
RETURN vsp1&returncode

/* ISIS/CMS - INVERTED FILE BACKUP/RESTORE UTILITY (DISK VERSION) */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVVSP35 EXEC */
/* cat = vsm catalog name */
/* catowner = vsm catalog unit userid */
/* catunit = vsm catalog unit */
/* catunitpw = vsm catalog unit password */
/* cin = cluster name */
/* dbn = data base name */
/* dbvtape = dcb for bkup tape */
/* func = function (here dump) */
/* igen = log generation */
/* prefix = prefix of data sets */
/* sortlib = sort txtlib */
/* txtlib = isis txtlib */
ADDRESS COMMAND
ARG parameters
PARSE SOURCE . . exename .
PARSE VAR parameters operands "(" options
"EXEC PROLOGC exename "(options
prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode
/* defaults_supported */
"GLOBALLY SELECT $EXEC GET PROLOGN1";
"GLOBALLY SELECT $EXEC GET PROLOGN2";
"GLOBALLY SELECT $EXEC GET PROLOGP1";
"GLOBALLY SELECT $EXEC GET PROLOGP2";
"GLOBALLY SELECT $EXEC GET PROLOGP3";

newedef = "VALUE (PROLOG*N1)"|"VALUE (PROLOG*N2)"|"VALUE (PROLOG*N3)"
 peakedef = "VALUE (PROLOG*P1)"|"VALUE (PROLOG*P2)"|"VALUE (PROLOG*P3)"
PARSE VAR newedef ";" newedef
PARSE VAR peakedef ";" peakedef
DO WHILE newedef = ""

/* ISIS/CMS (Mainframe Version) */
PARSE VAR nexecdef name "\" nexecdef
PARSE VAR pexecdef param "\" pexecdef
INTERPRET name = "para"
END

fn. = ""
ft. = ""
fs. = ""

PARSE VAR operands fn.1 operands

IF fn.1 = "" THEN DO
IF LENGTH(fn.1) > 1 THEN DO
  SAY "...... error in" execname "......"
  SAY "...... filename" fn.1 "has more than 1 character."
  RETURN 16
END
ft.1 = "BACKUP"
fn.1 = ["IF"||cin||"BB"
END

"GLOBAL TXTLIB PLILIB CMSLIB" txtlib sortlib

IF catunitsp = "" THEN DO
  "EXEC GIME" catowner catunit "iQ STACK MODE -P"
gimereturncode = RC
  PULL filename.
  IF gimereturncode > 4 THEN RETURN gimereturncode
END
ELSE DO
  "EXEC GIME" catowner catunit "iQ MR STACK MODE -P"
gimereturncode = RC
  PULL filename.
  IF gimereturncode > 4 THEN RETURN gimereturncode
END

"DLBL 1JSYSTC" filename "DSN" cat
"DLBL INVFS" filename "DSN" prefix cin "INVFS (VSAM CAT 1JSYSTC)"
"DLBL IFLOG" filename "DSN" prefix cin "IFLOG (VSAM CAT 1JSYSTC)"

"FILEDEF = CLEAR"
IF fn.1 = "" THEN "FILEDEF IFBKUP DISK" fn.1 ft.1 fn.1 1("dcbvbtape"
ELSE "FILEDEF IFBKUP DUMMY"
IF dbn = "" THEN suffix = SUBSTR(func1,1,1)||SUBSTR(cin,1,2)
ELSE suffix = SUBSTR(func1,1,1)||SUBSTR(dbn,1,2)

VM/CMS ISIS (Mainframe Version)

"FILEDEF IFLIST DISK VSP35 LISTING A"
"FILEDEF SYSPRINT DISK VSP35 suffix "LISTING A"
"FILEDEF LOGMSG DISK VSP35 suffix "LOGMSG A"
"FILEDEF PLDUMP DISK PLDUMP LISTING A"
stack = "FUNC"||func",DBN"||"dbn",CLN"||"cin",GEN="||igen";"
push stack

"EXEC VSP35"

returncode = RC
IF returncode > 4 THEN DO
  RETURN returncode
END

"SET CMSTYPE HT"
IF returncode = 0 THEN "EXEC DROP" filename ":(Q"
"FILEDEF = CLEAR"
"DLBL = CLEAR"
"SET CMSTYPE HT"

RETURN returncode
/* ISIS/CMS - LOAD/UPDATE VSAM FDT (VM/CMS) */
/* ISIS/CMS - HANDLE PARAMETERS FOR CVSP38 EXEC */

/* cat = vsam catalog name */
/* catowner = vsam catalog unit user id */
/* catunit = vsam catalog unit */
/* catunitpu = vsam catalog unit password */
/* func = function */
/* ptx = prefix of data sets */
/* scin = trfile/fdvs cluster name */
/* sortlib = sort txtlib */
/* txtlib = isis txtlib */

ADDRESS COMMAND

ARG parameters

PARSE SOURCE . execname .

PARSE VAR parameters operands "(" options

"EXEC PROLOG" execname "(FUNC=CHECK" options

prologreturncode = RC
IF prologreturncode = 0 THEN RETURN prologreturncode

/* defaults_supported */

"GLOBAL SELECT eEXEC GET PROLOG1" "GLOBAL SELECT eEXEC GET PROLOG2" "GLOBAL SELECT eEXEC GET PROLOG3" "GLOBAL SELECT eEXEC GET PROLOG4" "GLOBAL SELECT eEXEC GET PROLOG5" "GLOBAL SELECT eEXEC GET PROLOG6"

nexedef = "VALUE" (PROLOG*N1) "VALUE" (PROLOG*N2) "VALUE" (PROLOG*N3)

nexedef = "VALUE" (PROLOG*N1) "VALUE" (PROLOG*N2) "VALUE" (PROLOG*N3)

PARSE VAR nexedef "\"" nexedef
PARSE VAR nexedef "\"" nexedef
DO WHILE nexedef = ""
PARSE VAR nexedef name "\"" nexedef
PARSE VAR nexedef parem "\"" nexedef
INTERPRET name = parem

VM/CMS ISIS (Mainframe Version)

END
fn. = ""
ft. = ""
fs. = ""

index = 1
DO WHILE operands = ""
PARSE VAR operands fn.index operands
IF LENGTH(fn.index) > 8 THEN DO
  SAY "******** error in" execname "*****"
  SAY "******** filename" fn.index "has more than 8 characters."
  RETURN 16
END
index = index + 1
END

index = index - 1
IF fn.1 = "" THEN DO
  fn.1 = "BOOK"
  ft.1 = "FDT"
  fs.1 = "e"
  "ESTATE" fn.1 ft.1 fs.1
  rcsaved = RC
  IF rcsaved = 0 THEN DO
    SAY "******** error in" execname "*****"
    SAY "******** file" fn.1 ft.1 fs.1 "not found."
    RETURN rcsaved
  END
  index = 1
END
END
ELSE DO index = 1 TO index
  ft.index = "FDT"
  fs.index = "e"
  "ESTATE" fn.index ft.index fs.index
  rcsaved = RC
  IF rcsaved = 0 THEN DO
    SAY "******** error in" execname "*****"
    SAY "******** file" fn.index ft.index fs.index "not found."
    RETURN rcsaved
  END
END

suffix = SUBSTR(func,1,1)
"GLOBAL TXTLIB PLLIB CMSLIB* txtlib sortlib"

VM/CMS ISIS (Mainframe Version)
"EXEC GIME" catowner catunit "*" catunitpw "Q MR STACK MODE -P"
gimereturncode = RC
FULL filemode.
IF gimereturncode > 4 THEN RETURN gimereturncode

"DLBL 1JSYSC" filemode "DSN" cat
"DLBL FD1" filemode "DSN" pfx scin "FDYS VSAM CAT 1JSYSC"

"FILEDEF = CLEAR"
"FILEDEF SEQ DISK" fn.1 ft.1 fs.1
IF endex > 1 THEN DO
  DO index = 2 TO endex
    "FILEDEF SEQ DISK" fn.index ft.index fs.index "(APPEND)"
  END
END

"FILEDEF SYSPRINT DISK VSP38"suffix "LISTING A"
"FILEDEF PLIDUMP DISK PLIDUMP LISTING A"

stack = func
push stack

"EXECOS VSP38"

vsp38returncode = RC
IF vsp38returncode = 0 THEN DO
  RETURN vsp38returncode
END

"SET CMSTYPE HT"
IF gimereturncode = 0 THEN "EXEC DROP" filemode "Q"
"FILEDEF = CLEAR"
"DLBL = CLEAR"
"SET CMSTYPE RT"

RETURN vsp38returncode

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

PROLOG 178

/* ISIS/CMS PROLOG ROUTINE TO CHECK INCOMING PARAMETERS */

ADDRESS COMMAND

ARG parameters

length. = 0
length.8 = 1
length.12 = 1
length.13 = 3
length.27 = 1
length.29 = 1
length.36 = 2
length.38 = 1
length.40 = 2
length.45 = 1
length.48 = 4
length.50 = 1
length.53 = 3
length.55 = 3
length.56 = 1
length.57 = 1
length.58 = 1
length.61 = 1
length.63 = 1
length.64 = 2
length.65 = 1
length.66 = 1
length.68 = 4
length.69 = 1
length.70 = 1
length.71 = 1
length.75 = 1
length.78 = 1
length.79 = 1
length.83 = 1
length.85 = 1
length.91 = 2
length.93 = 1
length.96 = 1

param. = **
param.1 = "PERI"
param.2 = "8"
param.3 = "BOOK"

VM/CMS ISIS (Mainframe Version)
name. = "**" 
name.1 = "AELN" 
name.2 = "ADU" 
name.3 = "ANYCLN" 
name.4 = "CAT" 
name.5 = "CATOUMER" 
name.6 = "CATUNIT" 
name.7 = "CATUNITP" 
name.8 = "CC" 
name.9 = "CLN" 
name.10 = "CNOT" 
name.11 = "CREATE" 
name.12 = "CTLE" 
name.13 = "CT" 
name.14 = "DBN" 
name.15 = "DCBPOOS" 
name.16 = "DCBFBOO" 
name.17 = "DCBNUICK" 
name.18 = "DCBUNK" 
name.19 = "DCB31ST" 
name.20 = "DCBPH" 
name.21 = "DCBPHTO" 
name.22 = "DCBPHSEX" 
name.23 = "DCBPHYTEX" 
name.24 = "DCB31APE" 
name.25 = "DCB31B28" 
name.26 = "DL" 
name.27 = "DLM" 
name.28 = "DP" 
name.29 = "ELEM" 
name.30 = "EXT" 
name.31 = "FDF" 
name.32 = "FILE" 
name.33 = "FHT" 
name.34 = "FRT" 
name.35 = "FUC" 
name.36 = "GEN" 
name.37 = "HCVL" 
name.38 = "HIT" 
name.39 = "IDL" 
name.40 = "IND" 
name.41 = "INDEX" 
name.42 = "INL" 
name.43 = "INFRT" 
name.44 = "IPL" 
name.45 = "KEY" 
name.46 = "KL" 

VM/CMS ISIS (Mainframe Version)

name.47 = "EP" 
name.48 = "LEN" 
name.49 = "LFL" 
name.50 = "LOGEN" 
name.51 = "LCLN" 
name.52 = "LL" 
name.53 = "LP" 
name.54 = "LPL" 
name.55 = "LU" 
name.56 = "MODE" 
name.57 = "NC" 
name.58 = "NEMI" 
name.59 = "NEXND" 
name.60 = "NH" 
name.61 = "NTRANS" 
name.62 = "NS" 
name.63 = "NT" 
name.64 = "OLDGEN" 
name.65 = "OUT" 
name.66 = "PC" 
name.67 = "PDF" 
name.68 = "PM" 
name.69 = "PROP" 
name.70 = "PSTE" 
name.71 = "P" 
name.72 = "PYR" 
name.73 = "RECS" 
name.74 = "RL" 
name.75 = "RV" 
name.76 = "SCLN" 
name.77 = "SCL" 
name.78 = "SEQ" 
name.79 = "SIDC" 
name.80 = "SKE" 
name.81 = "SKLP" 
name.82 = "SKPA" 
name.83 = "SORT" 
name.84 = "SORTLIB" 
name.85 = "SORTTN" 
name.86 = "SPLITS" 
name.87 = "SY" 
name.88 = "SYS" 
name.89 = "TF" 
name.90 = "1D" 
name.91 = "1DL" 
name.92 = "TXTLIB" 
name.93 = "ULC" 

VM/CMS ISIS (Mainframe Version)
name.34 = "UTYP"
name.35 = "XEROX"
name.36 = "ZT"

iname = 96

/* defaults supported */
PARSE SOURCE .. execname ..
PARSE VAR parameters operands "(" parameters
PARSE VAR operands caller operands
IF caller = "" THEN DO
  PARSE SOURCE .. execname ..
  caller = execname
END
IF caller = "COUTL02" THEN length.s3 = 8
IF caller = "CVPERM" THEN length.s5 = 8
head. = ""
brace. = ""
index = 1
DO FOREVER
  PARSE VAR parameters head.index *** brace.index *** parameters
  IF parameters = "" THEN LEAVE
  ELSE index = index + 1
END

ndex = 0
DO kndex = 1 TO index
  coma = INDEX(head.kndex,"\n")
  DO WHILE coma = 1
    head.kndex = SUBSTR(head.kndex,2)
    coma = INDEX(head.kndex,"\n")
  END
  IF coma = 8 THEN equal = INDEX(head.kndex,"\n")
  ELSE equal = INDEX(SUBSTR(head.kndex,1,coma),"\n")
  IF equal = 8 THEN DO
    IF INDEX(param.kndex,"\n") = 0 THEN DO
      SAY "**** error in" caller " ****
      SAY "**** no """" found ****
      SAY "**** arguments: " head.kndex "****
      RETURN 16
    END
  END
END

VM/CMS ISIS (Mainframe Version)

END
ELSE DO
  IF coma = 0 THEN DO
    PARSE VAR chunk intermediate ":*" chunk
    IF intermediate = "" THEN LEAVE
  DO FOREVER
    IF intermediate = "" THEN LEAVE
    PARSE VAR intermediate cparm intermediate
    IF index = 1 TO iname
      IF name. = cparm THEN DO
        IF cparm = "" THEN DO
          param. = cparm
          IF param. = "" THEN param. = ""
          IF length. = 0 THEN DO
            SAY "**** parameter is too ",
            "long and will be truncated ****
            SAY "**** name.index ""(length.",
              "\n") = "param. "****" param. =
            SUBSTR(param.",1,length.")
        END
        ELSE DO
          SAY "**** error in" caller ,
          " ****
          SAY "**** parameter is too ",
          "short ****
          SAY "**** name.index ""(length.",
            "\n") = ""param. "****
        END
      END
    END
  END
ELSE DO
  IF brace.kndex = "" THEN DO
    param. = brace.kndex
  ELSE DO
...
VM/CMS ISIS (Mainframe Version)
REXX EXEC

END
END
END
IF index > iname THEN DO
  SAY "***** error in" caller *       ****
  SAY "***** parameter name not found in list *****
  SAY "***** name: " cnam "*****
RETURN 16
END
END
ELSE DO
  /s an = has been found, but no comma */
  chunk = head.kndex
  intermediate = head.kndex
  IF intermediate = "" THEN LEAVE
DO FOREVER
  IF intermediate = "" THEN LEAVE
  PARSE VAR intermediate cnam ="" cpar "" intermediate
  DO index = 1 TO iname
  IF cpar = "" THEN DO
    param.index = cnam
    IF param.index = "" | param.index = "" THEN ,
    param.index = **
    IF length.index = 0 THEN DO
      IF LENGTH(param.index) > length.index THEN DO
        SAY "***** warning in" caller ,
        "*****
        SAY "***** parameter is too long ",
        "and will be truncated *****
        SAY "***** name.index "("length.index,
        ") = "param.index "*****
        param.index = SUBSTR(param.index,1,length.index)
      END ELSE DO
        SAY "***** error in" caller ,
        "*****
        SAY "***** parameter is ",
        "too short *****
        SAY "***** name.index "("length.index,
        ") = "param.index "*****
      RETURN 16
VM/CMS ISIS (Mainframe Version)
Mainframe VM/CMS ISIS
REXX EXEC

SAY "** VSP09 CARDS * not found **"
RETURN 16
END

DD idummy = 1

MAKEBUF
buffno = RC
*EXECIO = DISKR VSP09 CARDS = 1 (FIFO)
inpsize = queued()
oldindex = 1

DO index = 1 TO inpsize
   PARSE PULL clin.index +8 +1 dbn.index +8 .
   dbn.index = STRIP(dbn.index,BI)
   clin.index = STRIP(clin.index,BI)
   IF param.14 = "" THEN DO
      IF clin.index = "" THEN clin.index = clin.oldindex
      ELSE oldindex = index
      IF dbn.index = param.14 THEN DO
         IF clin.index = param.9 THEN DO
            "DROPBUF" buffno
            LEAVE idummy
         END ELSE DO
            SAY "** warning in" caller " **"
            SAY "** inconsistent dbn ("param.14") / clin ("param.9") **"
            SAY "** clin set to" clin.index "**"
            param.9 = clin.index
            "DROPBUF" buffno
            LEAVE idummy
         END
      END
   END
   IF param.14 = "" THEN DO
      SAY "** error in" caller " **"
      SAY "** data base :" param.14 "not found **"
   END
   END
   END

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS
REXX EXEC

RETURN 16
END
ELSE DO
   SAY "** error in" caller " **"
   SAY "** cluster :" param.9 "not found **"
RETURN 16
END

DO index = 1 TO iname
   INTERPRET name.index = " *** param.index ***"
END

param = "\nname = "\n
DO index = 1 TO iname
   param = pars|name.index"\n   name = name|name.index"\nEND

param1 = SUBSTR(param,1,250)
param2 = SUBSTR(param,251,250)
param3 = SUBSTR(param,501)
name1 = SUBSTR(name,1,250)
name2 = SUBSTR(name,251,250)
name3 = SUBSTR(name,501)

"GLOBAL SELECT #EXEC SETLP" execname="P1" param1
"GLOBAL SELECT #EXEC SETLP" execname="P2" param2
"GLOBAL SELECT #EXEC SETLP" execname="P3" param3
"GLOBAL SELECT #EXEC SETLP" execname="N1" name1
"GLOBAL SELECT #EXEC SETLP" execname="N2" name2
"GLOBAL SELECT #EXEC SETLP" execname="N3" name3

IF caller = execname THEN DO
"GLOBAL SELECT #EXEC GET PROLOGN1" execname="P1" param1
"GLOBAL SELECT #EXEC GET PROLOGN2" execname="P2" param2
"GLOBAL SELECT #EXEC GET PROLOGN3" execname="P3" param3
"GLOBAL SELECT #EXEC GET PROLOGP1" execname="P1" param1
"GLOBAL SELECT #EXEC GET PROLOGP2" execname="P2" param2
"GLOBAL SELECT #EXEC GET PROLOGP3" execname="P3" param3

VM/CMS ISIS (Mainframe Version)
param. = ""
name. = ""

PARSE VAR nexecdef "\" nexecdef
PARSE VAR pexecdef "\" pexecdef

iname = 1
DO WHILE nexecdef = ""
    PARSE VAR nexecdef name "\" nexecdef
    PARSE VAR pexecdef name "\" pexecdef
    name.iname = name
    par.sname.iname = pars
    INTERPRET "boo = "name.iname
    aea = LEFT(name.iname,9)
    baa = LEFT(baa,35)
    SAY aea baa "length.iname "= length.iname
    iname = iname + 1
END
RETURN 0

VM/CMS ISIS (Mainframe Version)
IF operands == "" THEN DO
  PARSE VAR operands xdatabase xquery
END

CALL initisis

DO WHILE action == "C"
dvalid = 0
DO WHILE dvalid = 0 | Isso == "HELP"
  xquery = ""
  "TOS3270 ISIS PANEL ;SCREEN ;ENDSCR 1(801)"
  panelreturncode = RC
  message = ""
  IF panelreturncode >= 4 THEN CALL leaveisis
  IF Isso == "EXIT" THEN CALL leaveisis
  CALL selectsdb
  IF Isso == "HELP" THEN DO
    CALL accesshelp 1
    Isso == "HELP"
  END
END

CALL welcome

IF (dbase == "PREP" | dbase == "BOOK") & Isso == "EXE" THEN DO
  CALL order "E" dbase
  IF Isso == "EXIT" THEN CALL leaveisis
  Isso == "HELP"
END

END

action == "R"
DO WHILE action == "R"
  IF mode == "A" THEN CALL formulatequery
  IF action == "C" THEN DO
    CALL readquery
  END
  IF action == "C" THEN CALL execute
END

/# subroutine to initialize ISIS /
initisis:

  text == ""
  maxcount = 20

VM/CMS ISIS (Mainframe Version)

st. == ""
op. == ""
limits == ""
query == ""

IF xquery == "" THEN DO
  iquery == ""
  mode == xmode
END
ELSE DO
  iquery == xquery
  mode == "A"
END

savequery == ""
tempquery == ""
suffix == ""

query == ""

readnews = xreadnews
IF xdatabase == "" THEN DO
  database == "PREP"
  xindicator == ""
END
ELSE DO
  database == xdatabase
  xindicator == "K"
END

IF xls == "" THEN lw == "864" ELSE lw == xls
IF xcw == "" THEN cw == "864" ELSE cw == xcw
IF xlp == "" THEN lp == "864" ELSE lp == xlp
IF xnh == "" THEN nh == "0" ELSE nh == xnh
IF xtol == "" THEN tol == "07" ELSE tol == xtol
IF xnc == "" THEN nc == "1" ELSE nc == xnc
IF xind == "" THEN ind == "84" ELSE ind == xind
IF xnt == "" THEN nt == "1" ELSE nt == xnt

message == ""
call selectsdb

sterror == 0
operror == 0
opvalid == 1
stvalid == 1

VM/CMS ISIS (Mainframe Version)
action = "$C$

RETURN

/* subroutine to select the required database */
selectdb:
dbvalid = 1
database = SUBSTR(database,1,4)
call quit? database
database = RESULT

IF database = dBASE THEN DO
  IF database = 1 THEN dBASE = database
ELSE database = dBASE

"ESTATE VSP89 CARDS *"
rcsaved = RC

IF rcsaved = 0 THEN DO
  "EXECIO 1 EMSG STRING CNBISISM6E ISIS file VSP89 CARDS A", "not found."
call exit 12
END

"EXECIO DISKR VSP89 CARDS * 1 (FIFO"
inputsize = queued()
oldindex = 1

cname = "$"
DO index = 1 TO inputsize
  PARSE FULL cname,index +8 +1 dBASE.index +8,
  dBASE.index = STRIP(dBASE.index,8)
  cname,index = STRIP(cname,index,8)
  IF cname.index = 2 THEN cname.index = NAME.index
  ELSE oldindex = index
  IF dBASE.index = dBASE THEN DO
    cname = cname,index
  END
END

IF cname = "" THEN DO
  message = "** The database "database" does not exist."
database = "PREP"
dBASE = "PREP"
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXECS

VM/CMS ISIS (Mainframe Version)

SELECT:
  WHEN dBASE = "CONF" THEN DO
    maxpage = 27
  END
  WHEN dBASE = "BOOK" THEN DO
    maxpage = 28
  END
  WHEN dBASE = "PREP" THEN DO
    maxpage = 30
  END
  OTHERWISE NOP
END

printformat = SUBSTR(dBASE,1,2)||"01"
sortformat = "X9US"
limits = ""
text = ""
saytext = ""

VM/CMS ISIS (Mainframe Version)

REXX EXECS

/* subroutine to determine a query from search terms and operators */
formulatequery:
eeq = 0
firttime = 0
opcount = 1
leftparentheses = 0
rightparentheses = 0
query = ""

/* enter and verify user's query */
DO WHILE eq = 0 & opcount < maxcount
  DO FOREVER
    stvalid = 0
    opvalid = 0
    DO WHILE opvalid = 0 | stvalid = 0
      IF opcount > 1 THEN DO
        CALL insertleftpar
        query1 = SUBSTR(tempquery,1,79)
        IF LENGTH(tempquery) > 88 THEN query2 = SUBSTR(tempquery,88,79)
        ELSE query2 = ""
      END
    END
    stvalid = 1
  END
  opvalid = 1
  WHEN opvalid = 0 | stvalid = 0
  query = ""
IF LENGTH(tempquery) > 159 THEN query.3 = SUBSTR(tempquery,159,79)
ELSE query.3 = ""
END ELSE DO
   query. = ""
END
IF st.opcount = "" THEN searchterms = "HELP"
ELSE searchterm = st.opcount
IF op.opcount = "" THEN operator = "*
ELSE operator = op.opcount
searchterm = searchterm + operator + ""
ELSE operator specification = operator
IF dose = "CONF" THEN "1053278 ISIS PANEL ;SCONFN2 ;ECONFN2",
* (NOCLEAR"
ELSE "1053278 ISIS PANEL ;SCREEN2 ;ENDSCN2 (NOCLEAR"
pnareturncode = RC
message = ""
IF pnareturncode >= 4 THEN CALL leaveisis
IF IODO = "EXIT" THEN CALL leaveisis
IF IODO = "START" THEN DO
   action = "C"
RETURN
END
searchterm = searchterm + operator specification
operator = operatorspecification
IF IODO = "HELP" | operator = "HELP" | searchterm = "HELP" THEN DO
IF IODO = "HELP" THEN DO
   operator = "HELP"
   IODO = ""
END IF searchterm = "HELP" THEN CALL accesshelp 2
IF operator = "HELP" & IODO = "" THEN CALL accesshelp 3
END ELSE LEAVE
END
error = 0
CALL verifyop
IF opcount = 0 THEN operator = "-"
operator = 0
CALL verifyop
IF opvalid = 1 & stvalid = 1 THEN DO
   IF eoq = 1 THEN DO
      temprightpar = rightparentheses
      DO WHILE temprightpar <= 0
         suffix = suffix ["]
      temprightpar = temprightpar - 1
   END
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

qquery = ""
RETURN
END ELSE opcount = opcount + 1
END END

/# subroutine to read a query #/
readquery:
/# read and verify user's query #/
IF query = "" THEN DO
   CALL insertleftpar
   qquery = tempquery || suffix
   suffix = ""
END
DO FOREVER
   tempquery = qquery
   query.1 = SUBSTR(tempquery,1,79)
   IF LENGTH(tempquery) > 88 THEN query.2 = SUBSTR(tempquery,88,79)
   ELSE query.2 = ""
   IF LENGTH(tempquery) > 159 THEN query.3 = SUBSTR(tempquery,159,79)
   ELSE query.3 = ""
   IF LENGTH(tempquery) > 88 THEN query.2 = SUBSTR(tempquery,88,79)
   ELSE query.2 = ""
   wholequery.1 = SUBSTR(tempquery,1,77)
   IF LENGTH(tempquery) > 76 THEN wholequery.2 = SUBSTR(tempquery,76,77)
   ELSE wholequery.2 = ""
   IF LENGTH(tempquery) > 155 THEN wholequery.3 = SUBSTR(tempquery,155,77)
   ELSE wholequery.3 = ""
   IF readnew = "A" THEN "1053278 ISIS PANEL ;SCRN3 ;ENDSCR3 (0001"
   ELSE "1053278 ISIS PANEL ;SCRN4 ;ENDSCR4 (0001"
pnareturncode = RC
message = ""
IF pnareturncode >= 4 THEN CALL leaveisis
IF IODO = "EXIT" THEN CALL leaveisis
qquery = wholequery.1 || wholequery.2 || wholequery.3
wholequery = wholequery.1 || wholequery.2 || wholequery.3
text = STRIP(text)
IF text = "" THEN saytext = "TEXT "="text"
ELSE saytext = ""
limits = STRIP(limits)
VM/CMS ISIS (Mainframe Version)
UPPER printformat sortformat
printformat = STRIP(printformat)

IF IODO = "**" THEN ITERATE

IF IODO = "SP01" THEN DO
  sort = "N"
  IF printformat = "*" THEN DO
    print = ""
  END ELSE DO
    print = "P"||printformat
  END
END

IF IODO = "SP04" THEN DO
  sort = "Y"
  IF printformat = "*" THEN DO
    print = ""
  END ELSE DO
    IF LENGTH(printformat) = 4 THEN
      print = "PPRINT FMT="||printformat||":"
    ELSE print = "PPRINT "||printformat||":"
  END
END

IF IODO = "HELP" THEN DO
  CALL accessshelp 0
  ITERATE
END

IF SUBSTR(iquery,1,1) = "-" THEN iquery = SUBSTR(iquery,2)
iquery = STRIP(iquery)
DO WHILE LENGTH(iquery) > 0 & SUBSTR(iquery,1,1) = "(*
  iquery = SUBSTR(iquery,2)
END

eq = 0
firsttime = 0
opcount = 1
leftparentheses = 0
rightparentheses = 0

IF iquery = " * | iquery = "**" THEN DO
  IF readnext = "A" THEN DO
    IF IODO = "STRY" THEN action = "C"
  RETURN
VM/CMS ISIS (Mainframe Version)

END
ELSE DO
  iquery = ""
  IF IODO = "STRY" THEN DO
    action = "C"
  RETURN
END
ITERATE
END

END

DO WHILE eq = 0 & opcount < maxcount
  searchers = ""
  operator = ""
  stvalid = 0
  opvalid = 0
  message = ""
  sterror = 0
  opererror = 0
  CALL parse
  CALL verifyop
  IF opcount = maxcount THEN operator = "*"
  IF sterror = 0 THEN operator = "*"
  CALL verifyop
  IF opererror = 0 & sterror = 0 THEN DO
    IF IODO = "STRY" THEN DO
      action = "C"
    RETURN
  END
  iquery = wholequery
  ITERATE
END

IF opvalid = 1 & stvalid = 1 THEN DO
  IF eq = 1 THEN DO
    IF IODO = "STRY" THEN action = "C"
  RETURN
END ELSE opcount = opcount + 1
END

IF IODO = "STRY" THEN DO
  action = "C"
  RETURN
END

VM/CMS ISIS (Mainframe Version)
/* subroutine to parse the query */
parse:
  # parse searchers /
  iquery = STRIP(iquery, 'L')

  IF SUBSTR(iquery, LENGTH(iquery), 1) = "*" THEN DO
    iquery = SUBSTR(iquery, 1, LENGTH(iquery) - 1)
  END
  IF SUBSTR(iquery, 1, 1) = "*" THEN iquery = SUBSTR(iquery, 2)

  IF iquery = "" THEN RETURN
  IF SUBSTR(iquery, 1, 1) = "*" THEN DO
    index = POS("", SUBSTR(iquery, 2))
    IF index = 0 THEN DO
      searchers = SUBSTR(iquery, 1, index + 1)
      iquery = SUBSTR(iquery, index + 2)
      index = 1
      CALL parsetag
    END
    IF sterror = 1 THEN RETURN
  END
  ELSE DO
    message = "** unmatched " in query ("iquery")"
  sterror = 1
  RETURN
END

ELSE DO index = 2 TO LENGTH(iquery)
  IF SUBSTR(iquery, index, 1) = "|" | SUBSTR(iquery, index, 1) = "*" |,
     SUBSTR(iquery, index, 1) = "*" | SUBSTR(iquery, index, 1) = "*" |,
     SUBSTR(iquery, index, 3) = "." & LENGTH(iquery) > index + 3 |,
     SUBSTR(iquery, index, 3) = "*" & LENGTH(iquery) > index + 3
    THEN DO
      searchers = SUBSTR(iquery, index - 2)
      iquery = SUBSTR(iquery, index)
      index = 1
      LEAVE index
  END
IF SUBSTR(iquery, index, 1) = "*" THEN DO
  index = index + 1
  searchers = SUBSTR(iquery, index - 1)
  iquery = SUBSTR(iquery, index)
  index = 1
  CALL parsetag
  LEAVE index
END
CALL parsetag

VM/CMS ISIS (Mainframe Version)
END
DO WHILE SUBSTR(query[index,1]) = "(" 
  index = index + 1 
END 
LEAVE
END
operator = SUBSTR(query[1,index-1])
query = SUBSTR(query[index])
IF query = "" THEN DO
  IF operator = "~-" && (LENGTH(operator) > endrightpar) THEN DO
    message = "*** The query was incorrectly terminated by an operator: ".
    SUBSTR(operator,endrightpar+1)
    operator = "~
  END
END
RETURN

/* parse tag */
parsetag:
IF SUBSTR(query[index,2]) = "/*" THEN
  DO jindex = index + 2 TO LENGTH(query)
    IF VERIFY(SUBSTR(query[index,jindex,1],",")) = 0 THEN ITERATE
  IF VERIFY(SUBSTR(query[index,jindex,1],",")) = 0 THEN DO
    searchterm = searchterm || SUBSTR(query[1,index])
    query = SUBSTR(query[1,jindex])
    index = index + 1
  END
  RETURN
END
ELSE DO
  message = "*** Incorrect tag ***query****
  closingpar = POS("*/",query)
  IF closingpar = 0 THEN error = SUBSTR(query,closingpar+1)
  ELSE error = 1
  RETURN
END
RETURN

/* subroutine to verify searchterm */
verify:
CALL quit? searchterm searchterm = RESULT
VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS (Mainframe Version)

IF searchterm = "" THEN DO
  IF oppcount = 1 THEN DO
    operator = "-" 
    stvalid = 1
    st.oppcount = ""
  RETURN
END
ELSE DO
  message = "*** There is an empty search term."
  sterror = 1
  st.query = ""
  RETURN
END
CALL gencheck searchterm
IF sterror = 0 THEN DO
  IF SUBSTR(searchterm[3,1]) = "~" THEN CALL prefixed searchterm
  ELSE CALL unprefixed searchterm
END
ELSE IF message = "" THEN DO
  message = "*** The search term ***searchterm*** must be in quotation marks."
  st.query = ""
  END
IF sterror = 0 THEN DO
  stvalid = 1
  IF quotes = 1 THEN searchterm = "" || searchterm || ""
  st.oppcount = searchterm || tagqualification
END 
RETURN

/* subroutine to run general checks on a search term */
gencheck:
PARSE ARG searchterm
IF VERIFY(searchterm,"","") = 0 THEN DO
  message = "*** The search term ***searchterm*** contains an apostrophe."
  st.error = 1
  RETURN
END
quotes = 0
tagqualification = ""
tagposition = POS("/*",searchterm)
VM/CMS ISIS (Mainframe Version)
IF tagposition > 0 & SUBSTR(searchterm,LENGTH(searchterm)) = ")" THEN DO
  tagqualification = SUBSTR(searchterm,tagposition)
  searchterm = SUBSTR(searchterm,1,tagposition-1)
END IF
DO jndx = 3 TO LENGTH(tagqualification)
  IF verify(tagqualification,jndx,1,".*,0123456789") = 0 THEN,
    ITERATE
  END IF
END jndx
ELSE DO
  message = "** Incorrect tag ** tagqualification**"
  query = ""
  sterror = 1
END IF
RETURN
END IF
ELSE DO
  searchterm = SUBSTR(searchterm,2,len-2)
  quotes = 1
END IF
ELSE DO
  message = "** The search term ("searchterm,
" ) must contain ** as first and last.
"query = ""
  sterror = 1
END IF
RETURN
END IF
ELSE DO
  message = "** The search term ** searchterm,
"query = ""
  sterror = 1
END IF
RETURN
END IF

### INCLUDES
"INC CORRECT >> 119**
"INC CORRECT >> 118**

/* This subroutine to process an unprefixed search term */
unprefixed:
PARSE ARG searchterm
  IF POS(" ",searchterm) = 0 THEN DO
    message = "** An unprefixed search term cannot contain blanks: ",
      searchterm
    query = ""
  sterror = 1
  END IF
RETURN

/* This subroutine to process a prefixed search term */
prefixed:
PARSE ARG searchterm
  IF SUBSTR(searchterm,1,2) = "AU" THEN DO
    IF verify(searchterm," ","**") = 0 THEN DO
      message = "** The hyphens in the author's name: ",
        searchterm
      query = ""
      sterror = 1
    RETURN
  END IF
  IF verify(searchterm," ","*") = 0 THEN DO
    message = "** An author's name must have initials. ",
      terminate it by ", ** searchterm"
    query = ""
    sterror = 1
  END IF
END IF
RETURN

VM/CMS ISIS (Mainframe Version)
END
RETURN

/* subroutine to verify an operator */
verifyop:
CALL quit? operator
operator = RESULT
	 fleetpar = leftparentheses
temprightpar = rightparentheses
index = 1
DO WHILE SUBSTR(operator,index,1) = "("
index = index + 1
IF temprightpar > 0 THEN temprightpar = temprightpar - 1
ELSE templeftpar = templeftpar + 1
END
IF SUBSTR(operator,index,1) = ""
THEN eqo = 1
ELSE IF INDEX("=",SUBSTR(operator,index,1)) = 0 THEN index = index + 1
ELSE IF "(" = SUBSTR(operator,index,3) THEN index = index + 3
ELSE IF "=" = SUBSTR(operator,index,1) THEN DO
     operator = SUBSTR(operator,index,1)
     eqo = 1
END
ELSE IF POS("*",SUBSTR(operator,index,2)) > 0 THEN
CALL textsearch "*
ELSE IF POS(".",SUBSTR(operator,index,2)) > 0 THEN
CALL textsearch "."
ELSE DO
     message = "Clax Incorrect operator "** operator"**
     iquery = ""
     SIGNAL errorexit
END
DO WHILE eqo = 0 & SUBSTR(operator,index,1) = "("
     index = index + 1
temprightpar = temprightpar + 1
END
IF eqo = 0 & SUBSTR(operator,index) = "" THEN SIGNAL errorexit
IF st.VALID = 1 THEN DO
     IF st.opcount = "" THEN query = query||"."||st.opcount
VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS
REXX EXEC

IF operator = "=" THEN DO
     op.opcount = operator
tempparentheses = temprightpar
END
ELSE op.opcount = ""
opvalid = 1
     query = query||op.opcount
leftparentheses = templeftpar
END
RETURN
errorexit:
     message = "Incorrect operator must be one of the following : ",
     "+,-, =, <, > . It is: ** operator**
     iquery = ""
RETURN

/* subroutine to verify a textsearch operator */
textsearch:
PARSE ARG text
DO WHILE LENGTH(operator) > index
     DO WHILE SUBSTR(operator,index,1) = ""
            operator = SUBSTR(operator,index,1)||SUBSTR(operator,index+1)
     END
operator = SUBSTR(operator,index,1)||SUBSTR(operator,index)
IF SUBSTR(operator,index+1,1) = text THEN DO
     index = index + 1
END
INDEX = index + 2
operator = operator||""
RETURN

/* insert left parentheses into query */
insertleftpar:
     tempquery = query
templeftpar = leftparentheses
DO WHILE templeftpar > 0
     tempquery = "("||tempquery
templeftpar = templeftpar - 1
END
RETURN
VM/CMS ISIS (Mainframe Version)
/* subroutine to create the cards file and to execute the query */
execute:

CALL insertleftpar
query = tempquery
temprightpar = rightparentheses
DO WHILE temprightpar > 0
  query = query||"
  temprightpar = temprightpar - 1
END

query = query
leftparentheses = 0
rightparentheses = 0

query_1 = SUBSTR(query,1,79)
IF LENGTH(query_1) > 88 THEN query_2 = SUBSTR(query,89,79)
ELSE query_2 = ""

IF LENGTH(query) > 159 THEN query_3 = SUBSTR(query,160,79)
ELSE query_3 = ""

IF LENGTH(query) > 238 THEN DO
  message = "The query is too long, please shorten it!"
  action = R
  RETURN
END

fn = "YSPRD"
ft = "CARDS"
"ESTATE" fn ft "A"
rcasted = RC
IF rcasted = 0 THEN "ERASE" fn ft "A"

line_1 = "C"||limits
linenumber = 2
IF sort = "Y" THEN DO
  line.linenuber = "S"||sortformat
  linenumber = linenumber + 1
END

IF print = "Y" THEN DO
  line.linenuber = print
  linenumber = linenumber + 1
END

line.linenuber = "="||query_1
linenumber = linenumber + 1
IF LENGTH(query) > 88 THEN DO
  line.linenuber = "="||query_2
  linenumber = linenumber + 1
END

VM/CMS ISIS (Mainframe Version)

linenumber = linenumber + 1
END

IF LENGTH(query) > 159 THEN DO
  line.linenuber = "="||query_3
  linenumber = linenumber + 1
END

IF saytext = "" THEN DO
  line.linenuber = saytext
  linenumber = linenumber + 1
END

line.linenuber = "$"

DO index = 1 to linenumber
  "EXEC 1 DISKW fn ft "A" index F (STRING) line.index"
END

"IDENTIFY (FIFO)"

PULL note_1
note_1 = dbase||"||sort||"||mode||"||readnews||"||indicator||" +
note_2 = note_2 + SUBSTR(printformat,1,4)||" +
note_2 = note_2 + SUBSTR(sortformat,1,4)||"||saytext

DO index = 1 to 3
  jindex = index + 2
  note.jindex = query.jindex
END

"ESTATE ISI ISIS AB"
IF rc = 0 THEN "ERASE ISI ISIS AB"
DO index = 1 to 5
  "EXEC 1 DISKW ISI ISIS AB index F (STRING) note.index"
END

"EXEC SENDFILE ISI ISIS AB TO ISI (NOTYPE)"

"ESTATE ISI ISIS AB"
IF rc = 0 THEN "ERASE ISI ISIS AB"

IF sort = "N" THEN DO
  IF readnews = "A" THEN "1053270 ISIS PANEL ; SCREENS ; ENDS CRS (NOWAIT)"
  ELSE "1053270 ISIS PANEL ; SCREENS ; ENDS CRS (NOWAIT)"
  EXEC CVQUERY OBGN="dbase "CLNF="rcast
  returncode = RC
  IF dbase = "PREP" | dbase = "BOOK" THEN DO
    CALL order.sort dbase query
    IF IOCB = "STRT" THEN DO
      action = "C"
      RETURN
    END
  VM/CMS ISIS (Mainframe Version)
IF IOSD = "EXIT" THEN CALL leaveisis
END ELSE DO
  "XEDIT VSP01 LISTING A (PROF ISIS"
  xeditreturncode = RC
  IF xeditreturncode = 88 THEN message = ,
  "** too much output for display, consult VSP01 LISTING A"
END END ELSE DO
  IF readameus = "A" THEN "105270 ISIS PANEL ;SCEEN ;ESCRE (NOWAIT"
  ELSE "105270 ISIS PANEL ;SCEEN ;ESCRE (NOWAIT"
  "SET CMSTYPE HT"
  "EXEC CYQUERY  DBN="dbname" CLN="cmase" LW="lu" C4="cu" LP="lp"
  NP="nh" TL="tol" NC="nc" IND ="ind" NT=nt"
returncode = RC
  "SET CMSTYPE RT"
  IF returncode = 0 THEN DO
    IF dbname = "PREP" & dbname = "BOOK" THEN DO
      CALL order "N" dbname query
      IF IOSD = "STR" THEN DO
        action = "C"
      RETURN
      END IF IOSD = "EXIT" THEN CALL leaveisis
      END ELSE DO
        "XEDIT VSP01 LISTING A (PROF ISIS"
        xeditreturncode = RC
        IF xeditreturncode = 88 THEN message = ,
        "** too much output for display, consult VSP01 LISTING A"
      END END ELSE DO
    IF dbname = "PREP" & dbname = "BOOK" & nc = 1 THEN DO
    CALL order "S" dbname query
  IF IOSD = "STR" THEN DO
      action = "C"
      RETURN
  END IF IOSD = "EXIT" THEN CALL leaveisis
  END ELSE DO
    "XEDIT VSP04 LISTING A (PROF ISIS"
    xeditreturncode = RC
    IF xeditreturncode = 88 THEN message = ,
    "** too much output for display, consult VSP04 LISTING A"
  END
VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS

REXX EXECs

END
END
action = "C"
RETURN

/* subroutine to determine if user has opted to quit ISIS */
quitz:
  PARSE UPPER ARG reply
  IF LENGTH(reply) = 0 THEN RETURN reply
  reply = STR(reply)
  RETURN reply

/* subroutine to call the HELP facility */
accesshelp: PARSE ARG help
  IF dbname = "BOOK" & dbname = "PREP" & dbname = "CONF" THEN DO
    IF help = 1 THEN DO
      "105270 ISIS PANEL ;S"dbname"BE ;E"dbname"BE ;INOCLEAR 0001"
    END IF help = 2 THEN DO
      "105270 ISIS PANEL ;S"dbname"ST ;E"dbname"ST ;INOCLEAR 0001"
    END IF help = 3 THEN DO
      "105270 ISIS PANEL ;HELPPO ;ENDOPE ;INOCLEAR 0001"
    END panelreturncode = RC
    message = ""
    IF panelreturncode = 4 THEN CALL leaveisis
    IF IOSD = "HELP" THEN RETURN
    ELSE DO
      IF dbname = "BOOK" & dbname = "PREP" & dbname = "CONF" THEN DO
      page = 0
      DO FOREVER
        IF page = 0 THEN DO
          IF IOSD = "PREP" THEN npg = 11
          page = "01"
        END "105270 ISIS PANEL ;S"dbname||page" ;E"dbname||page,
        ;INOCLEAR 0001"
      panelreturncode = RC
      message = ""
      IF panelreturncode = 4 THEN CALL leaveisis
    VM/CMS ISIS (Mainframe Version)
Mainframe VM/CMS ISIS

REXX EXEC

IF I0SD = "EXIT" THEN CALL leaveisis
IF I0SD = "CONT" THEN RETURN
IF I0SD = "" THEN I0SD = "NEXT"
IF DATATYPE(npg) = "NUM" THEN npg = 0
IF DATATYPE(page) = "NUM" THEN page = 0
IF I0SD = "PREV" THEN DO
IF npg > 0 & npg <= maxpage THEN DO
  page = npg
  IF page <= maxpage THEN npg = page + 1
  ELSE npg = 0
END ELSE page = 0
END
ELSE DO
IF npg = page + 1 | page = maxpage THEN DO
  npg = page
  IF page > 11 THEN page = 1
ELSE DO
  page = 0
  npg = 0
END END ELSE DO
IF npg > 0 & npg <= maxpage THEN DO
  page = npg
  IF page <= maxpage THEN npg = page + 1
  ELSE npg = 0
END ELSE DO
  page = 0
  npg = 0
END END
END
RETURN

/* welcome user to ISIS */
welcome:
CALL quit? mode
mode = SUBSTR(RESULT,1,1)
IF mode = "A" THEN mode = "B"
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

CALL quit? readnews
readnews = SUBSTR(RESULT,1,1)
IF readnews = "Y" THEN DO
  "XEDIT ISIS NEWSG = (PROF ISIS"
  readnews = "N"
  I0SD = "HELP"
END IF readnews = "A" THEN readnews = "N"
RETURN

/* subroutine to quit ISIS exec, with comments if desired */
leaveisis:
IF dbase = "DBASE" THEN dbase = "PREP"
IF mode = "MODE" THEN mode = "B"
IF readnews = "READNEWS" THEN readnews = "N"
IF iquery = "IQUERY" & wherequery = "WHEREQUERY" THEN iquery = ""
ELSE IF iquery = "" & wherequery = "WHEREQUERY" THEN iquery = ""
ELSE IF iquery = "" & wherequery = "" THEN iquery = wherequery
variables = "\"dbase\""|mode|""readnews|""iquery|""iu|""cu|""ip|""nh,
|""to|""t|\""nth|\""ind|\""init""\"GLOBALLY SELECT WEXEC SETLP execname variables
EXIT 0

order: PROCEDURE EXPOSE message load
exec to reserve printed material under vs/cms
ADDRESS COMMAND
ARG parameters
PARSE SOURCE . . execname .
PARSE VAR parameters sort database query
message = ""
result = ""
external = "N"
IF sort = "E" THEN external = "Y"
DO idummy = 1 WHILE external = "N"
  number = 8
  total = 0
VM/CMS ISIS (Mainframe Version)
start = 0
outputsize = 0
output = ""
first. = 0
last. = 0

IF sort = "N" THEN DO
  "ESTATE VSP01 LISTING A"
  rcsaved = RC
  IF rcsaved = 0 THEN DO
    SAY "****** error in execname:******"
    SAY "VSP01 LISTING A not found."
    RETURN 16
  END
  *EXECIO DISKR VSP01 LISTING A (FIFO*)
  inputsize = queued()
  DO index = 1 TO inputsize
    PARSERC(temporary,1,1) = "1" THEN ITERATE
    ELSE temporary = SUBSTR(temporary,2)
    temp = STRIP(temporary,7)
    IF temp = "" THEN ITERATE
    IF SUBSTR(temp,14,33) = "UNESCO COMPUTERIZED DOCUMENTATION"
    THEN DO
      IF number = 0 THEN DO
        number = number + 1
        ITERATE index
      END
      ELSE DO
        start = -1
        ITERATE index
      END
    END
    IF number = 0 THEN DO
      ITERATE index
    END
    IF SUBSTR(temp,1,29) = "NUMBER OF RECORDS RETRIEVED = "
    THEN DO
      total = SUBSTR(temp,33,5)
      IF total > 0 THEN number = number - 1
      ELSE number = 0
      ITERATE index
    END
  END
VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS
REXX EXEC

END
IF start >= 0 THEN DO
  IF start = 0 THEN DO
    start = 1
    outputsize = outputsize + 1
    output.outputsize = ""
    first.number = outputsize + 1
    outputsize = outputsize + 1
    output.outputsize = ""||FORMAT(number,3)||"."
    start = start + 1
    outputsize = outputsize + 1
    output.outputsize = ""||temp
  END
  ELSE DO
    buffer = temp
    IF POS(""||buffer) > 0 THEN DO
      ind = 1
      DO FOREVER
        PARSE VAR buffer.head.ind "(" buffer
        ind = ind + 1
        PARSE VAR buffer.head.ind ")" buffer
        IF LENGTH(head.ind) = 36 THEN iterate
      IF DATATYPE(head.ind,"N") = 1 THEN ITERATE
      start = 0
      last.number = outputsize + 1
      number = number + 1
      LEAVE
    END
    outputsize = outputsize + 1
    output.outputsize = ""||temp
  END
END
END
IF number = 0 THEN DO
  total = FORMAT(number,5)
  message = "**no references found for query: **query"
END
ELSE DO
  "ESTATE VSP04 LISTING A"
  rcsaved = RC
VM/CMS ISIS (Mainframe Version)
IF $rcsaved = 0 THEN DO
  SAY "***** error in" execname "*****"
  SAY "*VSP04 LISTING A* not found."
  RETURN -16
END

*EXECIO = DISKR VSP04 LISTING A (FIFO)*
inputsize = queued(1)
DO index = 1 TO inputsize
  PARSE PULL $li temporary
  PARSE VAR temporary.head " figure "
  IF head = "\%" & DATATYPE(figures) = "NUM" THEN DO
    ITERATE index
  END
  temp = STRIP(temporary.T)
  IF temp = "" THEN ITERATE
  IF start = 0 THEN DO
    start = 1
    number = number + 1
    outputsize = outputsize + 1
    output.outputsize = 
    first.number = outputsize + 1
    output.outputsize = 
    output.outputsize = "\%" || FORMAT(number,3)|| "."
    start = start + 1
    output.outputsize = "\%" || temp
  END
ELSE DO
  buffer = temp
  IF POS("",buffer) > 0 THEN DO
    head = ""
    ind = 1
  DO FOREVER
    PARSE VAR buffer.head.ind "\%" buffer
    IF buffer = "\%" THEN LEAVE
    ind = ind + 1
    PARSE VAR buffer.head.ind "\%" buffer
    IF LENGTH(buffer.head.ind) >= 6 THEN ITERATE
    IF DATATYPE(head.ind,"\%") = 1 THEN ITERATE
    start = 0
    last.number = outputsize + 1
    LEAVE
  END
END
VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS

REXX EXEC

outputsize = outputsize + 1
output.outputsize = "\%" || temp
END
END
total = FORMAT(number,5)
IF number < 1 & start = 0 THEN DO
  number = 0
END
IF number = 0 THEN DO
  message = "*** no references found for query: query"
END
END
IF message = "" & query = "" THEN DO
  message = "Your query was: query"
END
DO jdummy = 1 WHILE number = 0
  IF message = "" & query = "" THEN DO
    message = "Your query was: query"
  END
  "103278 ISIS PANEL ; SCREEN ; ENDSRC"
  panaireturncode = RC
  message = ""
  IF panaireturncode = 4 THEN SIGNAL leaveorder
  IF IO9D = "EXIT" | IO9D = "STRT" THEN SIGNAL leaveorder
  IF IO9D = "EXIT" THEN DO
    external = "Y"
  LEAVE jdummy
END
IF sort = "H" THEN DO
  "XEDIT VSP01 LISTING A (PROF ISIS"
  xeditreturncode = RC
  IF xeditreturncode = 88 THEN message = ", "asm too much output for display, consult VSP01 LISTING A"
END
ELSE DO
  "XEDIT VSP04 LISTING A (PROF ISIS"
  xeditreturncode = RC
  IF xeditreturncode = 88 THEN message = ", "asm too much output for display, consult VSP04 LISTING A"
END
ITERATE jdummy
VM/CMS ISIS (Mainframe Version)
"IOS3270 ISIS PANEL :SCREENB :ENDSCRBN"
panel.returncode = RC
message = ""
 IF panel.returncode = 4 THEN SIGNAL leaveorder
 IF IOSD = "EXIT" | IOSD = "STR" THEN SIGNAL leaveorder

 IF IOSD = "EXTE" THEN DO
   external = "Y"
   LEAVE idummy
 END
 IOSD = "REDI"

 "ESTATE ORDER LISTING A"
 rc received = RC
 IF received = 8 THEN "ERASE ORDER LISTING A"

 DO ondex = 1 TO outputsize
   "EXECI 1 DISKW ORDER LISTING A* ondex" F 17S (STRING),
   output.ondex
 END

 DO idummy = 1 WHILE IOSD = "REDI"

 "XEDIT ORDER LISTING A (PROF ISIS"
 xedit.returncode = RC
 IF xedit.returncode = 88 THEN DO
   message = "too much output, narrow your query and try again."
   SIGNAL leaveorder
 END

 DO adummy = 1 WHILE IOSD = "REDI"

 "ESTATE LIBDESK ORDER AB"
 IF rc = 0 THEN "ERASE LIBDESK ORDER AB"
 num. = COPIES(T,4)
 IF message = "" & query = "" THEN DO
   message = "Your query was: " query
 END

 "IOS3270 ISIS PANEL :SCREENA :ENDSCRAN"
 panel.returncode = RC
 message = ""
 IF panel.returncode = 4 THEN SIGNAL leaveorder

 VM/CMS ISIS (Mainframe Version)

 IF IOSD = "EXIT" | IOSD = "STR" THEN SIGNAL leaveorder

 IF IOSD = "EXTE" THEN DO
   external = "Y"
   LEAVE idummy
 END

 IF IOSD = "REDI" THEN DO
   "NOTE.1 = "DEMAND FOR LOAN, DBN=" database
   "IDENTIFY IFIFO"
   PULL note.2
   NOTE.2 = ""
   "ESTATE LIBDESK ORDER AB"
   IF rc = 0 THEN "ERASE LIBDESK ORDER AB"
   outputsize = 3

   DO index = 1 TO 3
     num.index = TRANSLATE(num.index," "," ")
     num.index = STRIP(num.index,B)
     IF num.index = "" THEN ITERATE
     IF DATATYPE(num.index) = "NUM" & num.index = "" THEN DO
       message = "please select the number ",
       "of a publication, not" num.index
       IOSD = "REDI"
       ITERATE adummy
     END
   END
   IF num.index < 1 | num.index > total THEN DO
     message = "invalid publication ",
     "number ("num.index")"
     IOSD = "REDI"
     ITERATE adummy
   END
   num.index = FORMAT(num.index,4)
   num = STRIP(num.index,B)
   DO index = first.num TO last.num
   IF index = first.num THEN DO
     outputsize = outputsize + 1
     note.outputsize = ""
   END
 ELSE DO
   outputsize = outputsize + 1
   note.outputsize = outputsize
 END

 IF outputsize = 3 THEN DO

 VM/CMS ISIS (Mainframe Version)
message = "*** ISIS did not find the publication."
IOSD = "RED1"
ITERATE d dummy
END

IF message = "" & query = "" THEN DO
message = "Your query was: " query
END

"1053270 ISIS PANEL ;SCRENC ;ENDSRC"
panellreturncode = RC
message = ""
IF panellreturncode = 4 THEN SIGNAL leaveorder
IF IOSD = "EXIT" | IOSD = "STR" THEN SIGNAL leaveorder
IF IOSD = "EXT" THEN DO
external = "Y"
LEAVE d dummy
END

DO index = 1 to outpulsize
   "EXEC 1 DISKU LIBDESK ORDER A8 "index" F 135 (STRING) ,
   note.index
END

"EXEC SENDFILE LIBDESK ORDER A8 TO LIBDESK (NOTYPE)"

"ESTATE LIBDESK ORDER A8"
IF rc = 0 THEN "ERASE LIBDESK ORDER A8"
result = "... Your request for an internal loan has been",
"forwarded."

note.2 = note.2||" Internal Loan"

"ESTATE ISI ORDER A8"
IF rc = 0 THEN "ERASE ISI ORDER A8"
"EXEC 1 DISKU ISI ORDER A8 1 F 135 (STRING) note.2"

"EXEC SENDFILE ISI ORDER A8 TO ISI (NOTYPE)"
"ESTATE ISI ORDER A8"
IF rc = 0 THEN "ERASE ISI ORDER A8"

SIGNAL leaveorder
END

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

VM/CMS ISIS

END

DO d dummy = 1

IF DATABASE = "BOOK" THEN title.1 = COPIES("-",46)
ELSE title.1 = COPIES("-",66)
title.2 = COPIES("-",74)
IF DATABASE = "BOOK" THEN author = COPIES("-",55)
ELSE author = COPIES("-",62)

place = COPIES("-",81)
publisher = COPIES("-",84)
date = COPIES("-",12)
reportnumber = COPIES("-",58)
pertit = COPIES("-",42)
volume = COPIES("-",18)

vnumber = COPIES("-",18)
pages = COPIES("-",18)
isbn = COPIES("-",24)
comment.1 = COPIES("-",33)
comment.2 = COPIES("-",74)

DO d dummy = 1

IF message = "" & query = "" THEN DO
message = "Your query was: " query
END

IF database = "BOOK" THEN "1053270 ISIS PANEL ;SCREBOA ;ENDSOBA"
ELSE "1053270 ISIS PANEL ;SCREPO ;ENDSPRA"
panellreturncode = RC
message = ""
IF panellreturncode = 4 THEN SIGNAL leaveorder
IF IOSD = "EXIT" | IOSD = "STR" THEN SIGNAL leaveorder

IF database = "BOOK" THEN DO

   title.1 = TRANSLATE(title.1,"","")
title.2 = TRANSLATE(title.2,"","")
author = TRANSLATE(author,"","")
publisher = TRANSLATE(publisher,"","")
place = TRANSLATE(place,"","")
publisher = TRIP(publisher,B)
date = TRANSLATE(date,"","")

VM/CMS ISIS (Mainframe Version)
isbn = TRANSLATE(isbn, "", ":")
isbn = STRIP(isbn, B)
comment.1 = TRANSLATE(comment.1, "", ":")
comment.1 = STRIP(comment.1, B)
comment.2 = TRANSLATE(comment.2, "", ":")
comment.2 = STRIP(comment.2, B)
END
IF database = "PREP" THEN DO
  title.1 = TRANSLATE(title.1, "", ":")
title.1 = STRIP(title.1, B)
title.2 = TRANSLATE(title.2, "", ":")
title.2 = STRIP(title.2, B)
  author = TRANSLATE(author, "", ":")
  author = STRIP(author, B)
  place = TRANSLATE(place, ":", ":")
  place = STRIP(place, B)
publisher = TRANSLATE(publisher, ":", ":")
publisher = STRIP(publisher, B)
  pertit = TRANSLATE(pertit, ":", ":")
  pertit = STRIP(pertit, B)
  volume = TRANSLATE(volume, ":", ":")
  volume = STRIP(volume, B)
  volume = RIGHT(volume, 18)
  vnumber = TRANSLATE(vnumber, ":", ":")
  vnumber = STRIP(vnumber, B)
  vnumber = RIGHT(vnumber, 18)
  pages = TRANSLATE(pages, ":", ":")
  pages = STRIP(pages, B)
  date = TRANSLATE(date, ":", ":")
  date = STRIP(date, B)
  reportnumber = TRANSLATE(reportnumber, ":", ":")
  reportnumber = STRIP(reportnumber, B)
  comment.1 = TRANSLATE(comment.1, ":", ":")
  comment.1 = STRIP(comment.1, B)
  comment.2 = TRANSLATE(comment.2, ":", ":")
  comment.2 = STRIP(comment.2, B)
END
IF title.1 = "" & title.2 = "" & author = "" & place = ""
  & publisher = "" & date = "" & comment.1 = ""
  & comment.2 = ""
  THEN DO
  IF database = "BOOK" THEN DO
    IF isbn = "" THEN ITERATE idummy
  END
  IF database = "PREP" THEN DO
    IF pertit = "" & volume = "" & vnumber = "" & pages = ""
      THEN DO
      END
    END
    END
  END
  THEN DO
  END
  END
VM/CMS ISIS (Mainframe Version)
outputsize = outputsize + 1
note.outputsize = "Publisher: ":\|\|\|publisher
END
IF date == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "Year of Publication: ":\|\|date
END
IF isbn == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "ISBN: ":\|\|isbn
END
IF comment.1 == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "Comment: ":\|\|comment.1
END
IF comment.2 == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = comment.2
END
END
IF database == "PREP": THEN DO
    IF author == "": THEN DO
        outputsize = outputsize + 1
        note.outputsize = "Author(s): ":\|\|author
    END
    IF title.1 == "": THEN DO
        outputsize = outputsize + 1
        note.outputsize = "Title: ":\|\|title.1
    END
    IF title.2 == "": THEN DO
        outputsize = outputsize + 1
        note.outputsize = title.2
    END
IF reportnumber == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "Report Number: ":\|\|reportnumber
END
IF pertit == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "Title of Periodical: ":\|\|pertit
END
IF volume == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "Volume: ":\|\|volume
END
IF vnumber == "": THEN DO
    outputsize = outputsize + 1
END
VM/CMS ISIS (Mainframe Version)

note.outputsize = "Volume Number: ":\|\|vnumber
END
IF pages == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "Pages: ":\|\|pages
END
IF place == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "Place of Publication: ":\|\|place
END
IF publisher == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "Publisher: ":\|\|publisher
END
IF date == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "Year of Publication: ":\|\|date
END
IF comment.1 == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "Comment: ":\|\|comment.1
END
IF comment.2 == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = comment.2
END
END
IF query == "": THEN DO
    outputsize = outputsize + 1
    note.outputsize = "ISIS Query: ":\|\|query
END
DO index = 1 to outputsize
    EXCEID 1 DISKW CHACHA ORDER AB "index" F (STRING) note.index
END
"EXEC SENDFILE CHACHA ORDER AB TO CHACHA (NDTYPE"
"ESTATE CHACHA ORDER AB"
IF rc = 0 THEN "ERASE CHACHA ORDER AB"
note.2 = note.2\|\| Inter-Library Loan"
"ESTATE I5 ORDER AB"
IF rc = 0 THEN "ERASE I5 ORDER AB"
"EXECID 1 DISKW I5 ORDER AB 1 F I55 (STRING) note.2
VM/CMS ISIS (Mainframe Version)
"EXEC SENDFILE IS1 ORDER A9 TO IS1 (NOTYPE)"
"ESTATE IS1 ORDER A9"
IF rc = 0 THEN "ERASE IS1 ORDER A9"
   result = "... Your request for an inter-library loan has been",
   "forwarded."
SIGNAL leaveorder
END
END
leaveorder:
message = result
RETURN

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

:SCREEN1

LOAD Next Field, RETURN Continue, PF81 Help, PF83 Quit, PF85 Inter-library Loan
:CFY HELP % EXIT % % % % EXTE
:---- IS1, CERN PUBLIC EXEC TO INTERROGATE THE LIBRARY DATA BASES """
:---- AND TO REQUEST PUBLICATIONS FOR LOAN
:----
J & &  &
CINE
MESSAGE

Please indicate which data base you would like to search.
PREP Preprints and Reports, or
BOOK Books, Pamphlets and Standards, or
CONF Conferences

---------------------------------------------------------------
#database
Please indicate the mode of the search.
B Beginners (will be prompted for every search term), or
A Advanced user (will be prompted once for the whole query)

---------------------------------------------------------------
#mode
You may read the IS1 news number & dated &/&/&2 (Y/N)

---------------------------------------------------------------
#readnews
If you already know that the publication in which you are
interested is not available in the CERN Libraries you may
submit an inter-library loan request. Simply press PF85.

Comments, criticisms or suggestions concerning IS1 are welcome.
They should be sent to IS1 AT CERNVM.

:ENDSCR

:SCREEN2

LOAD Next Field, RETURN Continue, PF81 Help, PF83 Restart, PF85 Quit
:CFY HELP STR! EXIT

J & &  &
CINE
MESSAGE

The data base to be searched is : &database
So far you have formulated the query:
&query.1
&query.2
&query.3

Enter a full search term, or the first few letters of a search term

VM/CMS ISIS (Mainframe Version)
truncated by $. Example 1: HAD$; Example 2: AU=MILLER$

------------- #&searchteravariablenamespecification

Enter an operator (+, -, $, .) or terminate (-). An operator may be preceded by any number of "","", and followed by any number of "","".

------------- #&operatorspecification

ENDSCR
;SCREEN
:B
;TAB Next Field, RETURN Return Screen, PF01 Help, PF02 Restart, PF03 Quit
;CYF HELP STRT EXIT

;J &amp & CNE!

&MESSAGE
The data base to be searched is: &dbase
So far you have formulated the query:
&query.1
&query.2
&query.3
&querytext

Enter a full search term, or the first few letters of a search term truncated by $. Example 1: CC=LEARCSBE$; Example 2: PL=LES ARC$

------------- #&searchteravariablenamespecification

Enter an operator (+, -, $, .) or terminate (-). An operator may be preceded by any number of "","", and followed by any number of "","".

------------- #&operatorspecification

;ENDSCR
;SCREEN
:B
PF05 Execute Search (unsorted), PF06 Execute Search (sorted)
;CYF HELP STRT EXIT & SP01 SP04

;J &amp & CNE!

&MESSAGE
The data base to be searched is: &dbase
So far you have formulated the query:

Mainframe VM/CMS ISIS

------------- #&printformat
SORT(format) -------- #&sortformat
TEXT ------------- #&text
LIMITS -------- #&limits

LW=#36w CL=#36c w LP=#36x p NH=#1&nh TOL=#2&tol NC=#1&nc IND=#2&ind NT=#1&nt

VM/CMS ISIS (Mainframe Version)
Mainframe VN/CMS ISIS

REXX EXEC

DISPLAY 221

TAB Next Field, RETURN Refresh Screen, PF01 Help, PF02 Restart, PF03 Quit

;ENDSCR

;SCREENS

B

Please wait - do not press any key

.CY

====== YOUR QUERY HAS BEEN SUBMITTED TO THE VM/CMS SYSTEM ======

====== PLEASE WAIT PATIENTLY UNTIL IT HAS BEEN EXECUTED ======

.J & & & &

.CNE!

The database to be searched is : &base

You have formulated the query:

&query.1

&query.2

&query.3

&saytext

The output will not be sorted.

;ENDSCR

Printformat --------> &printformat

Limits --------> &limits

;ENDCAS

;SCREENS

B

Please wait - do not press any key

.CY

====== YOUR QUERY HAS BEEN SUBMITTED TO THE VM/CMS SYSTEM ======

====== PLEASE WAIT PATIENTLY UNTIL IT HAS BEEN EXECUTED ======

.J & & & &

.CNE!

The database to be searched is : &base

You have formulated the query:

&query.1

&query.2

&query.3

VM/CMS ISIS (Mainframe Version)

Mainframe VN/CMS ISIS

REXX EXEC

DISPLAY 222

&saytext

The output will be sorted.

;ENDSCR

Printformat --------> &printformat

Sortformat --------> &sortformat

Limits --------> &limits

LU=fl UP C=nc LP=fp NH=nh TL=to=to NC=nc IND=ind NT=nt

;ENDCAS

;SCREENS

B

RETURN Continue, PF02 Restart, PF03 Quit, PF10 Inter-library Loan

.CYF % STRT EXIT % % % % EXTE

====== REQUEST FOR PUBLICATIONS FROM THE CERN LIBRARIES ======

.J & & & &

.CNE!

&message

ISIS found &total publication(s) in the database &database satisfying your criteria.

If you press RETURN, the list of publications will be displayed numbered from 1 to &total. Please note the numbers of the publications in which you are interested. You may later request up to three publications for loan.

If none of the publications found in the CERN database is of interest to you, then you may submit an inter-library loan request. Simply press PF89.

Important: If you are not registered in the CERN phone book, please apply personally to the Library Information Desk.

;ENDSCR

;SCREENS

B

PF02 Restart, PF03 Quit, PF10 Inter-library Loan, PF10 Display anyway

.CYF % STRT EXIT % % % % EXTE DISP

VM/CMS ISIS (Mainframe Version)
NO PUBLICATIONS FOUND IN THE CERN LIBRARIES

If you wish, you may submit an inter-library loan request.

Important: If you are not registered in the CERN phone book, please apply personally to the library information desk.

Please give the list number(s) of the publication(s) you wish to request:

First Number: #4&num.1
Second Number: #4&num.2
Third Number: #4&num.3

If none of the listed publications satisfied you, you may request the desired publication as an inter-library loan. In this case, press PF80.

You selected the following entries from the list of references retrieved by ISIS and would like the Library to send you the related documents on loan.

Number: #num.1
Number: #num.2
Number: #num.3

Please press RETURN again to confirm your demand.

Would you like to borrow a book via inter-library loan?

If so, fill in as much information as possible. To request a report or an article published in a periodical, press PF82, then select PREP and press PF83 again.

Author or Editor: #5&author
Title or Conference Title: #4&title.1
#4&title.2
Place of Publication: #5&place
Name of Publisher: #5&publisher
Year of Publication: #1&date
(If you do not know the year, please send us a copy of your reference.)
ISBN: #2&isbn
Comments: #5&comment.1
#4&comment.2
Would you like to borrow a report or an article published in a periodical via inter-library loan?

If so, fill in as much information as possible. To request a book or a standard, press PF02, then select BOOK and press PF03 again.

Author(s): &author
Title: &title.1
Report Number: &reportnumber
Place of Publication: &place
Name of Publisher: &publisher
Title of Periodical: &periodical
Volume: &volume Number: &vnumber Pages: &pages
Year of Publication: &year
(If you do not know the year, please send us a copy of your reference.)
Comments: &comment.1

You would like to borrow the following book via inter-library loan.

Author or Editor: &author
Title or Conference Title: &title.1

You would like to borrow the following report or article published in a periodical via inter-library loan.

Author(s): &author
Title: &title.1
Report Number: &reportnumber
Place of Publication: &place
Name of Publisher: &publisher
Title of Periodical: &periodical
Volume: &volume Number: &vnumber Pages: &pages
Year of Publication: &year
Comments: &comment.1

You want all references to either the formulated query
or another search term, enter the operator --> +
You want the references to both the formulated query
and another search term, enter the operator --> *
You want the references to the formulated query and wish to exclude
one or more search terms, enter the operator --> -
You want the references where there are exactly n-1 words between
the last search term and another one (n : number of dollar signs,
where the search terms must be adjacent), enter the operator --> $ ($
The formulated query is final, enter --> -

The priority of operators is as follows:
(lowest) $<"*"<"-"<"+$<"."

Any number of "*" before each operator and any number of "(" after each
operator are accepted, the missing parentheses at the beginning and at
the end of the query formulation will be added by ISIS.

ENDPRE

RETURN Leave and Continue, PF81 More Help, PF83 Leave

.CVF HELP % CONT

J 666.

The CERN preprints and reports data base contains the library holdings of
Preprints from 1980 onwards,
CERN preprints from 1978 onwards,
Reports from 1977 onwards;
and Progress reports from 1976 onwards.

ISIS will search the data base for references which you have specified by
means of "search terms".

ISIS is driven by screens. You may enter information in the input fields.

The available keys and their functions are indicated on the bottom line.
TAB moves the cursor to the next input field.
RETURN displays the next screen or redisplay the present screen.
P81 provides help information.
P82 goes back to the beginning of ISIS.
P83 quits ISIS or the help facility.
P85 submits the prepared query for unsorted output.
P86 submits the prepared query for sorted output.
P87 displays, within the help facility, the previous help page.
P88 displays, within the help facility, the next help page.
Note, that the next page is an input field and may be modified.

VM/CMS ISIS (Mainframe Version)
RETURN displays the next screen or redisplay the present screen.
PFO provides help information.
PFO goes back to the beginning of ISIS.
PFO3 quits ISIS or the help facility.
PFO submits the prepared query for unsorted output.
PFO submits the prepared query for sorted output.
PFO7 displays, within the help facility, the previous help page.
PFO8 displays, within the help facility, the next help page.

Note. to that the next page is an input field and may be modified.

;CONFBE
;SREPST

.B
RETURN Continue, PFO MORE Help, PFO3 Leave
.CVF HELP % CONT
.J & &
HELP Search Terms:

There are two types of search terms, unprefixed and prefixed.

An unprefixed search term is any significant word in the title or added keyword. Examples: HADRONIC, GUT etc.

A prefixed search term is, for example, an author's name, a report number, a conference place etc. To search for a prefixed term, type a two-letter prefix followed by "-" before the term, e.g. AU-VAN Hove L$ for the author Van Hove, Leon. Common prefixes are:

Author's name : AU Example : AU-MILLER$
Collaboration : CO Example : CO-BONN$
Experiment no. : EX Example : EX-DA$
Conference place : PL Example : PL-CAMBRIDGE, MA$
Report and shelf no. : RN Example : RN-CERN 83-02$

If you do not know the exact spelling of a search term, type the first few known characters of that search term truncated by $ (e.g. COLO$).
All search terms beginning with these characters will then be found.

;EPREPST
;SBOOKST

.B
RETURN Leave, PFO MORE Help, PFO3 Leave
.CVF HELP % CONT
.J & &
HELP Search Terms:

There are two types of search terms, unprefixed and prefixed.

An unprefixed search term is any significant word in the title or added keyword. Examples: HADRONIC, GUT etc.

A prefixed search term is, for example, the place, or the opening date of a conference. To search for a prefixed term, type a two-letter prefix followed by "-" before the term, e.g. PL-LES ARCS (or PL-ARC$) for the place Les Arcs. Common prefixes are:

Conference code : CC Example : CC-LESARCS86$8$
Conference date : DA Example : DA-86-8$
Conference place : PL Example : PL-LES ARCS$
Subject code : SU Example : SU-82$
Type of meeting : ME Example : ME-8$

If you do not know the exact spelling of a search term, type the first few known characters of that search term truncated by $ (e.g. COLO$).
All search terms beginning with these characters will then be found.

;ECONFST
;SREPST

.B
RETURN Forward, PFO3 Leave, PFO7 Backward, PFO8 Forward
.CVF % CONT % % PREV FORW
.J & &
The HELP facility is divided into pages. Each page has a two digit number.

1.1 Contents of the PREP database

1.2 How to use ISIS (general remarks)

1.3 Options of ISIS

1.4 Further information and help

1.5 Search terms (introduction)

1.6 Search terms (coding conventions)

1.7 Search terms (peculiarities)

1.8 Operators (Boolean operators)

1.9 Operators (the text search operators " ")

1.10 Operators (example and priorities)

2.0 Operators and parenthesis (example)

2.1 Operators and parenthesis

2.2 Operators and parenthesis

2.3 Operators and parenthesis (example)

2.4 Prefixes of search terms (index)

2.5 Prefixes of search terms (AC, AU, DP, DA)

2.6 Prefixes of search terms (CC, CO, CT)

2.7 Prefixes of search terms (PL, PA)

2.8 Prefixes of search terms (EX, RN, SU)

2.9 Prefixes of search terms (TY, YR)

2.8 Prefixes of search terms (SIU)

You may select the next page ----> #26&mpg (redisplay the page index = 8)

;EPREP01
;EPREP11

RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CVF % CONT % % PREV FORW

J & & &
1.1 Contents of the PREP database

The database is installed under COS/ISIS on the central IBM computers. COS/ISIS is an information storage and retrieval system distributed by UNESCO and has a large number of users worldwide. The procedure to search the database is called ISIS.

The database contains the

Library holdings of preprints from 1988 onwards,
CERN preprints from 1978 onwards,
Reports from 1977 onwards, and
Progress reports from 1976 onwards.

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

Note that earlier CERN catalogues of preprints and reports are also available on microfiche.

;C
You may select the next page ----> #26&mpg (redisplay the page index = 8)

;EPREP11
;EPREP12

RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CVF % CONT % % PREV FORW

J & & &
1.2 How to use ISIS (general remarks)

ISIS is driven by function keys and by screens on which you may change or specify your input. The TAB key will move the cursor to the next input field. The bottom line of each screen indicates which function keys are available for that screen.

The first screen determines the name of the database you would like to search and the mode of the dialogue with ISIS. You may return to the first screen by pressing PF02 when indicated.

ISIS provides two modes of dialogue. The first one, which is intended for beginners, prompts the user for only one search term and one operator at the time. The query must be successively formulated and a maximal checking session must be performed before execution, the query formulated so far will be displayed, and there is a last chance for modifications. The second mode, intended for advanced users of ISIS, skips the screening phase for the search term and one operator and goes immediately to the screen where a whole query may be formulated. It is then the responsibility of the user that his query is correct.

;C
You may select the next page ----> #26&mpg (redisplay the page index = 8)

;EPREP12
;EPREP13

RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CVF % CONT % % PREV FORW

J & & &
1.3 Options of ISIS (first part)

You may quit ISIS or the help facility by pressing PF03. You may continue by pressing RETURN. Most screens have a HELP facility which is available through PF01. PF00 or PF08 will, within the help facility, display the previous or next page, respectively. Note, that the next page is an input.

VM/CMS ISIS (Mainframe Version)
field and may be modified.

When you have answered all of the prompts a final query is prepared. You can correct typing mistakes before submitting the query for search. If you mistyped an answer, press PF02 to restart ISIS. In the same way you may select another database. You may have unsorted output (chronological order, PF05), or output sorted by main heading (PF06).

The unsorted output is stored as VSP01 LISTING A1 (error diagnostics if the search was unsuccessful), the sorted output as VSP04 LISTING A1. It is printed on 140 character line, 66 characters/line, 120 lines/page. Results are automatically displayed by the program XEDIT. It is your responsibility to provide enough space on your A disk.

.C
You may select the next page ----> #2&npg (redisplay the page index = 0)
.;EPREP13
.;SPREP14
.B
RETURN Forward, PF83 Leave, PF87 Backward, PF88 Forward
.;CYF % CONT % % PREV FORW
.J & &
1.4 Further information and help .................. PREP page number : 14

When you have found through ISIS the references which interest you, you may consult them in the library or borrow them from the library.

If you have any comment, suggestion or criticism concerning the search procedure ISIS, please send a message to W. Simon, ISIS@DX AT GEN, or call 4953.

More general questions concerning the results of the search should be addressed to the library, tel.2444.

.C
You may select the next page ----> #2&npg (redisplay the page index = 0)
.;EPREP15
VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

.REXX EXEC

.;SPREP15
.B
RETURN Forward, PF83 Leave, PF87 Backward, PF88 Forward
.;CYF % CONT % % PREV FORW
.J & &
1.5 Search Terms (introduction) .................. PREP page number : 15

Search Terms and Operators.
For fast retrieval, indexes are provided to "search terms" in the data base. ISIS will search the data base for references specified by search terms. There are two types of search terms, unprefixd and prefixed.

An unprefixd search term is any significant word in the title of a publication, in a series title or in a conference title. In addition keywords (title augmentation), which have been assigned since mid 1983, are unprefixd search terms. A significant word is any string of alphanumeric characters, excluding such words as THE, AND, FROM etc.

Examples: HADRON, HADRONIC, QUT, etc.

A prefixd search term is an author's name, a report number a conference place etc. To search for a prefixd term, type the two letter prefix followed by "*" before the term, e.g. AU-VAN Hove L$ for the author Van Hove, Leon. No spaces are allowed before or after the "*". For a list of all available prefixes see below.

.C
You may select the next page ----> #2&npg (redisplay the page index = 0)
.;EPREP15
.;SPREP16
.B
RETURN Forward, PF83 Leave, PF87 Backward, PF88 Forward
.;CYF % CONT % % PREV FORW
.J & &
1.6 Search terms (coding conventions) ............... PREP page number : 16

To retrieve various forms of a search term, truncate, i.e. type the first few significant characters followed by "*". For example, NUCLEI$ will find NUCLEaire, nuclear, nucleI, nucleon, nucleons etc. It is strongly recommended to use truncation initially.

Coding conventions in search terms, examples:
Greek characters are expressed in English words.

E.g. search for ............... ALPHA
Consider apostrophes and hyphens in author names and conference places as
blanks.
E.g. Cecile DeWitt-Morette search for ............... AU-DEWITTMORETTE C$
or .................. AU-DEWITT

Consider apostrophes and hyphens in titles as blanks.

e.g. high-energy search for .......... HIGH · ENERGY

e.g. Muller with umlaut search for .......... AU-MULLER

.C
You may select the next page ----> #2&npq (redisplay the page index = 8)
"EPREP16"
"SPREP17"
"B"
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
"CPF 1 % CONT 1 % PREV FORW"
"J 664"

Search terms (peculiarities), Operators ........ PREP page number : 17

Search terms, which contain the special characters +, -, ( ), # must be
included in quotation marks (e.g. "SUZUKI").

ISIS will add to every search term a period as the first character.

Up to twenty search terms may be combined by the logical operators OR
"A", AND "B", and AND NOT "A", or by the full text search operators
PERIOD ".", or DOLLAR "$".

No two operators may be adjacent to each other with the exception of
repeats of "A", or "B", which may not, however, be mixed together.

.C
You may select the next page ----> #2&npq (redisplay the page index = 8)
"EPREP17"
"SPREP18"
"B"
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
"CPF 1 % CONT 1 % PREV FORW"
"J 664"

Operators (Boolean operators) ................. PREP page number : 18

The operator "A" will find all references to either the search term to

VM/CMS ISIS (Mainframe Version)

the left or the search term to the right of the OR. It broadens the scope
of the search.

The operator "A" is commutative, i.e. A&B is equivalent to B&A.

The operator "A" will find all references to both the search term to the
left and the search term to the right of the AND. It narrows the scope of
the search.

The operator "A" is commutative, i.e. A&B is equivalent to B&A.

The operator "A" will find all references to the search term to the left
and not to the search term to the right of the AND NOT. It narrows the
scope of the search.

The operator "A" is NOT commutative, i.e. A-B is NOT equivalent to B-A.

C
You may select the next page ----> #2&npq (redisplay the page index = 8)
"EPREP18"
"SPREP19"
"B"
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
"CPF 1 % CONT 1 % PREV FORW"
"J 664"

Operators (the text search operator " ") .... PREP page number : 19

The operator " ",", will find all references where there are at most n-1
words between the two search terms, n being the number of periods. Note,
that ISIS will insert the necessary blanks for you. Compared to the
operator "A", the operator " ",", further narrows the scope of the search.

The operator " A B" is commutative, i.e. A . B is equivalent to B . A. E.g.
A . B means that A and B must be adjacent, A . B means that there may
be up to one word between A and B, etc. Unfortunately, the " "," operator
will not work for added keywords. In this case all references which
contain the added keywords A and B will be retrieved. The operator " "
should not be used between prefixed search terms. The resulting expression
will in general fail to retrieve references.
The operator " $ " will find all references where there are exactly n-1 words between the two search terms, n being the number of dollars. Note, that ISIS will insert the necessary blanks for you. Compared to the operator " $ ", the operator " * " further narrows the scope of the search. The operator " $ " is commutative, i.e., A $ B is equivalent to B $ A. E.g., A $ $ means that A and B must be adjacent, A $ $ $ B means that there must be exactly one word between A and B, etc. Unfortunately, the " $ " operator will not work for added keywords. In this case all references which contain the added keywords A and B will be retrieved. The operator " $ " should not be used between prefixed search terms. The resulting expression will in general fail to retrieve references.

Example:
The query ATOMIC $ $ $ ENERGY retrieves the references:

ATOMIC ENERGY levels
ATOMIC ENERGY agency
ATOMIC ground state ENERGY
ATOMIC and ionic ENERGY levels
ENERGY shifts in ATOMIC k x rays
high-ENERGY collisions with ATOMIC nuclei

whereas the query ATOMIC $ $ $ $ ENERGY retrieves only the last four references. Note the inversion of the search terms in accordance with the commutative law.

The priority of operators is as follows:

(lowest) " $ ", " $ $ ", " $ $ $ ", etc., and

(highest) " $ $ $ $ ", " $ $ $ $ $ ", etc.

Example:

AU+G0$=AU+WE1SSKOF$=YR+84-NUCL$ . . . ENERG$

is equivalent to:

AU+G0$=AU+WE1SSKOF$=YR+84-(NUCL$ . . . ENERG$)

However, the priority may be changed by parentheses, e.g.:

((AU+G0$=AU+WE1SSKOF$=YR+84)-(NUCL$ . . . ENERG$))

Note that the redundant parentheses are accepted.

ISIS accepts any number of " $ " before each operator and any number of " ( " after each operator. It will add the missing parentheses at the beginning and the end of the query formulation.
2.3 Operators and parentheses (example) ............ PREP page number : 23

Example:

((AU=GO$+$AU=W150FF$)*VR=8$)-(NUCL$ . . ENERG$)

would be input to ISIS as follows:

first search term:

second search term:

third search term:

fourth search term:

fifth search term:

terminate query:

Note, that ISIS supplies the first "(" and the last ")". ISIS adds a period at the beginning of each search term. The query generated by ISIS looks as follows:

((AU=GO$+$AU=W150FF$)*.VR=8$)-(NUCL$ . . ENERG$)

.J & &

You may select the next page ----- #2&npg (redisplay the page index = 8)

.J & &

2.4 Prefixes of search terms (index) ............ PREP page number : 24

Prefixes of search terms. List of search terms with their prefixes display page number

Accelerator : AC Example : AC=CERN ISR 25
Author's name : AU Example : AU=MILLER R B 25
Conference code : CC Example : CC=LEA0RCS8680$ 26
CERN paper : CP Example : CP=EP 25
CERN work : CW Example : CW=SPS 25
Collaboration : CD Example : CD=LEONN$ 26
Conference date : DA Example : DA=82-87$ 26
Conference place : PL Example : PL=FLORENCE 27
Corporate author : CA Example : CA=CERN$ 27
Experiment no. : EX Example : EX=UAI 28
Report and shelf no. : RN Example : RN=CERN 84-8$ 26
Subject category : SU Example : SU=AB 38

VM/CMS ISIS (Mainframe Version)

.J & &

You may select the next page ----- #2&npg (redisplay the page index = 8)

.J & &

2.5 Prefixes of search terms (AC, AU, CP, CW) ........ PREP page number : 25

Accelerator (prefix AC) gives the name of the accelerator at which an experiment was carried out, the name has at most 18 characters.
Example : AC=CERN ISR

Author's name (prefix AU) gives the surname of the author followed by one or more initials without full stops, separated by spaces. Names may consist of more than one word. Try various forms when in doubt, or use truncation. Note that only initials are available, no full first names.
Examples : AU=MILLER$, AU=MILLER C$, AU=MILLER C A
Examples : AU=+YAN HOYE L$, AU=TRAN THANH VAN J$, AU=TRANS

CERN paper (prefix CP) gives the division responsible for the publication.
Example : CP=EP

CERN work (prefix CW) gives the division where the work is carried out or the division of the CERN staff member working outside CERN.
Example : CW=SPS

.J & &

You may select the next page ----- #2&npg (redisplay the page index = 8)

.J & &

2.6 Prefixes of search terms (CC, CD, DA) ............. PREP page number : 26

The conference code (prefix CC) is constructed from the place name, in English if there is an English form, and without spaces or other punctuation, followed immediately by the six-digit opening date in the order year, month, day. Note that conference codes have been implemented

VM/CMS ISIS (Mainframe Version)
systematically only since week 41 of 1986.
Particularly if you do not know the exact date, truncate at first.
If two or more conferences in the same place start on the same day, the
second and subsequent dates are distinguished by the suffixes a, b ...
Examples : CC-LEARNCS86#8316A, CC-LEARNCS87#4

Collaboration (prefix CO) gives the name of the collaboration. The search
term must be specified with right hand truncation.
Example : CO-BOHEMIAN

Conference date (prefix DA) must be specified in the order year-month-day
and is the opening date of the conference. Note that conference dates are
only implemented between week number 29 of 1982 and number 41 of 1986.
Example : DA-82-076

.C
You may select the next page -----> #2&npg (redisplay the page index = 0)
;EPREP26
;SPREP27

2.7 Prefixes of search terms (PL, CA) .................. PREP page number : 27
Conference place (prefix PL) gives the English name of the conference
place, if it exists, otherwise it is the name of the conference place in the
official language of the country. If it is necessary to distinguish
between two places of the same name, the name of the state or country is
added. For conferences held at CERN use PL-CERN#. Note that conference
dates are only implemented between week number 29 of 1982 and number 41
of 1986.
Examples for English names : PL-FLORENCE, PL-MUNICH, PL-CAMBRIDGE, MA
Examples for original names : PL-LES HOUCHES, PL-SANTIAGO DE COMPOSTELA

Corporate author (prefix CA) gives the name of the institute, laboratory,
university etc., responsible for the content of the document. The name of
the body is given in its own language. Several forms of the name
appear on the document the most prominent one is used. The English form
is used for an international organisation.
Example : CA-CERN

.C
You may select the next page -----> #2&npg (redisplay the page index = 0)
;EPREP27

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

;SPREP28
.B
RETURN Forward, PF83 Leave, PF87 Backward, PF88 Forward
.CYF % CONT % % PREY FORM
.J &&
2.8 Prefixes of search terms (EX, RN, SW) ............ PREP page number : 28
Experiment number (prefix EX) gives the number of the experiment(s)
described in the document.
Example : EX-U1

Report and shelf numbers (prefix RN) gives the identification number of
the document, usually in the form of the acronym of the issuing body
followed by a number.
Examples : RN-CERN PRE 82-84B, RN-CERN 83-82

Status week (prefix SW) gives since week number 29 of 1982 the week when
the reference was entered into the data base. The week is immediately
preceded by the year when the reference was entered into the data base.
For older references only the year when the reference was entered into the
data base is available.
Examples : SW-8326, SW-83

.C
You may select the next page -----> #2&npg (redisplay the page index = 0)
;EPREP28
;SPREP29
.B
RETURN Forward, PF83 Leave, PF87 Backward, PF88 Forward
.CYF % CONT % % PREY FORM
.J &&
2.9 Prefixes of search terms (TY, YR) .................. PREP page number : 29
Type of subject matter (prefix TY) gives the type of the contents of the
document, i.e.
AB : abstract  BI : bibliography
DA : data  DI : dictionary
FO : formulae  HI : history
LE : lecture  MA : mathematical tables
PA : patents  PO : popular account
PR : progress report  RE : review
SP : specification  ST : standards
SU : survey  TA : tables
TH : thesis  TP : text, programmed

VM/CMS ISIS (Mainframe Version)
TR : text, research level
TU : text, university level
Example : TR=EI
Year of publication (prefix YR) gives the year of publication of the document.
Example : YR=83

You may select the next page ----> #2&mpg (redisplay the page index = 0)
;EPREP23
;SPREP38
.B
RETURN Forward, PF83 Leave, PF87 Backward, PF88 Forward
.CVF % % CONT % % PREV FORW
.J & & &
3.0 Prefixes of search terms (SU) ............... PREP page number : 30

Subject categories (prefix SU) should be seen more as a series of user
categories than as a strict subject classification.

AA : particle physics, review
AB : particle physics, experimental results
AC : particle physics, phenomenology
AD : particle physics, theory and experimental results
BB : nuclear physics, theory and instrumentation
CC : general theoretical physics and instruments in particle and nuclear physics
EE : accelerators and storage rings
FF : health physics and general safety
GG : computers and data processing
HH : mathematics
JJ : physics and astronomy (general)
KK : chemistry
LL : engineering
MM : information transfer and management
NN : other aspects of science
OO : geography, history, biography
PP : social sciences and sociology
QQ : nuclear physics
RR : other subjects
Example : SU=AB

You may select the next page ----> #2&mpg (redisplay the page index = 0)
;EPREP23
;SBOKDI
.B
RETURN Forward, PF83 Leave, PF87 Backward, PF88 Forward
.CVF % % CONT % % PREV FORW
.J & & &
The HELP facility is divided into pages. Each page has a two digit number

VM/CMS ISIS (Mainframe Version)
You may select the next page ----> #2&npg (redisplay the page index = 8)

C

RETURN Forward, PFB3 Leave, PFB7 Backward, PFB8 Forward
.CVF % % CONT % % PREV FORW
.J & & &

1.2 How to use ISIS (general remarks) ............... BOOK page number : 12

ISIS is driven by function keys and by screens on which you may change certain input fields to specify your query. The TAB key will move the cursor to the next input field. The bottom line of each screen indicates which function keys are available for that screen.

The first screen determines the name of the database you would like to search and the mode of the dialogue with ISIS. You may return to the first screen by pressing PF82 when indicated.

ISIS provides two modes of dialogue. The first one, which is intended for beginners, permits the user for only one search term and one operator at the time. The query must be successively formulated and a maximal checking is done. Before execution, the query formulated so far will be displayed, and there is a last chance for modifications.

The second mode, intended for advanced users of ISIS, skips the screens prompting for one search term and one operator and goes immediately to the screen where a whole query may be formulated. It is then the responsibility of the user that his query is correct.

C

You may select the next page ----> #2&npg (redisplay the page index = 8)

RETURN Forward, PFB3 Leave, PFB7 Backward, PFB8 Forward
.CVF % % CONT % % PREV FORW
.J & & &

1.3 Options of ISIS (second part) .................... BOOK page number : 13

You may quit ISIS or the help facility by pressing PFB3. You may continue by pressing RETURN. Most screens have a HELP facility which is available through PFB1. PFB7 or PFB8 will, within the help facility, display the previous or next page, respectively. Note, that the next page is an input field and may be modified.

When you have answered all of the prompts a final query is prepared. You

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

DISPLAY 246

can correct typing mistakes before submitting the query for search. If you mistyped an answer, press PFB2 to restart ISIS. In the same way you may select another data base. You may have unsor ted output (chronological order, PFB6), or output sorted by main heading (PFB7).

The unsorted output is stored as VSP81 LISTING A1 (error diagnostics if the search was unsuccessful), the sorted output as VSP94 LISTING A1, it is formatted to be printed on W paper (CTN, 68 characters/line, 64 lines/page). Results are automatically displayed by the program XEDIT. It is your responsibility to provide enough space on your A disk.

C

You may select the next page ----> #2&npg (redisplay the page index = 8)

RETURN Forward, PFB3 Leave, PFB7 Backward, PFB8 Forward
.CVF % % CONT % % PREV FORW
.J & & &

1.4 Further information and help .................... BOOK page number : 14

When you have found through ISIS the references which interest you, you may consult them in the library or borrow them from the library.

If you have any comment, suggestion or criticism concerning the search procedure ISIS, please send a message to W. Simon, ISIS@DX AT GEN, or call 4953.

More general questions concerning the results of the search should be addressed to the library, tel.2444.

C

You may select the next page ----> #2&npg (redisplay the page index = 8)

RETURN Forward, PFB3 Leave, PFB7 Backward, PFB8 Forward

VM/CMS ISIS (Mainframe Version)
.CYou may select the next page ----> #26&pg (redisplay the page index = 8)
;EBOOK16
;SBOK16
 .B
RETRUM Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CYF % CONT % % PREV FORW
 .J & & 
1.6 Search terms (coding conventions) .......... BOOK page number : 16

To retrieve various forms of a search term, truncate, i.e. type the first
few significant characters followed by "$$. For example, NUCLEAT$ will
find NUCLEATE, NUCLEAR, NUCLEI, NUCLEON, NUCLEONS etc.

Coding conventions in search terms, examples:
Greek characters are expressed in English words.

.e.g. search for = ALPHA
Consider apostrophes and hyphens in author names and conference places as
blanks.
.e.g. Cecile DeWitt-Morette search for = AU=DEWITTMORETTE$ or = AU=DEWIT$T#
.e.g. L'Aquila search for = PL=L'AQUILA
Consider apostrophes and hyphens in titles as blanks.

VM/CMS ISIS (Mainframe Version)
The operator "*" is commutative, i.e. A*B is equivalent to B*A.

The operator "*" will find all references to both the search term to the left and the search term to the right of the AND. It narrows the scope of the search.

The operator "*" is commutative, i.e. A*B is equivalent to B*A.

The operator "*" will find all references to the search term to the left and not to the search term to the right of the AND NOT. It narrows the scope of the search.

The operator "*" is NOT commutative, i.e. A*B is NOT equivalent to B*A.

You may select the next page ----> #Z&mpg (redisplay the page index = 0)
;EBOOK18
;SBOOK13
.B
RETURN Forward, PF#3 Leave, PF#7 Backward, PF#8 Forward
.CYF % % CONT % % PREV FORW
.J & & &
1.5 Operators (the text search operator "."). .... BOOK page number : 19

The operator "." will find all references where there are at most n-1 words between the two search terms, n being the number of periods. Note, that ISIS will insert the necessary blanks for you. Compared to the operator ".", the operator "." further narrows the scope of the search. The operator "." is commutative, i.e. A*B is equivalent to B*A. E.g., A . B means that A and B must be adjacent, A . B means that there may be up to one word between A and B, etc. Unfortunately, the "." operator will not work for added keywords. In this case all references which contain the added keywords A and B will be retrieved. The operator "." should not be used between prefixed search terms. The resulting expression will in general fail to retrieve references.

You may select the next page ----> #Z&mpg (redisplay the page index = 0)
;EBOOK19

Mainframe VM/CMS ISIS (Mainframe Version)

;SBOOK20
.B
RETURN Forward, PF#3 Leave, PF#7 Backward, PF#8 Forward
.CYF % % CONT % % PREV FORW
.J & & &
2.6 Operators (the text search operator " "$") .... BOOK page number : 20

The operator "$" will find all references where there are exactly n-1 words between the two search terms, n being the number of dollars. Note, that ISIS will insert the necessary blanks for you. Compared to the operator ".", the operator "$" further narrows the scope of the search. The operator "$" is commutative, i.e. A $# B is equivalent to B $# A. E.g., A $# B means that A and B must be adjacent, A $# B means that there must be exactly one word between A and B, etc. Unfortunately, the "$" operator will not work for added keywords. In this case all references which contain the added keywords A and B will be retrieved. The operator "$" should not be used between prefixed search terms. The resulting expression will in general fail to retrieve references.

You may select the next page ----> #Z&mpg (redisplay the page index = 0)
;EBOOK20
;SBOOK21
.B
RETURN Forward, PF#3 Leave, PF#7 Backward, PF#8 Forward
.CYF % % CONT % % PREV FORW
.J & & &
2.1 Operators (examples and priorities) ......... BOOK page number : 21

Example: The query ATOMIC ... ENERGY retrieves the references:

- ATOMIC ENERGY levels
- ATOMIC ENERGY agency
- ATOMIC ground state ENERGY
- ATOMIC and ionic ENERGY levels
- ENERGY shifts in ATOMIC k x rays
- high-ENERGY collisions with ATOMIC nuclei

whereas the query ATOMIC $# $# ENERGY retrieves only the last four references. Note the inversion of the search terms in accordance with the
The priority of operators is as follows:

(least) "+" = "-" = "*", 
(highest) "$", "$\$", "$\$\$", etc., and 
"\$", "$\$", "$\$\$", etc.

.C
You may select the next page ----> #2&npg (redisplay the page index = 0)
;EBOOK21
;SBOOK22
.B
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
;CFY \% CONT \% \% PREV FORW
.J & &
2.2 Operators and parentheses .................. BOOK page number : 22

Example:

AU+GO$+AU=WEISSKOPF$+YR=84=NUCL$ ... ENERG$

is equivalent to:

AU+GO$(AU=WEISSKOPF$+YR=84=NUCL$ ... ENERG$)

However, the priority may be changed by parentheses, e.g.:

(AU+GO$(AU=WEISSKOPF$)+YR=84=NUCL$ ... ENERG$)

Note that the redundant parentheses are accepted.

ISIS accepts any number of "(" before each operator and any number of "(" after each operator, it will add the missing parentheses at the beginning and the end of the query formulation.

.C
You may select the next page ----> #2&npg (redisplay the page index = 0)
;EBOOK23
;SBOOK23
.B
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
;CFY \% CONT \% \% PREV FORW
.J & &
2.3 Operators and parentheses (example) ........... BOOK page number : 23

Example:

(AU+GO$(AU=WEISSKOPF$)+YR=84=NUCL$ ... ENERG$)

would be input to ISIS as follows:

- first search term : AU+GO$
- second operator : =
- second search term : AU=WEISSKOPF$
- second operator : =
- third search term : YR=84
- third operator : =
- fourth search term : NUCL$
- fourth operator : =
- fifth search term : ENERG$
- terminate query :

Note, that ISIS supplies the first "(" and the last ")". ISIS adds a period at the beginning of each search term. The query generated by ISIS looks as follows:

((AU+GO$+(AU=WEISSKOPF$)+YR=84=NUCL$ ... ENERG$))

.C
You may select the next page ----> #2&npg (redisplay the page index = 0)
;EBOOK23
;SBOOK24
.B
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
;CFY \% CONT \% \% PREV FORW
.J & &
2.4 Prefixes (index) ......................... BOOK page number : 24

Prefixed Search Terms.
List of search terms with their prefixes ............... display page number

Author's name : AU Example : AU=MILLER R B 26
Conference date : DA Example : DA=82-07$ 27
Conference place : PL Example : PL=FLORENCE 27
Corporate author : CA Example : CA=CERN 26
Report and shelf no. : RN Example : RN=CERN 83-82$ 28
Standard no. : ST Example : ST=ISO 885 28
Type of subject matter : TV Example : TV=81 29
Year of publication : YR Example : YR=83 28
Corporate author (prefix CA) gives the name of the institute, laboratory, university, etc., responsible for the content of the document. The name of the body is given in its own language. If several forms of the name appear on the document, the most prominent one is used. For an international organisation, the English form is used.

Examples: CA=CERN#, CA=INTERNATIONAL ATOMIC ENERGY AGENCY VIENNA

Conference place (prefix PL) gives the English name of the conference place in the official language of the country. If it is necessary to distinguish between two places of the same name, the name of the state or country is added. For conferences held at CERN use PL=CERN.

Example for English names: PL=UNICHI
Examples for original names: PL=LES HOUCHES, PL=SANTIAGO DE COMPOSTELA

Standard number (prefix ST) gives the identification number of the standard, usually in the form of the acronym of the issuing body followed by a number.

Example: ST=AFNOR NF C 18-415#, ST=JEC 68-2-27#
Type of subject matter (prefix TY) gives the type of the contents of the document, i.e.

- AB: abstract
- BI: bibliography
- DA: date
- DI: dictionary
- FO: formulae
- HI: history
- LE: lecture
- MA: mathematical tables
- PA: patents
- PD: popular account
- PR: progress report
- RE: review
- SP: specification
- ST: standards
- SU: survey
- TA: tables
- TH: thesis
- TP: text, programmed
- TR: text, research level
- TS: text, school level
- TU: text, university level

Example: TY=BI

The HELP facility is divided into pages. Each page has a two digit number.

1.1 Contents of the CONF data base .............. CONF page number: 11
1.2 How to use ISIS (general remarks) ........... CONF page number: 12
1.3 Options of ISIS ............................. CONF page number: 13
1.4 Further information and help .................. CONF page number: 14
1.5 Search terms (introduction) .................... CONF page number: 15
1.6 Search terms (coding conventions) .......... CONF page number: 16
1.7 Search terms (peculiarities) .................. CONF page number: 17
1.8 Operators (Boolean operators) ................ CONF page number: 18
1.9 Operators (the text search operator "*" ) .... CONF page number: 19
1.10 Operators (the text search operator " !") .... CONF page number: 20
1.11 Operators (Examples and priorities) ......... CONF page number: 21
1.12 Operators and parenthesis ..................... CONF page number: 22
1.13 Operators and parenthesis (example) ....... CONF page number: 23
1.14 Prefixes of search terms (index) .......... CONF page number: 24

VM/CMS ISIS (Mainframe Version)
The first screen determines the name of the data base you would like to search and the mode of the dialogue with ISIS. You may return to the first screen by pressing PF02 when indicated.

ISIS provides two modes of dialogue. The first one, which is intended for beginners, prompts the user for only one search term and one operator at the time. The query must be successively formulated and a maximal checking is done. Before execution, the query formulated so far will be displayed, and there is a last chance for modifications. The second mode, intended for advanced users of ISIS, skips the screens prompting for search terms and one operator and goes immediately to the screen where a whole query may be formulated. It is then the responsibility of the user that his query is correct.

You may select the next page ———> #2&mpg (redisplay the page index = 0)
1.2 ECONF12
1.3 Options of ISIS (first part) ................. CONF page number : 13
You may quit ISIS or the help facility by pressing PF03. You may continue by pressing RETURN. Most screens have a HELP facility which is available through PF01. PF07 or PF08 will, within the help facility, display the previous or next page, respectively. Note, that the next page is an input field and may be modified.

When you have answered all of the prompts a final query is prepared. You can correct typing mistakes before submitting the query for search. If you also want an answer, press PF02 to restart ISIS. In the same way you may select another data base. You may have unsorted output (chronological order, PF05), or output sorted by conference code (PF06).

The unsorted output is stored as YSP01 LISTING A1 (error diagnostics if the search was unsuccessful), the sorted output as YSP04 LISTING A1. It is formatted to be printed on A paper (CTN, 68 characters/line, 64 lines/page). Results are automatically displayed by the program XEDIT. It is your responsibility to provide enough space on your A disk.

You may select the next page ———> #2&mpg (redisplay the page index = 0)
1.4 Further information and help ................. CONF page number : 14
When you have used CONF to find the conference which interests you, you may change to PREP and use the conference code (CC+) to search for references to any preprints submitted to the conference.

If you have any comment, suggestion or criticism concerning the search procedure ISIS, please send a message to W. Simon, ISIS@X@GEN, or call 4953.

More general questions concerning the results of the search should be addressed to the library, tel.2444.
such words as THE, AND, FROM, etc.

Examples: HADRON, HADRONIC, GUT, etc.

A prefixed search term is a conference place, date etc.
To search for a prefixed term, type the two-letter prefix followed by "=" before the term, e.g. PL-PARIS for the place Paris.
No spaces are allowed before or after the "=".
For a list of all available prefixes see below.

.C
You may select the next page ----> #2&npg (redisplay the page index = 0)
:ECNF15
:SCNF16
.B
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CYF % CONT % PREV FOWR
.J & &
1.6 Search terms (coding conventions) .......... CONF page number : 16

To retrieve various forms of a search term, truncate, i.e. type the first few significant characters followed by "$". For example, NUCLES will find NUCLEARE, NUCLEAR, NUCLEI, NUCLEON, NUCLEONS etc.
It is strongly recommended to use truncation initially.

Coding conventions in search terms, examples:
Greek characters are expressed in English words.
.e.g. search for .............. ALPHA
Ignore apostrophes in conference places.
.e.g. L'Aquila search for .............. PL-LAQUILA
Conference places with more than one word can be searched for as a whole,
or by parts.
.e.g. Les Arcs search for .............. PL-LES ARCS
or .............. PL-ARES
Consider apostrophes and hyphens in titles as blanks.
.e.g. high-energy search for .............. HIGH-ENERGY
Ignore all accents and other diacritical marks.
.e.g. Mossbauer (with.realpath) search for .............. MOSSBAUER

.C
You may select the next page ----> #2&npg (redisplay the page index = 0)
:ECNF16
:SCNF17
.B
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CYF % CONT % PREV FOWR
.J & &

VM/CMS ISIS (Mainframe Version)

Mainframe VM/CMS ISIS

REXX EXEC

DISPLAY 260

1.7 Search terms (peculiarities), Operators .......... CONF page number : 17

Search terms, which contain the special characters +, *, =, (, #, must be
included in quotation marks (e.g. "SU(3)!").

ISIS will add to every search term a period as the first character.

Up to twenty search terms may be combined by the logical operators OR
"+", AND "&", and AND NOT "-", or by the full text search operators
PERIOD ".", or DOLLAR "$".

No two operators may be adjacent to each other with the exception of
repeats of ".", or "$", which may not, however, be mixed together.

.C
You may select the next page ----> #2&npg (redisplay the page index = 0)
:ECNF17
:SCNF18
.B
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CYF % CONT % PREV FOWR
.J & &
1.8 Operators (Boolean operators) ............... CONF page number : 18

The operator "+" will find all references to either the left or the search term to
the right of the OR. It broadens the scope of the search.

The operator "&" is commutative, i.e. A&B is equivalent to B&A.

The operator "&" will find all references to both the search term to the
left and the search term to the right of the AND. It narrows the scope of the
search.

The operator "&" is commutative, i.e. A&B is equivalent to B&A.

The operator "-" will find all references to the search term to the left
and not to the search term to the right of the AND NOT. It narrows the
scope of the search.

VM/CMS ISIS (Mainframe Version)
The operator "-" is NOT commutative, i.e. A-B is NOT equivalent to B-A.

2. Operators (the text search operator " ") .... CONF page number : 19

The operator " ") will find all references where there are at least n-1 words between the two search terms, n being the number of periods. Note, that ISIS will insert the necessary blanks for you. Compared to the operator ",", the operator " ") further narrows the scope of the search. The operator " ") is commutative, i.e. A B is equivalent to B A. E.g. A . B means that A and B must be adjacent, A . B means that there may be up to one word between A and B, etc. Unfortunately, the " ") operator will not work for added keywords. In this case all references which contain the added keywords A and B will be retrieved. The operator " ") should not be used between prefixed search terms. The resulting expression will in general fail to retrieve references.

VM/CMS ISIS (Mainframe Version)

VM/CMS ISIS (Mainframe Version)
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CVF % % CONT % % PREV FORW
.J & & &

2.2 Operators and parentheses ............... CONF page number: 22

Example:
AU-GO$+AU-WEISSKOPF#*YR-84-NUCL$ .. ENERG$

is equivalent to:
AU-GO$+(AU-WEISSKOPF#*YR-84-(NUCL$ .. ENERG$))

However, the priority may be changed by parentheses, e.g.:
((AU-GO$+AU-WEISSKOPF#*YR-84)-(NUCL$ .. ENERG$))

Note that the redundant parentheses are accepted.

ISIS accepts any number of "(" before each operator and any number of "(" after each operator, it will add the missing parentheses at the beginning and the end of the query formulation.

You may select the next page ---> #2&npg (redisplay the page index = 0)
;ECONF22
;SCONF23

RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CVF % % CONT % % PREV FORW
.J & & &

2.3 Operators and parentheses (example) ........... CONF page number: 23

Example:
((AU-GO$+AU-WEISSKOPF#*YR-84)-(NUCL$ .. ENERG$)

would be input to ISIS as follows:

first search term: AU-GO$
first operator: +
second search term: AU-WEISSKOPF#
second operator: *
third search term: YR-84
third operator: 
fourth search term: NUCL$
fifth search term: ...

Note, that ISIS supplies the first "(" and the last ")." ISIS adds a period at the beginning of each search term. The query generated by ISIS looks as follows:

((.AU-GO$+.AU-WEISSKOPF#*_.YR-84)-(_.NUCL$ .. ._.ENERG$))

You may select the next page ---> #2&npg (redisplay the page index = 0)
;ECONF23
;SCONF24

RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CVF % % CONT % % PREV FORW
.J & & &

2.4 Prefixes of search terms (index) ............ CONF page number: 24

Prefix Search Terms.
List of search terms with their prefixes ............ display page number

Conference code: CC Example: CC-LESARCS6803#
Conference date: DA Example: DA-86-83#
Conference place: PL Example: PL-LES ARC#
Language of conference: LA Example: LA-EN
Status week: SW Example: SW-8648
Subject category: SU Example: SU-AZ
Type of meeting: ME Example: ME-S

You may select the next page ---> #2&npg (redisplay the page index = 0)
;ECONF24
;SCONF25

RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CVF % % CONT % % PREV FORW
.J & & &

2.5 Prefixes of search terms (CC, DA) ............ CONF page number: 25

The conference code (prefix CC) is constructed from the place name, in

VM/CMS ISIS (Mainframe Version)
English if there is an English form, and without spaces or other punctuation, followed immediately by the six-digit opening date in the order year, month, day. Particularly if you do not know the exact date, truncate at first. If two or more conferences in the same place start on the same day, the second and subsequent codes are distinguished by the suffixes, a, b ...
Examples: CC+LESARCS868031G, CC+LESARCS8, CC+ERICE8680785A

Conference date (prefix DA) is the opening date of the conference, in the order year-month-day (with hyphens). Examples: DA=86-03-16, DA=86-03#

.C
You may select the next page ---> #2&next (redisplay the page index = 0)
;ECNF25
;SCNF26
.B
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CVF % % CONT % % PREV FORW
.J & & &
2.6 Prefixes of search terms (PL, LA, SW, ME) ...... CONF page number : 26

Conference place (prefix PL) is normally the English name of the town where the conference is held, but may also be the native name. Compound place names may be searched as a whole or by parts. Countries and US States may also be searched. For CERN, use PL=CEHERN Examples: PL=LES HOUCHES, PL=FLORENCE
Examples: PL=CAMBRIDGE#PL=MA, PL=CAMBRIDGE#PL=UK

Language of the conference (prefix LA) is a two-letter code for each of the official languages of the conference.
Examples: LA=EN, LA=FR

Status week (prefix SW) gives the year and week number (since week 03 of 1980) when the reference was entered into the data base (or last altered).
Examples: SW=8603, SW=8648

Type of meeting (prefix ME) distinguishes between Conferences (ME=C) and Schools (ME=S).
Example: ME=S

VM/CMS ISIS (Mainframe Version)

.C
You may select the next page ---> #2&next (redisplay the page index = 0)
;ECNF26
;SCNF27
.B
RETURN Forward, PF03 Leave, PF07 Backward, PF08 Forward
.CVF % % CONT % % PREV FORW
.J & & &
2.7 Prefixes of search terms (SU) ................. CONF page number : 27

Subject categories (prefix SU) should be seen more as a series of user categories than as a strict subject classification.

AZ : particle physics, theory and experimental results
BB : nuclear physics, theory and experimental results
CC : general theoretical physics
DD : experimental techniques and instruments in particle and nuclear physics
EE : accelerators and storage rings
FF : health physics and general
GG : computers and data processing
HM : mathematics
JJ : physics and astronomy (general)
KK : chemistry
LL : engineering
MM : information transfer and management
NN : other aspects of science
PP : social sciences and sociology
QQ : geography, history, biography
RR : other subjects

Example : SU-AZ

.C
You may select the next page ---> #2&next (redisplay the page index = 0)
;ECNF27

VM/CMS ISIS (Mainframe Version)