OS Utilities

Here is a chart of action words and parameters and a list of rules for ten frequently used OS Utilities:

Reminder: Continuation cards for utility control statements (read from the SYSIN DD) must start in column 16 and the previous statement (the one being continued) must contain a non-blank character in column 72.

Utility	Action Words	Parameters	Page
IEHLIST	LISTVTOC	FORMAT, VOL=	2
	LISTPDS	DSNAME=, VOL=	2
	LISTCTLG	CVOL=	2
IEHDASDR	ANALYZE	TODD=, VTOC=, EXTENT=, NEWVOLID=, PURGE=YES, PASSES=	3
	DUMP	FROMDD=, TODD=, BEGIN=, END=	4
	RESTORE	FROMDD=, TODD=, PURGE=YES	5
	LABEL	TODD=, NEWVOLID=	4
IEBGENER			5
IEBCOPY	**C**OPY	**I**NDD=, **O**UTDD=	6
	SELECT	**M**EMBER=	7
	EXCLUDE	**M**EMBER=	7
IEHMOVE	COPY	PDS=, FROM=, FROMDD=, TO=, TODD=	9
IEHPROGM	CATLG	DSNAME=, VOL=	10
	UNCATLG	DSNAME=	11
	SCRATCH	DSNAME=, VOL=, PURGE, VTOC, MEMBER=	12
	RENAME	DSNAME=, VOL=, MEMBER=, NEWNAME=	11
IEBPTPCH	PRINT	TYPORG=PO, MAXNAME=, MAXFLDS=	13
	MEMBER	NAME=	13
	RECORD	FIELD=	13
	PUNCH	TYPORG=PO, MAXNAME=, MAXFLDS=, CDSEQ=, CDINCR=	14
IEBUPDTE	CHANGE	NAME=, UPDATE=INPLACE	15
	ADD	NAME=, LIST=ALL	15
	NUMBER	NEW1=, INCR=	15
	DELETE	SEQ1=, SEQ2=	16
	ENDUP		15
IEBCOMPR	COMPARE	TYPORG=	18
	LABEL	DATA=	18
	EXITS	INHDR=, INTLR=, ERROR=, PRECOMP=	19
IEHINITT	INITT	SER=, NUMBTAPE=, OWNER=, DISP=	19

OS Utilities: The Rules with Examples

IEHLIST

1. IEHLIST needs one DD card for the disk pack you want information from, specifying the whole pack, with any DD name.

   ```
   //DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
   ```

2. Printing a VTOC using IEHLIST requires one control card, and LISTVTOC is the action word. The parameters are: FORMAT, which stands alone; and VOL, which gives the unit type and number of the pack whose VTOC is being listed.

   ```
   //LISTVTOC JOB (accnt),NAME,MSGLEVEL=(1,1)
   //STEP1 EXEC PGM=IEHLIST
   //SYSPRINT DD SYSOUT=A
   //DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
   //SYSIN DD *
      LISTVTOC FORMAT,VOL=3330=D10040
   /*
   //
   ```

3. Listing the directory of a partitioned file using IEHLIST requires one control card, and LISTPDS is the action word. The parameters are: DSNAME (which may be abbreviated DSN), to name the file; and VOL, to give the unit type and pack number the file is on.

   ```
   //LISTLIB JOB (accnt),NAME,MSGLEVEL=(1,1)
   //STEP1 EXEC PGM=IEHLIST
   //SYSPRINT DD SYSOUT=A
   //DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
   //SYSIN DD *
      LISTPDS DSNAME=LOADLIB,VOL=3330=D10040
   /*
   //
   ```

4. Listing the catalog using IEHLIST requires one control card, and LISTCTLG is the action word. There are no other parameters.

   ```
   //LISTCAT JOB (accnt),NAME,MSGLEVEL=(1,1)
   //STEP1 EXEC PGM=IEHLIST
   //SYSPRINT DD SYSOUT=A
   //DRUM DD UNIT=2305,VOL=SER=DD2305,DISP=OLD
   //SYSIN DD *
      LISTCTLG
   /*
   //
   ```

OS Utilities: The Rules with Examples

5. Listing a secondary catalog using IEHLIST requires one control card, and LISTCTLG is the action word. The one parameter is CVOL, where you give the unit type and serial number of the disk pack or drum that the secondary catalog is on.

```
//SECCATLG JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
   LISTCTLG CVOL=3330=D10040
/*
//
```

IEHDASDR

Warning: If you have installed Jim Morrison's modifications to provide 3380/3390 support under MVS, be advised that IEHDASDR does not recognize these device types and will not function correctly, if at all, with volumes of these types.

6. Initializing a new disk pack using IEHDASDR requires no DD card, and the disk must be offline. ANALYZE is the action word. The parameters are: TODD, to specify the offline disk drive; VTOC, which tells where the VTOC is to begin; EXTENT, which gives the length of the VTOC; and NEWVOLID, which assigns a volume serial number.

```
//INITLIZE JOB (accnt),NAME,MSGLEVEL(1,1)
//STEP1 EXEC  PGM=IEHDASDR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
   ANALYZE TODD=230,VTOC=1,EXTENT=5,NEWVOLID=D10040
/*
//
```

7. Reinitializing a disk pack using IEHDASDR, the disk is left online, and one DD card with any DD name is needed for the whole pack. ANALYZE is the action word. The parameters VTOC, EXTENT, and NEWVOLID are the same as for the offline initialization, TODD names the DD card, and PURGE=YES allows it to erase information on the pack.

```
//ANALYZE JOB (accnt),NAME,MSGLEVEL(1,1)
//STEP1 EXEC  PGM=IEHDASDR
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
   ANALYZE TODD=DISK,VTOC=1,EXTENT=5,NEWVOLID=D10040,   C
                   PURGE=YES
/*
//
```

OS Utilities: The Rules with Examples

8. Giving an old pack a new volume serial number by using IEHDASDR is called relabeling a disk pack. It has one DD card, referring to the whole pack being relabeled. The action word on the control card is LABEL, and the two parameters are TODD, naming the DD card, and NEWVOLID, giving the new volume serial number.

```
//RELABEL JOB (accnt),NAME,MSGLEVEL(1,1)
//STEP1 EXEC PGM=IEHDASDR
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
   LABEL TODD=DISK,NEWVOLID=D10041
/*
//
```

9. Creating a backup tape using IEHDASDR requires two DD cards, one for the disk you're dumping from (specifying the whole pack), and the other for the tape you're dumping to (specifying a new tape). On the control card, DUMP is the action word. The parameters are: FROMDD, giving the name of the disk DD card; and TODD, giving the name of the tape DD card.

```
//DUMP JOB (accnt),NAME,MSGLEVEL(1,1)
//STEP1 EXEC PGM=IEHDASDR
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//TAPE DD DSN=TAPE,UNIT=3400-3,DISP=(NEW,KEEP)
//SYSIN DD *
   DUMP FROMDD=DISK,TODD=TAPE
/*
//
```

10. Printing specific tracks of a disk pack using IEHDASDR is part of the dump function, and is the same as dumping the pack to tape except that the DD card for the tape is left out. The TODD parameter refers to the SYSPRINT file (dumping to the printer), and the two new parameters are BEGIN and END, giving the cylinder and track numbers (eight hexadecimal numbers in all) with which to start and stop printing.

```
//PRINTRKS JOB (accnt),NAME,MSGLEVEL(1,1)
//STEP1 EXEC PGM=IEHDASDR
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
   DUMP FROMDD=DISK,TODD=SYSPRINT,BEGIN=00050002,   C
              END=00060002
/*
//
```

OS Utilities: The Rules with Examples

11. Restoring a disk from a backup tape using IEHDASDR requires two DD cards: one is for the tape you are restoring from, specifying an old tape (DSN, VOL, UNIT, DISP=OLD); the other is for the disk you are restoring to, specifying the whole disk. On the control card, RESTORE is the action word. The parameters are: FROMDD, naming the DD card for the tape; TODD, naming the DD card for the disk; and PURGE=YES, allowing it to erase any information still on the disk.

```
//RESTORE JOB (accnt),NAME,MSGLEVEL(1,1)
//STEP1 EXEC PGM=IEHDASDR
//SYSPRINT DD SYSOUT=A
//TAPE DD DSN=TAPE,VOL=SER=004000,UNIT=3400-3,
//                   DISP=OLD
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
   RESTORE FROMDD=TAPE,TODD=DISK,PURGE=YES
/*
//
```

IEBGENER

12. IEBGENER requires a special parameter on its SYSUT2 (output) DD card -- DCB=BLKSIZE= -- where you specify how big you want to block the information on the output file. The blocksize must be an even multiple of the size of a record. For the card punch, it is always DCB=BLKSIZE=80 (unblocked); for the printer, it is always equal to the size of one record (also not blocked).

13. IEBGENER copies information from an input file to an output file. The name of the DD card for the input file must be SYSUT1, and can be cards (//SYSIN DD *) or an old file on tape or disk (DSName, UNIT, VOL=SER, DISP). The name of the DD card for the output file must be SYSUT2, which can be the printer (SYSOUT=A), punched cards (SYSOUT=B), a new file on tape (DSName, UNIT, DISP), or a new file on disk (DSN, UNIT, VOL=SER, DISP, SPACE). The DCB=BLKSIZE parameter must also be on this card. No control card is needed, and its absence is indicated by a //SYSIN DD DUMMY card.

Copying tape to tape:

```
//DUPTAPE JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT-A
//SYSUT1 DD DSN=TAPE,UNIT=3400-3,VOL=SER=004000,
//                   DISP=OLD
//SYSUT2 DD DSN=TAPE,UNIT=3400-3,DISP=(NEW,KEEP),
//                   DCB=BLKSIZE=80
//SYSIN DD DUMMY
```

OS Utilities: The Rules with Examples

Tape to print:

```
//PRNTTAPE JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=TAPE,UNIT=3400-3,VOL=SER=004000,
//                       DISP=OLD
//SYSUT2 DD SYSOUT=A,DCB=BLKSIZE=80
//SYSIN DD DUMMY
```

Card to tape:

```
//CARDTAPE JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD *

    [card file to be put on tape goes here]

/*
//SYSUT2 DD DSN=TAPE,UNIT=3400-3,DISP=(NEW,KEEP),
//                     DCB=BLKSIZE=80
//SYSIN DD DUMMY
```

IEBCOPY

14. IEBCOPY always uses two temporary work files on disk, and a DD card is needed for each (UNIT and SPACE parameters needed only). The DD names must be SYSUT3 and SYSUT4; each usually needs about five tracks of space.

```
//SYSUT3 DD UNIT=3330,SPACE=(TRK,5)
//SYSUT4 DD UNIT=3330,SPACE=(TRK,5)
```

15. Copying a partitioned file from one disk to another using IEBCOPY requires four DD cards, two for work files SYSUT3 and SYSUT4 plus one for the old file you are copying from (DSName, UNIT, VOL=SER, DISP) and one for the new file you are copying to (DSName, UNIT, VOL=SER, DISP, SPACE), both with any DD names. There is one control card, and COPY is the action word. The two parameters are: INDD, giving the name of the DD card describing the old file; and OUTDD, giving the name of the DD card describing the new file. The control card key words (COPY, INDD, OUTDD) may all be abbreviated to their first letters.

OS Utilities: The Rules with Examples

```
//COPY JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//FROM DD DSN=FROMLIB,UNIT=3330,VOL=SER=D10040,
//                    DISP=OLD
//TO DD DSN=TOLIB,UNIT=3330,VOL=SER=D10080,
//            DISP=(NEW,KEEP),SPACE=(TRK,(400,,50))
//SYSUT3 DD UNIT=3330,SPACE=(TRK,5)
//SYSUT4 DD UNIT=3330,SPACE=(TRK,5)
//SYSIN DD *
   COPY INDD=FROM,OUTDD=TO
/*
//
```

16. Copying individual members from one partitioned file to another using IEBCOPY requires four DD cards: two for the SYSUT3 and SYSUT4 work files, plus one for the old file from which you are copying (DSN, UNIT, VOL=SER, DISP) and one for the old file to which you are copying (DSN, UNIT, VOL=SER, DISP), both with any DD names. There are two control cards. COPY is the action word of the first, with the parameters INDD and OUTDD: the action word for the second is either SELECT or EXCLUDE, with the parameter MEMBER giving the names of the members you either want included or excluded from the copy operation. The control card key words (COPY, INDD, OUTDD, SELECT, EXCLUDE, MEMBER) may all be abbreviated to their first letters.

```
//COPYMEM JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//FROM DD DSN=FROMLIB,UNIT=3330,VOL=SER=D10040,
//                    DISP=OLD
//TO DD DSN=TOLIB,UNIT=3330,VOL=SER=D10080,DISP=OLD
//SYSUT3 DD UNIT=3330,SPACE=(TRK,5)
//SYSUT4 DD UNIT=3330,SPACE=(TRK,5)
//SYSIN DD *
   COPY INDD=FROM,OUTDD=TO
   SELECT MEMBER=(PROGA,PROGB)
/*
//
```

If the SELECT control statement results in an attempt to copy a member from the input file, and a member with the same name is already present in the output file, the member will **not** be copied from the input file and the copy operation will continue with the next member from the input file. In order to replace a member in the output file with a duplicate named member from the input file, you must enclose the member name with an additional set of parentheses and add an additional parameter, R, which indicates REPLACE:

```
SELECT MEMBER=(PROGA,(PROGB,,R))
```

OS Utilities: The Rules with Examples

17. Merging the members from two, or more, partitioned files to an output partitioned file requires a slightly different control card:

    ```
    COPY INDD=(FROM1,FROM2),OUTDD=TO
    ```

    During a copy operation of this type, all the members contained in the file specified by the FROM1 DD card will be copied to the file specified by the TO DD card. The copy operation will continue by copying all the members contained in the file specified by the FROM2 DD card to the output file. If the copy operation encounters a member in an of the input file that has the same name as a member already present in the output file, the member from the input file will be skipped and the copy operation will continue with the next member from the same input file. The control card key words (COPY, INDD, and OUTDD may all be abbreviated to their first letters.

    In order to **replace** members in the output partitioned file with members from an input partitioned file when a duplicate name is encountered, you must enclose the DD name for the input file with an additional set of parentheses and add an additional parameter, R, which indicates REPLACE:

    ```
    COPY INDD=(FROM1,(FROM2,R)),OUTDD=TO
    ```

18. A member from an input partitioned file may be renamed when being copied to the output partitioned file by enclosing the member name with an additional set of parentheses and supplying the new name following the existing name:

    ```
    SELECT MEMBER=(PROGA,(PROGB,PROGX))
    ```

    The control card key words (SELECT and MEMBER) may be abbreviated to their first letters. If a member with the new name already exists in the output file, the member will **not** be copied unless the REPLACE parameter is also specified:

    ```
    SELECT MEMBER=(PROGA,(PROGB,PROGX,R))
    ```

19. Compressing a partitioned file (reclaiming space and directory entries from previously deleted members) using IEBCOPY requires one DD card -- besides those for the SYSUT3 and SYSUT4 work files -- for the file being compressed, which is an old file on disk (DSN, UNIT, VOL, DISP). The card can have any DD name, and there is one control card, COPY is the action word. The two parameters INDD and OUTDD both specify the same name of the DD card for the file being compressed. The control card key words (COPY, INDD, OUTDD) may all be abbreviated to their first letters.

    Warning: Specifying the same library as both input and output for the purposes of compressing the library can be hazardous; if the job terminates abnormally the library being compressed may be destroyed and the contents unrecoverable.

```
//COMPRESS JOB (accnt),NAME,MSGLEVEL=(1,1)
// STEP1 EXEC PGM=IEBCOPY,REGION=100K
//SYSPRINT DD SYSOUT=A
//LIB DD DSN=LIBFILE,UNIT=3330,VOL=SER=D10040
//              DISP=OLD
//SYSUT3 DD UNIT=3330,SPACE=(TRK,5)
//SYSUT4 DD UNIT=3330,SPACE=(TRK,5)
//SYSIN DD *
   COPY INDD=LIB,OUTDD=LIB
/*
//
```

20. When compressing a partitioned file using IEBCOPY, add the parameter REGION=100K to the EXEC card.

IEHMOVE

21. IEHMOVE needs a special DD card for disk work space that looks like this:

    ```
    //SYSUT1 DD UNIT=3330,SPACE=(TRK,30)
    ```

    More disk space is needed in the rare instances when the partitioned file you are working with has over 800 members. If you experience an SB37 abend as a result of all extents for SYSUT1 being exhausted, you should add a PARM='POWER=9' to the EXEC statement.

22. When using IEHMOVE to move a partitioned file from disk onto tape (unloading) or from the tape back onto the disk (restoring), two DD cards are needed, one for disk and one for tape. They can have any DD names. The one for the disk specifies a whole pack (UNIT, VOL, DISP). The DD card for the tape specifies a new file on tape (DSN, UNIT, DISP=(NEW,KEEP)) when unloading the file to tape, and specifies an old file on tape (DSN, UNIT, VOL, DISP=OLD) when restoring.

    Unloading:

    ```
    //DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
    //TAPE DD DSN=TAPE,UNIT=3400-3,DISP=(NEW,KEEP)
    ```

    Restoring:

    ```
    //DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
    //TAPE DD DSN=TAPE,UNIT=3400-3,VOL=SER=000040,DISP=OLD
    ```

OS Utilities: The Rules with Examples

23. lEHMOVE needs one control card, and COPY is the action word; PDS gives the name of the partitioned file. FROM and TO specify the unit type and volume serial numbers of the devices we are moving the file from and to (when unloading, say TO=3400-3=SCRTCH). If the dataset resides on a tape device, the serial number must be enclosed in parentheses, and the dataset sequence number must be included as: (serial,sequence number).

Unloading:

```
//UNLDPDS JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=3330,SPACE=(TRK,30)
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//TAPE DD DSN=TAPE,UNIT=3400-3,DISP=(NEW,KEEP)
//SYSIN DD *
   COPY PDS=PARTFILE,FROM=3330=D10040,         C
              TO=3400-3=(SCRTCH,1),TODD=TAPE
/*
//
```

Restoring:

```
//RSTRPDS JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=3330,SPACE=(TRK,30)
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//TAPE DD DSN=TAPE,UNIT=3400-3,VOL=SER=000040,
//                 DISP=OLD
//SYSIN DD *
   COPY PDS=PARTFILE,FROM=3400-3=(000040,1),    C
              TO=3330=D10040,FROMDD=TAPE
/*
//
```

IEHPROGM

24. IEHPROGM needs one DD card for the disk or drum you are working with, specifying the whole pack (UNIT, VOL, DISP).

```
//DRUM DD UNIT=2305,VOL=SER=DD2305,DISP=OLD
```

25. Cataloging a file using IEHPROGM requires one control card, and CATLG is the action word. The two parameters are DSNAME (may not be abbreviated DSN), giving the name of the file, and VOL, giving the unit type and volume serial number of the disk or tape the file is on.

OS Utilities: The Rules with Examples

```
//CATLG JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
  CATLG DSNAME=FILE,VOL=3330=D10040
/*
//
```

It is also permissible to indicate a specific catalog on the control statement by the inclusion of the CVOL parameter on the control statement:

```
CATLG DSNAME=FILE,VOL=3330=D10040,CVOL=3330=MVTRES
```

although this will be most applicable if you are running under MVT rather than MVS.

26. Uncataloging a file using IEHPROGM requires one control card, and UNCATLG is the action word. The one parameter is DSNAME (may not be abbreviated DSN), giving the name of the file to be uncataloged.

```
//UNCATLG JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
  UNCATLG DSNAME=FILE
/*
//
```

It is also permissible to indicate a specific catalog on the control statement by the inclusion of the CVOL parameter on the control statement:

```
UNCATLG DSNAME=FILE,VOL=3330=D10040,CVOL=3330=MVTRES
```

although this will be most applicable if you are running under MVT rather than MVS.

27. Renaming a file using IEHPROGM requires one control card, and RENAME is the action word. The parameters are: DSNAME (may not be abbreviated DSN), giving the file's current name; VOL, giving the unit type and volume serial number of the pack the file is on; and NEWNAME, specifying the new name you want to give to the file.

```
//RENAME JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
  RENAME DSNAME=LIB,VOL=3330=D10040,NEWNAME=LIBA
/*
//
```

OS Utilities: The Rules with Examples

28. Renaming a member of a partitioned file using IEHPROGM requires one control card, and RENAME is the action word. The parameters are: DSNAME (may not be abbreviated DSN), giving the name of the whole partitioned file; VOL, with the unit type and volume serial number of the pack that the file is on; MEMBER, giving the name of the member to be renamed; and NEWNAME specifies what you want the member's new name to be.

```
//RENAMEM JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
  RENAME DSNAME=LIB,VOL=3330=D10040,MEMBER=MEM,   C
                    NEWNAME=MEMA
/*
//
```

29. Scratching (deleting) a file using IEHPROGM requires one control card, and SCRATCH is the action word. The parameters are: DSNAME (may not be abbreviated DSN), naming the file you want scratched; and VOL, giving the unit type and number of the pack it's on; the parameter PURGE can be added if the file is to be scratched before its expiration date.

```
//SCRFILE JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
  SCRATCH DSNAME=SOMEFILE,VOL=3330=D10040
/*
//
```

30. Scratching a member of a partitioned file using IEHPROGM requires one control card, and SCRATCH is the action word. The parameters are: DSNAME (may not be abbreviated DSN), giving the name of the partitioned file; VOL, telling which unit type and pack the file is on; and MEMBER, naming the member to be scratched.

```
//SCRMEM JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
  SCRATCH DSNAME=SOMEFILE,VOL=3330=D10040,       C
                    MEMBER=SOMEMEM
/*
//
```

31. Scratching all the files on a disk pack using IEHPROGM requires one control card, and SCRATCH is the action word. The parameters are: VTOC, with no equal sign; and VOL, giving the unit type and number of the pack to be cleaned out; the parameter PURGE can be added to scratch files that haven't yet expired.

OS Utilities: The Rules with Examples

```
//SCRVTOC JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DISK DD UNIT=3330,VOL=SER=D10040,DISP=OLD
//SYSIN DD *
  SCRATCH VTOC,VOL=3330=D10040,PURGE
/*
//
```

IEBPTPCH

32. IEBPTPCH needs two DD cards. The first, which must have the name SYSUT1, describes the file you are printing from (often a cataloged, partitioned file on disk, needing DSN and DISP). The second DD card, which must have the name SYSUT2, describes the printer (SYSOUT=A).

```
//SYSUT1 DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSUT2 DD SYSOUT=A
```

33. Printing out individual members of a partitioned card image file using IEBPTPCH requires three types of control cards. On the first card, PRINT is the action word. Its three parameters are: TYPORG=PO, specifying the partitioned organization of the file; MAXNAME, giving the number of MEMBER cards that follow; and MAXFLDS, giving the number of RECORD cards that follow. After that comes a pair of cards for every member to be printed. MEMBER is the action word of the first card of each pair; its parameter, NAME, names the member to be printed; RECORD is the action word on the second card of each pair, and its parameter, FIELD, gives the length of the record (usually FIELD=(80)).

```
//PRINTMEM JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBPTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSUT2 DD SYSOUT=A
//SYSIN DD *
  PRINT TYPORG=PO,MAXNAME=2,MAXFIELDS=2
  MEMBER NAME=WTR
  RECORD FIELD=(80)
/*
//
```

34. Printing all the members of a partitioned card image file using IEBPTPCH requires two control cards. On the first card, PRINT is the action word. Its parameters are: TYPORG=PO, specifying partitioned organization; and MAXFLDS=1, saying that one RECORD card follows. RECORD is the action word of the second card, and its parameter is FIELD=(80), specifying the length of each record.

OS Utilities: The Rules with Examples

```
//PRINTLIB JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBPTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSUT2 DD SYSOUT=A
//SYSIN DD *
  PRINT TYPORG=PO,MAXFIELDS=1
  RECORD FIELD=(80)
/*
//
```

35. Punching out members of a partitioned card image file using IEBPTPCH involves the same DD cards and control cards used for printing the members, except that the SYSUT2 (output) DD card must refer to the card punch (SYSOUT=B), and the action word on the first control card is PUNCH.

36. To duplicate a card deck and put new sequence numbers in columns 73-80 of the new card deck using IEBPTPCH requires two DD cards. SYSUT1 is the name of the first, which refers to the old card deck (SYSUT1 DD *) and is followed by the deck itself; the name of the second DD card is SYSUT2, which refers to the card punch (SYSOUT=B). There are two control cards. PUNCH is the action word of the first, and its three parameters are: MAXFLDS=1, CDSEQ, to assign a beginning sequence number; and CDINCR, to assign an increment for following sequence numbers. The second control card is RECORD FIELD=(72), meaning you just want to duplicate the first 72 columns of the old card deck.

```
//SEQCARDS JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBPTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD *

    [card deck to be resequenced goes here]

/*
//SYSUT2 DD SYSOUT=B
//SYSIN DD *
   PUNCH MAXFIELDS=1,CDSEQ=0,CDINCR=100
   RECORD FIELD=(72)
/*
//
```

IEBUPDTE

37. Modifying a partitioned card image file using IEBUPDTE requires two DD cards, the first with the name SYSUT1, and the second with the name SYSUT2, they both refer to the same file, the one being updated, which is an old file on disk.

```
//SYSUT1 DD DSN=SYS1.PARMLIB,DISP=OLD
//SYSUT2 DD DSN=SYS1.PARMLIB,DISP=OLD
```

If UPDATE=INPLACE is specified, the SYSUT2 DD card is optional and may be omitted.

38. Adding a new member to a partitioned card image file using IEBUPDTE requires three control cards, each with a ./ in columns 1 and 2. ADD is the action word on the first. Its two parameters are: NAME, giving a name to the new member; and LIST=ALL saying you want the member printed out. NUMBER is the action word on the second control card. Its parameters are: NEW1, specifying the number the first card image is to have in columns 73-80; and INCR, saying how much larger than the one before it you want the number in each card image to be. ENDUP is the action word of the last card; it has no parameters. Between the NUMBER and ENDUP cards are put the actual cards that are to become the new member.

```
//ADDMEM JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBUPDTE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYS1.PARMLIB,DISP=OLD
//SYSUT2 DD DSN=SYS1.PARMLIB,DISP=OLD
//SYSIN DD DATA
./ ADD NAME=SAMPLE,LIST=ALL
./ NUMBER NEW1=10,INCR=10
//STEP1 EXEC PGM=EXAMPLE
//SYSPRINT DD SYSOUT=A
./ ENDUP
/*
//
```

39. When using IEBUPDTE to modify a card image file containing JCL statements, the SYSIN DD card should look like this:

```
//SYSIN DD DATA
```

-- telling the reader/interpreter not to treat any JCL cards that follow as JCL cards until after the next /* card.

40. Replacing card images of a member in a partitioned card image file using IEBUPDTE requires two control cards, both with a ./ in columns 1 and 2. CHANGE is the action word on the first. Its two parameters are: NAME, giving the name of the member to be updated; and UPDATE=INPLACE, saying that the member can remain in the room it now occupies. ENDUP is the action word on the second control card, which has no parameters. In between them are put the replacement cards, each having in columns 73-80 the number that matches the card image it is to replace.

OS Utilities: The Rules with Examples

```
//UPDATE JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBUPDTE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYS1.PARMLIB,DISP=OLD
//SYSUT2 DD DSN=SYS1.PARMLIB,DISP=OLD
//SYSIN DD DATA
./ CHANGE NAME=SAMPLE,UPDATE=INPLACE
//STEP2 EXEC PGM=EXAMPLE2              00000010
./ ENDUP
/*
//
```

Note that when UPDATE=INPLACE is specified, only a single update operation may be specified for that execution of IEBUPDTE..

41. Adding new card images to a member of a partitioned card image file using IEBUPDTE requires two control cards, both with a ./ in columns 1 and 2. CHANGE is the action word of the first. Its one parameter is NAME, giving the name of the member to be changed. ENDUP is the action word of the second control card, and it has no parameters. ln between them are put the cards to be added, each having in columns 73-80 a number that indicates in between which card images it should be inserted.

```
//UPDATE2 JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBUPDTE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYS1.PARMLIB,DISP=OLD
//SYSUT2 DD DSN=SYS1.PARMLIB,DISP=OLD
//SYSIN DD DATA
./ CHANGE NAME=SAMPLE
//EXTRADD DD SYSOUT=A                  00000015
./ ENDUP
/*
//
```

42. Deleting card images from a member of a partitioned card image file using IEBUPDTE requires two cards, both with a ./ in columns 1 and 2. The action word of the first is CHANGE, and its one parameter is NAME giving the name of the member to be changed; DELETE is the action word of the second control card, where SEQ1 gives the sequence number of the first card image to be deleted and SEQ2 gives the sequence number of the last card image to be deleted; the card images with those two sequence numbers and any in between are deleted. If only one card image is to be deleted, both SEQ1 and SEQ2 give the sequence number of the same card image, the one to be deleted.

```
//DELETE JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBUPDTE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYS1.PARMLIB,DISP=OLD
//SYSUT2 DD DSN=SYS1.PARMLIB,DISP=OLD
//SYSIN DD *
./ CHANGE NAME=SAMPLE
./ DELETE SEQ1=15,SEQ2=20
/*
//
```

43. When creating a new partitioned card image file using IEBUPDTE, the PARM field on the EXEC statement specifies NEW; the SYSUT1 DD card is left out; the SYSUT2 DD card is for a new file on disk (DSN, UNIT, DISP=(NEW,KEEP), SPACE (reserving space for the directory too), and usually VOL=SER to put it on a specific pack). The control cards are the same as for adding to an existing partitioned card image file, with the ADD and NUMBER control cards being repeated for and followed by each new member. The ADD control card gives each member a different name, and there is only one ENDUP card, which goes at the very end.

```
//NEWPDS JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSN=NEWLIB,UNIT=3330,VOL=SER=D10040,
//                     SPACE=(TRK,(50,,34)),DISP=(NEW,KEEP)
//SYSIN DD DATA
./ ADD NAME=SAMPLE,LIST=ALL
./ NUMBER NEW1=10,INCR=10
//STEP1 EXEC PGM=EXAMPLE
//SYSPRINT DD SYSOUT=A
./ ADD NAME=SAMPLE2,LIST=ALL
./ NUMBER NEW1=10,INCR=10
//STEP2 EXEC PGM=EXAMPLE2
//SYSPRINT DD SYSOUT=A
./ ENDUP
/*
//
```

IEBCOMPR

44. The IEBCOMPR utility provides the ability to verify that the contents of two datasets are identical. The datasets to be compared may be physical sequential (PS) or partitioned (PO) organization. Four DD cards are required: the datasets to be compared are specified with the SYSUT1 and SYSUT2 DD cards, the output produced by the utility is written to the SYSPRINT DD card, and the utility control statements are read from the SYSIN DD card.

OS Utilities: The Rules with Examples

```
//COMPARE JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=DATASET1,UNIT=3330,VOL=SER=D10040,
//                       DISP=SHR
//SYSUT2 DD DSN=DATASET2,UNIT=3330,VOL=SER=D10040,
//                       ,DISP=SHR
//SYSIN DD *
  COMPARE TYPORG=PS
/*
//
```

45. IEBCOMPR needs one control card, and TYPORG is the action word; PS specifies that the datasets to be compared are physical sequential organization and PO specifies that the datasets are partitioned organization. Both datasets must be of identical organization.

    Since the default for IEBCOMPR is TYPORG=PS, the utility may be executed with no control statements (utilizing a DUMMY SYSIN statement) if the datasets to be compared are of the physical sequential organization. There are two conditions under which the COMPARE control statement is required: if the datasets to be compared are partitioned organization; **or** if you need to use any other control statement, it must be preceded by the COMPARE control statement, even if the dataset organization is physical sequential.

    When comparing libraries it is not necessary that the list of members in both libraries be identical in order to receive an equal comparison. If the SYSUT2 library contains all of the members that are in the SYSUT1 library plus some additional members and if the members that are in both libraries are identical, IEBCOMPR will indicate an equal condition.

    If the comparison of the datasets is successful (all data compared are equal), a return code of 0 is set. If the comparison fails, a return code of 8 is set. 12 indicates that an unrecoverable error occurred, and 16 indicates that a routine specified in the EXITS statement failed.

46. The LABEL control statement specifies how IEBCOMPR is to handle user labels, and DATA is the action word. Since only tape datasets have user labels, this control statement is only utilized when the datasets to be compared reside on tape. If the LABEL statement is coded as:

```
LABEL DATA=YES
```

(which is the default) the user labels are treated as data after they have been processed by the user's label processing routine. If NO is coded, the user labels are not treated as data and are not processed by IEBCOMPR. ONLY may be coded in place of YES, which results in comparison of only the user labels and not the data. The final alternative is to code ALL, which is used when there is more than one user label (there may be up to 16 user labels following the standard tape label). ALL specifies that even if an error is found in a user label, the remaining user labels are still to be compared.

47. The EXITS control statement is used to supply the names of user exit routines to be utilized. The four possible exits are INHDR, INTLR, ERROR, and PRECOMP, illustrated by the following example:

```
EXITS INHDR=UHRTN,INTLR=UTRTN,ERROR=UERTN,PRECOMP=PCRTN
```

If the EXITS control statement is supplied, it is only necessary to code the routines you wish to utilize; it is not necessary to code all four routines. INHDR and INTLR supply the name of the routine to process the user header and user trailer labels, respectively. ERROR supplies the name of the routine that receives control when an unequal comparison is encountered. If no user ERROR routine is supplied, after ten unequal comparisons the comparison ends, except when members of a library are being compared, in which case IEBCOMPR proceeds to the next member of the partitioned dataset. PRECOMP supplies the name of a routine that is to process the data to be compared before they are made available to IEBCOMPR.

IEHINITT

48. The IEHINITT utility provides the ability to initialize empty tape volumes for use. The program writes a set of standard labels onto the tape, which includes a VOL1 label containing a six character serial number and an optional owner identification string and a dummy HDR1 label. In the Hercules' environment, the HETINIT utility, included with the Hercules installation, may be utilized to provide the identical functionality of IEHINITT under the host operating system. Three DD cards are required: the tape volume(s) to be initialized may be specified with any valid DD name (which must match the name on a corresponding Utility Control Statement), the output produced by the utility is written to the SYSPRINT DD card, and the utility control statements are read from the SYSIN DD card.

OS Utilities: The Rules with Examples

```
//INITTAPE JOB (accnt),NAME,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IEHINITT
//SYSPRINT DD SYSOUT=A
//TAPE1    DD UNIT=(TAPE,,DEFER),DCB=DEN=3
//SYSIN DD *
TAPE1    INITT
SER=000ABC,NUMBTAPE=1,OWNER='HERCULES/MVS',DISP=UNLOAD
/*
//
```

49. IEHINITT requires one control card, and INITT is the action word. The parameters are: SER (serial number), which specifies the one to six character identification number to be written in the VOL1 label; NUMBTAPE, which indicates the number of tape to initialized; OWNER, which specifies the optional one to ten characters of owner identification information; and DISP, which specifies the action to take after initialization.

SER - The serial number specified may not contain blanks, commas, apostrophes, equal signs, or special characters other than periods, hyphens, dollar signs, pound signs (#), and at signs (@). If NUMBTAPE is specified and is greater than 1, the serial number specified is incremented by 1 for each subsequent tape; 999999 will be incremented to 000000.

NUMBTAPE - If specified, indicates the number of tapes that are to be initialized as a result of the information provided with this control statement; the default is 1 and the maximum that may be specified is 255. If a number other than 1 is specified, the serial number (SER) specified must be numeric in order to accommodate automatic increment for second and subsequent tapes.

OWNER - If specified, provides up to ten characters of information to be written to the owner identification field of the VOL1 label. The information must be enclosed in delimiting apostrophes if blanks, commas, apostrophes, equal signs or other special characters (except periods or hyphens) are included.

DISP - If specified, designates action to be taken after the labels are written to the tape volume; UNLOAD (the default) and REWIND are the acceptable options.