LIBMAN USER'S GUIDE

February 25th, 1985

Nand Lal, DESIGN/PROGRAM
Teresa B. Sheets, DOCUMENTATION

NASA/Goddard Space Flight Center
Greenbelt, Maryland

ABSTRACT

The LIBMAN process has been set up to archive and restore catalogued data sets on the IBM 3081. JCL PROCEDURES have been set up to handle the LIBMAN functions. They are: INIT, ADD, ARCHIVE and RESTORE.

CONTENTS

1.0 INIT

1.1 OVERVIEW

INIT initializes a specified library, where libraries are defined by type. INIT is run only once per library.

1.2 INPUT/OUTPUT

The input to the INIT procedure consists of a standard labelled tape (SL) and 3 symbolic parameters: the userid, the type and the volume serial number of the input tape.

PARAMETER	DEFAULT	DESCRIPTION
USERID		A 5 character project id
TYPE		One character A-Z that arbitrarily specifies the type of data stored on the tape.
VOL	SCRTCH	Volume serial number of the input tape

The output generated by this program is a catalog of the specified library type on the specified tape.

1.3 RESTRICTIONS

The input tape must be labelled SL, and in the proper tape slot.

1.4 STEPS FOR EXECUTING INIT

1. Label a new tape SL
2. Set up JCL to execute 'SB#HP.LIBMAN.CNTL(INIT)' procedure. The best way to generalize this is by setting up a CLIST to request the input, build the JCL, and submit the job.
3. The JCL format to execute the INIT procedure is:

```
//STEP1 EXEC INIT,USERID=xxxxx,TYPE=y,VOL=zzzzzz
```

where xxxxx is the project id, y is the type, and zzzzzz is the volume serial number of the input tape.

1.5 ERROR HANDLING

If there is an error, then the job will not be executed. If the job abends with user 39, then the requested library type already exists. Re-submit the job with a different type. If the job abends with user 139, the volume serial number was not specified. Re-submit the job with a valid volume serial number.

1.6 SAMPLE JCL

```
//USERIDRUN JOB (SBOXX,BOX,EX),LIBMAN,TIME=(0,30),
//      MSGCLASS=X,NOTIFY=USERID
//INIT   PROC  USERID=,       SB#HP.LIBMAN.CNTL(INIT)11/02/82
//             TYPE=,
//             VOL=SCRTCH
//*****************************************************************
//*  INIT PROC is used to initialize library defined by type TYPE.    *
//*                                                                   *
//*  All symbolic parameters must be assigned values.                 *
//*                                                                   *
//*  If the library specified already exists on this machine, program *
//*  will ABEND with user 39.                                         *
//*  If volume serial number is not specified, or SCRTCH is specified *
//*  as the volume serial number, user ABEND 139 will result.         *
//*  If the INIT step ABENDs with any other code, send a message to   *
//*  se2NL.                                                           *
//*****************************************************************
//LIBINIT  EXEC  PGM=LIBMAN,PARM='INIT,&TYPE.'
//STEPLIB  DD DSN=SB#HP.LIBMAN.LOAD,DISP=SHR
//LIBVOLS  DD DSN=&USERID..&TYPE.LIBVOLS.V0000000,UNIT=(6250,,DEFER),
//            DISP=(NEW,PASS),VOL=(,RETAIN,SER=&VOL.)
//SNAP     DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//         PEND
// EXEC INIT,USERID=SB#EG,TYPE=P,VOL=EGBK01
// EXEC NTSO
```

2.0 ARCHIVE

2.1 OVERVIEW

The ARCHIVE procedure copies the specified disk data set onto a tape in the specified library type. The tape data set name, which includes the library type, is catalogued.

2.2 INPUT/OUTPUT

The input to the ARCHIVE procedure consists of the userid, the library type, the data set name, the data set name qualifier, and the output class.

PARAMETER	DEFAULT	DESCRIPTION
USERID		Five character project userid
TYPE	A	One character to specify library type
PROG		Data set name
QUAL		Data set qualifier
OUT	A	Output class for messages

The output generated by this procedure is a catalogued copy of the specified data set in the specified library type, on the LIBMAN tape.

2.3 RESTRICTIONS

NOTE: There are only two symbolic parameters to enter the data set name and qualifier. If the data set name has more that one qualifier, then the name must be entered in quotes because is contains the special character '.'. The names can be entered in PROG or QUAL. For example, a standard data set name would be entered (without quotes) ...PROG=LIB,QUAL=CNTL... A data set name with more qualifiers would be entered ... PROG='X.Y',QUAL='Z.CNTL'...

2.4 STEPS FOR EXECUTING

Set up JCL to execute 'SB#HP.LIBMAN.CNTL(ARCHIVE)' procedure. The best way to generalize this is by setting up a CLIST to request the input, build the JCL, and submit the job.

The JCL format for executing the ARVHIVE procedure is:

```
//STEP1 EXEC ARCHIVE,USERID=xxxxx,TYPE=y,PROG=zzzzzz,
//       QUAL=aaaa,OUT=b
```

where xxxxx is the project id, y is the type, zzzzzz is the data set name, aaaa is the data set qualifier, and b is the system output class.

2.5 SAMPLE JCL

```
//USERIDRUN JOB (SBOXX,BOX,EX),LIBMAN,TIME=(0,30),
//      MSGCLASS=X,NOTIFY=USERID
//ARCHIVE  PROC  USERID=,   Five-character USERID of library
//         TYPE=A,       One-character identifier for library
//         PROG=,        Name of the program
//         QUAL=,        Dataset qualifier
//         OUT=A         System output class for messages
//*****************************************************************
//ARCHIVE  EXEC  PGM=LIBMAN,PARM='SAVE,&TYPE.N',REGION=200K
//STEPLIB  DD  DSN=SB#HP.LIBMAN.LOAD,DISP=SHR
//*****************************************************************
//LIBVOLS  DD  DSN=&USERID..&TYPE.LIBVOLS.V0000000,DISP=(OLD,PASS),
//  UNIT=(,,DEFER),VOL=(,RETAIN),LABEL=(,,,OUT)
//UNLIN    DD  DUMMY
//UNLPRINT DD  SYSOUT=&OUT,DCB=BLKSIZE=3509
//UNLWORK1 DD  DSN=&USERID..&PROG..&QUAL,DISP=SHR
//UNLWORK2 DD  UNIT=AFF=LIBVOLS,DISP=(OLD,PASS),
//  VOL=(,RETAIN,SER=DUMUNL)
//UNLWORK3 DD  UNIT=SYSDA,SPACE=(TRK,(1,10))
//UNLWORK4 DD  UNIT=SYSDA,SPACE=(TRK,(1,10))
//SNAP     DD SYSOUT=A
//SYSUDUMP DD  SYSOUT=A
//COPY     EXEC  PGM=IEBCOPY,REGION=200K,COND=(573,NE,ARCHIVE)
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DSN=*.ARCHIVE.UNLWORK1,DISP=(OLD,KEEP)
//SYSUT2   DD  DSN=*.ARCHIVE.UNLWORK2,DISP=OLD,VOL=(,RETAIN)
//SYSUT3   DD  UNIT=SYSDA,SPACE=(TRK,(1,10))
//SYSUT4   DD  UNIT=SYSDA,SPACE=(TRK,(1,10))
//SYSIN    DD  DUMMY
//CATALOG  EXEC  PGM=LIBMANC,PARM='SAVE,&TYPE.N',REGION=200K,
//   COND=((573,NE,ARCHIVE),(0,NE,COPY))
//STEPLIB  DD  DSN=SB#HP.LIBMAN.LOAD,DISP=SHR
//*****************************************************************
//LIBVOLS  DD  DSN=&USERID..&TYPE.LIBVOLS.V0000000,DISP=(OLD,PASS),
//  UNIT=(,,DEFER),VOL=(,RETAIN),LABEL=(,,,OUT)
//UNLWORK1 DD  DSN=&USERID..&PROG..&QUAL,DISP=SHR
//SNAP     DD SYSOUT=A
//SYSUDUMP DD  SYSOUT=A
//ARCHIVE  PEND
// EXEC ARCHIVE,USERID=SB#EG,TYPE=P,PROG=LIB,QUAL=CNTL,OUT=A
// EXEC NTSO
```

3.0 RESTORE

3.1 OVERVIEW

The RESTORE procedure finds and copies the requested data set from the catalogued library tape to a catalogued disk data set.

3.2 INPUT/OUTPUT

The input to the RESTORE procedure consists of the userid, the library type, the data set name, the data set qualifier, and system output class.

PARAMETER	DEFAULT	DESCRIPTION
USERID		Five character project userid
TYPE	A	One character to specify library type
PROG		Data set name
QUAL		Data set qualifier
OUT	A	Output class for messages

The output generated by this procedure is a the specified disk data set restored from the specified library type tape.

3.3 RESTRICTIONS

NOTE: There are only two symbolic parameters to enter the data set name and qualifier. If the data set name has more that one qualifier, then the name must be entered in quotes because is contains the special character '.'. The names can be entered in PROG or QUAL. For example, a standard data set name would be entered (without quotes) ...PROG=LIB,QUAL=CNTL... A data set name with more qualifiers would be entered ... PROG='X.Y',QUAL='Z.CNTL'... The output data set must be allocated and catalogued before the restore procedure can be executed.

3.4 STEPS FOR EXECUTING

Allocate and catalog the output data set. Set up JCL to execute 'SB#HP.LIBMAN.CNTL(RESTORE)' procedure. The best way to generalize this is by setting up a CLIST to request the input, build the JCL, and submit the job.

The JCL format for executing the RESTORE procedure is:

```
//STEP1 EXEC RESTORE,USERID=xxxxx,TYPE=y,VOL=zzzzzz,
//       QUAL=aaaa,OUT=b
```

where xxxxx is the project id, y is the type, zzzzzz is the data set name, aaaa is the data set qualifier, and b is the system output class.

3.5 SAMPLE JCL

```
//USERIDRUN JOB (SBOXX,BOX,EX),LIBMAN,TIME=(0,30),
//     MSGCLASS=X,NOTIFY=USERID
//RESTORE  PROC  USERID=,   Five-character USERID of library
//         TYPE=A,      One-character identifier for library
//         PROG=,       Name of the program
//         QUAL=,       Dataset qualifier
//         OUT=A        System output class for messages
//*****************************************************************
//RESTORE  EXEC  PGM=LIBMAN,PARM='REST,&TYPE.N',REGION=200K
//STEPLIB  DD  DSN=SB#HP.LIBMAN.LOAD,DISP=SHR
//*****************************************************************
//LIBVOLS  DD  DSN=&USERID..&TYPE.LIBVOLS.V0000000,DISP=(OLD,PASS),
//  UNIT=(,,DEFER),VOL=(,RETAIN)
//LODIN    DD  DUMMY
//LODPRINT DD  SYSOUT=&OUT,DCB=BLKSIZE=3509
//LODWORK2 DD  DSN=&USERID..&PROG..&QUAL,DISP=SHR
//LODWORK1 DD  UNIT=AFF=LIBVOLS,DISP=(OLD,PASS),
//             VOL=(,RETAIN,SER=DUMLOD)
//SYSUDUMP DD  SYSOUT=A
//SNAP     DD  SYSOUT=A
//COPY     EXEC  PGM=IEBCOPY,REGION=200K,COND=(273,NE,RESTORE)
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DSN=*.RESTORE.LODWORK1,DISP=(OLD,KEEP)
//SYSUT2   DD  DSN=*.RESTORE.LODWORK2,DISP=OLD
//SYSUT3   DD  UNIT=SYSDA,SPACE=(TRK,(1,10))
//SYSUT4   DD  UNIT=SYSDA,SPACE=(TRK,(1,10))
//SYSIN    DD  DUMMY
//RESTORE  PEND
// EXEC RESTORE,USERID=SB#EG,TYPE=P,PROG=LIB,QUAL=CNTL,OUT=A
// EXEC NTSO
```

4.0 ADD

4.1 OVERVIEW

ADD adds a new tape to the specified library type. ADD is executed when the library type needs to be extended onto another output library tape.

4.2 INPUT/OUTPUT

The input to the ADD procedure consists of a tape labelled SL and 3 symbolic parameters: the userid, the type and the volume serial number of the input tape.

PARAMETER	DEFAULT	DESCRIPTION
USERID		A 5 character project id
TYPE		One character A-Z that arbitrarily specifies the type of data stored on the tape.
VOL	SCRTCH	Volume serial number of the input tape

The output generated by this program is the addition of the specified tape to the requested library type.

4.3 RESTRICTIONS

The input tape must be labelled SL, and in the proper tape slot.

4.4 STEPS FOR EXECUTING INIT

1. Label a new tape SL

2. Set up JCL to execute 'SB#HP.LIBMAN.CNTL(ADD)' procedure. The best way to generalize this is by setting up a CLIST to request the input, build the JCL, and submit the job.

3. The JCL format for executing the ADD procedure is:

```
//STEP1 EXEC ADD,USERID=xxxxx,TYPE=y,PROG=zzzzzz
```

where xxxxx is the project id, y is the type, and zzzzzz is the volume serial number of the input tape.

4.5 ERROR HANDLING

If there is an error, then the job will not be executed. If the job abends with user 39, then the requested library type already exists. Re-submit the job with a different type. If the job abends with user 139, the volume serial number was not specified. Re-submit the job with a valid volume serial number.

4.6 SAMPLE JCL

```
//USERIDRUN JOB (SBOXX,BOX,EX),LIBMAN,TIME=(0,30),
//      MSGCLASS=X,NOTIFY=USERID
//ADD     PROC  USERID=,        SB#HP.LIBMAN.PROC(ADD)11/02/82
//              TYPE=,
//              VOL=SCRTCH
//*****************************************************************
//LIBADD    EXEC  PGM=LIBMAN,PARM='ADD,&TYPE.,&VOL.',REGION=200K
//STEPLIB   DD DSN=SB#HP.LIBMAN.LOAD,DISP=SHR
//LIBVOLS   DD DSN=&USERID..&TYPE.LIBVOLS.V0000000,UNIT=(,,DEFER),
//             DISP=(OLD,PASS),VOL=(,RETAIN)
//SNAP      DD SYSOUT=A
//SYSUDUMP  DD SYSOUT=A
//          PEND
// EXEC ADD,USERID=SB#EG,TYPE=P,VOL=EGBK02
// EXEC NTSO
```