Transfer modules to other z/OS with sftp

This utility set has been created for cases that an installation decided to exchanges USS and/or MVS executables from one z/OS system to another one.

Two jobs have been created for doing this. One job in the source system is for unloading MVS libraries, MVS modules, USS structures or USS modules. The second job is for restoring this data on the target system after transferring the data with sftp. To do so, three REXX procedures have been created/used. Following some explanation about these routines.

paxunload

You should place this file to USS, for example "/usr/local/bin". This routine is needed on the source system to unload a USS structure, starting from a directory down within a specific file system, or a single USS module.

xmit2uss

You should transfer this file to a TSO REXX PDS(E) library. It is needed in the source system to unload an MVS library or single module to USS.

recvfuss

You should transfer this file to a TSO REXX PDS(E) library. It is needed in the target system to restore an MVS library or single module from USS.

Next two sample jobs are provided, that show how to use these function for transferring the MVS and USS modules.

Next you see a sample unload job for the source system combining all in one (MVS library, MVS module, USS structure and USS module)

```
//HERIJOB  JOB ,'XMIT Job',NOTIFY=&SYSUID.,REGION=0M,SYSTEM=SC70
//TSOBATCH EXEC PGM=IKJEFT1A
//SYSEXEC  DD DSNAME=HERING.REXX.EXEC,DISP=SHR
//SYSTSIN  DD DATA,DLM=##
xmit2uss 'hering.tso.load' /tmp/tso.load.xmit
xmit2uss 'hering.tso.load(liblist)' /tmp/tso.load.liblist.xmit
bpxbatsl pgm /bin/sh -c $STDIN|sh
##
//SYSTSPRT DD SYSOUT=*,LRECL=136,RECFM=VB
//STDIN    DD DATA,DLM=##
paxunload /usr/local/bin /tmp/hering.ubin.pax.Z &&
paxunload /bin/echo /tmp/hering.binecho.pax.Z &&
cd /tmp &&
sftp -oBatchMode=no -b - hering@wtsc65oe << _EOF_
lpwd
cd /tmp
put tso.load.xmit tso.load.xmit
put tso.load.liblist.xmit
put hering.ubin.pax.Z
put hering.binecho.pax.Z
quit
_EOF_
RC=$?
rm /tmp/tso.load.xmit /tmp/tso.load.liblist.xmit
rm hering.ubin.pax.Z hering.binecho.pax.Z
echo sftp processing is done...
if test $RC -ne 0; then; echo "\nRC= $RC"; exit 1; else; exit 0; fi
##
//STDENV   DD DATA,DLM=##
STDIN=/bin/cat //dd:STDIN
_BPX_BATCH_UMASK=0022
_BPX_SHAREAS=YES
PATH=/usr/local/bin:/bin
##
//STDOUT   DD SYSOUT=*,LRECL=136,RECFM=VB
```

Notes...

- The job is provided according to ccsids 1047 and 037.
- We currently run without superuser mode here, assuming that all data sets and USS structures can get accessed and can be read.
- Also all temporary files should get created and written to by the user of the job.
- If we need superuser mode (I currently do not expect that!), we need to make changes best in the USS REXX "paxunload".
- The sample job combines all cases via "xmit2uss" and "paxunload".

Next you see a sample restore job for the target system combining all in one (MVS library, MVS module, USS structure and USS module)

```
//HERIJOB  JOB ,'RECV Job',NOTIFY=&SYSUID.,REGION=0M,SYSTEM=SC65
//TSOBATCH EXEC PGM=IKJEFT1A
//SYSEXEC  DD DSNAME=HERING.REXX.EXEC,DISP=SHR
//SYSTSIN  DD DATA,DLM=##
recvfuss 'hering.tso.load.copy' /tmp/tso.load.xmit
recvfuss 'hering.tso.load.copy' /tmp/tso.load.liblist.xmit
bpxbatsl pgm /bin/sh -c $STDIN|sh
##
//SYSTSPRT DD SYSOUT=*,LRECL=136,RECFM=VB
//STDIN    DD DATA,DLM=##
if test $(id -u) -ne 0; then; su; fi
cd /tmp/hering &&
pax -pe -rf /tmp/hering.ubin.pax.Z &&
pax -pe -rf /tmp/hering.binecho.pax.Z &&
rm /tmp/tso.load.xmit /tmp/tso.load.liblist.xmit &&
rm /tmp/hering.ubin.pax.Z /tmp/hering.binecho.pax.Z
RC=$?
echo restore processing ended...
if test $RC -ne 0; then; echo "\nRC= $RC"; exit 1; else; exit 0; fi
##
//STDENV   DD DATA,DLM=##
STDIN=/bin/cat //dd:STDIN
_BPX_BATCH_UMASK=0022
_BPX_SHAREAS=YES
#PATH=/usr/local/bin:/bin
##
//STDOUT   DD SYSOUT=*,LRECL=136,RECFM=VB
```

Notes...

- The job is provided again according to ccsids 1047 and 037.
- Here we run in authorized or just superuser mode in USS to be sure being able to restore the USS data as needed.
- For MVS data we do not need this.