Sterling Connect:Direct

Statement Examples

March 31, 2012

Sterling Connect:Direct

Statement Examples

March 31, 2012

Note
Before using this information and the product it supports, read the information in "Notices" on page 149.

This edition applies to the March 31, 2012 version of IBM Sterling Connect:Direct Process Language Reference Guide and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. PROCESS Statement Examples

Basic PROCESS Statement

This example illustrates the minimum requirements of a PROCESS statement. The label CD1 must begin in column one. PROCESS is the statement identifier and is required. The only required parameter is SNODE, which specifies the secondary node to be used in the Sterling Connect:Direct® Process.

```
CD1 PROCESS SNODE=CD.VM.NODE
```

Detailed PROCESS Statement Using the RETAIN Parameter

According to parameters specified, this Process runs in CLASS 4 and is assigned the highest priority, 15. The parameter RETAIN=INITIAL causes the Process to remain in the queue (TCQ) after execution and run every time the Sterling Connect:Direct system is initialized. In addition, accounting data is specified for both nodes (PACCT and SACCT).

```
CDACCT  PROCESS  SNODE=CD.OS390.NODE               -
                 CLASS=4                          -
                 PRTY=15                          -
                 PACCT='ACCOUNTING DATA FOR PNODE'  -
                 RETAIN=INITIAL                   -
                 SACCT='INFORMATION FOR SNODE'
```

Detailed PROCESS Statement Using the NOTIFY Parameter

The priority for this Process is set to 8 and runs in CLASS 4. Because of the NOTIFY parameter, USER1 will be notified upon Process completion.

For Sterling Connect:Direct for UNIX, Sterling Connect:Direct for VSE, Sterling Connect:Direct for OpenVMS, and Sterling Connect:Direct for HP NonStop nodes, NOTIFY is ignored.

```
PROC1  PROCESS            -
       SNODE=CD.AS400     -
       PRTY=8             -
       NOTIFY=USER1       -
       CLASS=4            -
       SNODEID=(USER1,PWD)
```

Use %PNODE

This Process shows using a variable to substitute to %PNODE. %PNODE is set to the node name from which the Process is submitted. (%PNODE is only valid for Sterling Connect:Direct for z/OS® nodes.)

```
PROC1  PROCESS      &NODE=%PNODE
STEP1  COPY  FROM  (PNODE DSN=JSMITH.FILE DISP=SHR)  -
             TO    (SNODE DSN=JSMITH.FILE DISP=NEW)
       IF (STEP1 EQ 0)
             RUN TASK (PGM=USERPGM                   -
                       PARM='&NODE, &DSN')
       EIF
```

Specify a List of Ciphers in a Particular Process

This example involves overriding default settings in the Sterling Connect:Direct Secure Plus parameter files used to establish a connection between two business partners. The business partners agreed by default all sessions are non-secure but that when a secure communication line is required for a particular session, they would use the SSL protocol and a list of cipher suites in a specific order.

Although the SSL protocol is not enabled in the Sterling Connect:Direct Secure Plus parameter files, the remote node records specify OVERRIDE=Y, and all other parameters required to perform the handshake to establish an SSL session are defined.

To specify that the session for this PROCESS is to be secure using SSL and to tell Sterling Connect:Direct to use a specific list of cipher suites, the business partners use the following PROCESS statement:

```
SSLCIPHERS PROCESS SNODE=OTHERBP
SECURE=(SSL,(SSL_RSA_WITH_3DES_EDE_CBC_SHA,SSL_RSA_AES_128_SHA,
SSL_RSA_AES_256_SHA,SSL_RSA_WITH_DES_CBC_SHA) )
```

The four cipher suites are listed in the order of preference, and the first one that matches a cipher suite defined for the other node is used to establish a session.

Encrypt Only Control Block Information—Not Data Being Sent

This example involves overriding default settings in the Sterling Connect:Direct Secure Plus parameter files used to establish a connection between two business partners. The business partners agreed by default all sessions are secure and that everything should be encrypted, that is, both the information sent during the handshake to set up communication sessions and the actual files being transferred.

Both partners specified the following configuration in their Sterling Connect:Direct Secure Plus parameter files:

- Specified ENCRYPT=Y in both the Local and Remote Node records
- Specified OVERRIDE=Y in both the Local and Remote Node records

To not go through the expense of encrypting and decrypting data being transferred, they use the following PROCESS statement when transferring a particular file:

```
ENCNO PROCESS SNODE=OTHERBP SECURE=ENCRYPT.DATA=N
```

In this scenario, both business partners are more concerned with increasing throughput and using less CPU while protecting the information being exchanged to establish the session.

Note: Both sides must have support for ENCRYPT.DATA=N or the Process fails.

Use %DD2DSN to Pass DSN from JCL to a Process

In this example, Sterling Connect:Direct uses the DSN specified on the allocated FROMDD statement in the JCL and a randomly generated six- digit number to copy to a data set with a unique name.

```
TEST  PROCESS  SNODE=S_NODE               -
               &RAN = %PRAND              -
               &DSN = %DD2DSN(FROMDD)
STEP  COPY  FROM  (DSN=&DSN) TO (XXX.T&RAN)
```

Use a TCP/IP Address for the SNODE Keyword (IPv4)

This Process shows how to code the TCP/IP address using the IPv4 standard dotted-name format when the real address is known to the user.

```
PROC1  PROCESS  PNODE=CD.OS390.NODE          -
                SNODE=TCPNAME=111.222.333.444 -
                NOTIFY=USER1
```

Use a TCP/IP Address for the SNODE Keyword (IPv6)

This Process shows how to code the TCP/IP address using the IPv6 format with colons when the real address is known to the user.

```
PROC1  PROCESS  PNODE=CD.OS390.NODE                -
                SNODE=TCPNAME=1111::6666:7777:8888  -
                NOTIFY=USER1
```

Use TCPNAME to Identify the PNODE/SNODE Sites

This Process shows how the user can identify the TCP/IP nodes by name when the real network addresses for the nodes are unknown. The names specified must be identified to the TCP/IP network name resolution task.

```
PROC1  PROCESS  PNODE=CD.OS390.NODE   -
                SNODE=RESTON.SCENTER  -
                NOTIFY=USER1
```

HP NonStop PROCESS Statement

The following is an example HP NonStop PROCESS statement.

```
PROC1   PROCESS         SNODE=CD.NODE
                        SNODEID=(JONES,OPENUP)
                        CLASS=4
                        HOLD=YES
                        PACCT='OPERATIONS, DEPT. 87'
                        RETAIN=NO
```

The Process named PROC1 specifies a secondary node (SNODE) of CD.NODE. The corresponding security user IDs and passwords (SNODEID) have been included.

This Process will run in CLASS 4.

The Process will be placed on the Hold queue until it is released for execution with a CHANGE PROCESS command.

The PACCT parameter specifies that all accounting information will be attributed to the operations account, department 87, if the node has a program that maintains this information.

Once the Process executes, it will be deleted because the RETAIN parameter is set to NO. Note that NO is the default value for the RETAIN parameter.

Example OpenVMS PROCESS Statement

The following is a example PROCESS statement.

```
PROC1     PROCESS     SNODE=CD.NODE.A
                      SNODEID=(JONES,OPENUP)
                      CLASS=4
                      HOLD=YE
                      PACCT='OPERATIONS, DEPT. 87'
                      RETAIN=NO
```

The Process named PROC1 specifies a secondary node (SNODE) of CD.NODE.A. The corresponding security user IDs and passwords (SNODEID) are included.

This Process will run in CLASS 4.

The Process will be placed on the Hold queue until it is released for execution with a CHANGE PROCESS command.

The PACCT parameter specifies that all accounting information will be attributed to the operations account, department 87, if the node has a program that maintains this information.

After the Process executes it will be deleted because the RETAIN parameter is set to NO.

Example VMESA Process

The Process named PROC1 specifies a secondary node (SNODE) of CD.NODE.A. The corresponding security user IDs and passwords (SNODEID) are included.

```
PROC1  PROCESS  SNODE=CD.NODE.A              -
                SNODEID=(JONES,OPENUP)       -
                CLASS=4                      -
                HOLD=YES                     -
                NOTIFY=%USER                 -
                PACCT='OPERATIONS, DEPT. 87' -
                RETAIN=NO
```

This Process will run in CLASS 4.

The Process is placed in the Hold queue until it is released for execution with a CHANGE PROCESS command. As indicated by the NOTIFY parameter, the VM user who submitted the Process is notified upon completion of the Process.

The PACCT parameter specifies that all accounting information will be attributed to the operations account, department 87, if the node has a program that maintains this information.

After the Process executes, it is deleted because the RETAIN parameter is set to NO.

VSE PROCESS Statement

The Process named PROC1 specifies a secondary node (SNODE) of CD.NODE.A. The corresponding security user IDs and passwords (SNODEID) have been included.

```
PROC1  PROCESS  SNODE=CD.NODE.A              -
                SNODEID=(JONES,OPENUP)       -
                CLASS=4                      -
                HOLD=YES                     -
                PACCT='OPERATIONS, DEPT. 87' -
                RETAIN=NO
```

This Process will run in CLASS 4.

The Process will be placed on the Hold queue until it is released for execution with a CHANGE PROCESS command.

The PACCT parameter specifies that all accounting information will be attributed to the operations account, department 87, if the node has a program that maintains this information.

After the Process executes, it is deleted because the RETAIN parameter is set to NO.

Use Symbolics in a UNIX Process

This example shows a UNIX Process that uses symbolics to specify the file and data set names at submission. Process accounting data is specified for the **pnode** and **snode**.

```
copyseq  process      snode=dallas
                      pacct="dept-59"
                      sacct="dept-62"
step01   copy  from  (file=&file)
               to    (file=&dsn
                      snode)
  pend
```

The following **submit** command specifies the file and data set names to be used in a file transfer.

```
submit  file=copyseq
        &file=myfile
        &dsn=abc;
```

The following Process is generated by the previous input:

```
copyseq  process      snode=dallas
                      pacct="dept-59"
                      sacct="dept-62"
step01   copy  from  (file=myfile)
               to    (file=abcsnode)
  pend
```

Chapter 2. COPY Statement Examples

HP NonStop

Copy a File from Sterling Connect:Direct for HP NonStop to Sterling Connect:Direct for z/OS

This Process copies a file from a Sterling Connect:Direct for HP NonStop node to a new SMS-controlled file at a Sterling Connect:Direct for z/OS node.

```
SMS1   PROCESS  PNODE=CD.HPNONSTOP                -
                SNODE=CD.OS390.NODE
STEP1  COPY     FROM  (PNODE                      -
                       DSN=$VOL.SUBVOL.TANFILE    -
                       DISP=SHR                   -
                TO    (SNODE                     -
                       DISP=(NEW,CATLG)           -
                       DSN=DATA1.SEQ.OS390FILE    -
                       DCB=(DSORG=PS, LRECL=80)   -
                       SPACE=(80,(1500,100),RLSE) -
                       UNIT=SYSDA                 -
                       SYSOPTS="AVGREC=U          -
                       MGMTCLAS=TEST01            -
                       STORCLAS=TEST01            -
                       DATACLAS=TEST01")
```

Submit a Process from a Sterling Connect:Direct for HP NonStop Node that Copies a File from z/OS to HP NonStop

This Process, submitted from Sterling Connect:Direct for HP NonStop, copies (pulls) a file from Sterling Connect:Direct for z/OS to the HP NonStop node. The TO file specification contains a "SET TYPE E" parameter in SYSOPTS, in case the file is not present and must be created. The FROM file normally needs no file attributes (DCB or SYSOPTS). The XLATE option is used to specify translation. This must always be specified in the SYSOPTS clause associated with the HP NonStop dataset. A checkpoint interval of 10 MB is specified.

```
EPROC  PROCESS     SNODE=CD.OS390                 -
                   SNODEID=(UID,USER_PASSWORD)
PULL   COPY  TO   (PNODE DISP=RPL                 -
                   DSN=$WORK.DATA.EFILE           -
                   SYSOPTS="SET TYPE E XLATE ON") -
             FROM (SNODE DISP=SHR                 -
                   DSN=OFILES.ACH.D120897)        -
                   CKPT=10M
```

Copy a Code 0 Unstructured file from HP NonStop to z/OS

This Process copies an unstructured file to a z/OS node. Because an Enscribe unstructured file has no intrinsic record length, the DCB parameter is used on the FROM file specification to define the record length with which the file is read.

When an unstructured file is sent from an HP NonStop system, the default LRECL is the "buffersize" attribute of the file, unless overridden by the DCB parameter.

```
UNSTR    PROCESS      SNODE=CD.OS390
STEP1    COPY  FROM  (PNODE DISP=SHR            -
                      DSN=$WORK02.HPDATA.UNST1  -
                      DCB=(LRECL=100))          -
                TO    (SNODE DISP=NEW           -
                      DSN=DATA1.VSAME001.DATA   -
                      DCB=(DSORG=PS             -
                          ,RECFM=FB             -
                          ,LRECL=100            -
                          ,BLKSIZE=10000))      -
                      CKPT=1M
```

Copy a File from HP NonStop to z/OS After Running DMRTDYN on z/OS

In this Process, STEP1 will invoke DMRTDYN to delete and unallocate DATA1.SEQ.KSDSFILE at the SNODE. Then $B.FILETEST.KSDSFILE will be copied from the HP NonStop node to DATA1.SEQ.VSAMKSDS at the HP NonStop node.

```
VSAMKS  PROCESS     PNODE=CD.HPNONSTOP                                  -
                    SNODE=CD.OS390.NODE
STEP1   RUN TASK   (PGM=DMRTDYN,                                        -
                    PARM=(C'ALLOC DSN=DATA1.SEQ.KSDSFILE                -
                    DISP=(OLD,DELETE)'                                  -
                          F'-1'                                         -
                          C"UNALLOC DSN=DATA1.SEQ.KSDSFILE")            -
                    SNODE)
STEP2   COPY FROM  (DSN=$B.FILETEST.KSDSFILE                            -
                    PNODE                                               -
                    DISP=SHR)                                           -
        TO         (DSN=DATA1.SEQ.VSAMKSDS                              -
                    DISP=NEW                                            -
                    SPACE=(2048,(2048,10))                              -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=1024,BLKSIZE=2048))
```

Copy a File from an HP NonStop Spooler to a z/OS Node

In this multi-step Process, STEP1 will execute a RUN TASK to invoke DMRTDYN at the SNODE to delete DATA1.SEQ.EXTOS390F. STEP2 will copy a file from the HP NonStop spooler to DATA1.SEQ.EXTOS390F at the SNODE. Because a spooler job number was not specified, the most recent job in the spooler for the job owner will be copied. A record format of FBA is used to maintain ANSI control characters.

```
EXTOS390  PROCESS    PNODE=CD.HPNONSTOP                      -
                     SNODE=SS.CD.OS390
STEP1     RUN TASK  (PGM=DMRTDYN,                           -
                     PARM=(C'ALLOC DSN=DATA1.SEQ.EXTOS390F,  -
                     DISP=(OLD,DELETE)'                      -
                     F'-1'                                   -
                     C"UNALLOC DSN=DATA1.SEQ.EXTOS390F"))    -
                     SNODE
STEP2     COPY FROM (DSN=\TANEXT.$S.#EXTTANF                 -
                     PNODE                                   -
                     DISP=SHR)                               -
               TO    (DSN=DATA1.SEQ.EXTOS390F                -
                     SNODE                                   -
                     DISP=(NEW,CATLG)                        -
                     DCB=(DSORG=PS,RECFM=FBA,LRECL=132,      -
                     BLKSIZE=13200)                          -
                     SPACE=(13200,(10,2))                    -
                     UNIT=SYSDA)
```

Copy Between a z/OS Node and a Remote HP NonStop Spooler on an EXPAND Network

This multi-step Process copies between a z/OS node and a remote HP NonStop spooler on an EXPAND network. The Process is submitted from the HP NonStop node. STEP01 copies a file from a DSN on the z/OS node to the HP NonStop spooler. STEP02 copies a file from the HP NonStop spooler to a z/OS node.

Note that \SYSEXT is an EXPAND node other than the one that the Sterling Connect:Direct for HP NonStop server resides on. Because the HP NonStop files are spooler files, the SYSOPTS SET XLATE subparameter does not have to be specified for translation; spooler files and edit files (unstructured, code 101) are translated automatically. A record format of FBA is used to maintain ANSI control characters.

```
SPL     PROCESS      SNODE=CD.SMITH
STEP01  COPY  FROM  (DSN=JSMITH.CDACT SNODE             -
                     DISP=SHR)                          -
              TO    (FILE='\SYSEXT.$S.#SPL1' PNODE      -
                     DISP=RPL
STEP02  COPY  FROM  (FILE='\SYSEXT.$S.#SPL2' PNODE      -
                     DISP=SHR)                          -
              TO    (DSN=RJONES.ACCT SNODE              -
                     DCB=(RECFM=FBA,DSORG=PS,LRECL=132,  -
                     BLKSIZE=13200)                     -
                     SPACE=(13200,(10,2))               -
                     DISP=RPL)
```

Copy Files Between the HP NonStop Spooler System and a z/OS Node Using Job Number

This Process transfers a file from the HP NonStop spooler $S to a z/OS sequential file. The HP NonStop file has a job number of 3722 and a location (name) of #SPLFILE. The SPOOLER command is used to specify a supervisor other than the default of $SPLS. A record format of FBA is used to maintain ANSI control characters.

```
SPLPROC  PROCESS      SNODE=CD.OS390.JSMITH
STEP01   COPY  FROM  (DSN=\SYSEXT.$S.#SPLFILE             -
                      SYSOPTS=("SET SPOOLER=$SPLA"        -
                               "SET SPOOLNUM=3722")       -
                      DISP=SHR)                           -
               TO    (DSN=RJONES.SPLFILE                  -
                      DCB=(RECFM=FBA,DSORG=PS,LRECL=132,  -
                      BLKSIZE=13200)                      -
                      SPACE=(1320,(10,2))                 -
                      DISP=RPL)
```

Copy a Disk File from HP NonStop to a Tape Device at z/OS

This Process is submitted on the HP NonStop node to copy a disk file on HP NonStop to a tape device at the z/OS node. Because NL (no labels) is specified, DCB attributes must be specified to identify the file on the volume.

```
PROCESS1  PROCESS      PNODE=CD.HPNONSTOP                                -
                       SNODEID=(RJONES,ROGER)                            -
                       SNODE=CD.OS390
STEP01    COPY  FROM  (DSN=$C.BILLPROC.ACCTDATA                         -
                       DISP=SHR)                                         -
                TO    (DSN=RJONES.ACCTTAPE                              -
                       DISP=RPL                                          -
                       DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=4096)     -
                       VOL=(,,,1)                                        -
                       LABEL=(,NL,,EXPDT=91044)                          -
                       UNIT=REEL)
```

Copy a Tape File from z/OS to a Disk File on HP NonStop

This Process is submitted at the HP NonStop node to copy a tape file from the z/OS node to a disk file on the HP NonStop node. Because NL (no labels) is specified, DCB attributes must be specified to identify the file on the volume. SYSOPTS is used to specify file creation parameters specific to HP NonStop.

```
PROCESS1  PROCESS      PNODE=CD.HPNONSTOP                                -
                       SNODEID=(RJONES,ROGER)                            -
                       SNODE=CD.OS390
STEP01    COPY  FROM  (DSN=RJONES.ACCTTAPE SNODE                        -
                       UNIT=REEL                                         -
                       VOL=SER=010196                                    -
                       LABEL=(,NL,,EXPDT=91044)                          -
                       DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=4096)     -
                       DISP=SHR)                                         -
                TO    (PNODE                                            -
                       DSN=$C.BILLPROC.ACCTDATA                          -
                       DISP=RPL                                          -
                       SYSOPTS=("SET TYPE U"                             -
                                "SET BLOCK 4096"                         -
                                "SET EXT (5,5)"))
```

Copy a File from z/OS to HP NonStop Using the FASTLOAD Option

This Process is submitted at the HP NonStop node to copy an entry-sequenced file from z/OS to a key-sequenced file at the HP NonStop node. The SYSOPTS subparameter SET FASTLOAD SORTED sets FASTLOAD and indicates to FUP that the data is sorted.

This option (particularly useful for key-sequenced files) will bypass invocation of FASTSORT by FUP. The FASTLOAD option can be used to reduce disk I/O overhead. The XLATE subparameter is included in the Process to turn on the text conversion utility and translate from EBCDIC to ASCII.

```
FASTLOAD  PROCESS      PNODE=CD.HPNONSTOP                -
                       SNODE=CD.OS390
STEP01    COPY  FROM  (DSN=OS390.ESDS.FILE               -
                       DISP=SHR SNODE)                   -
                TO    (DSN=$C.DATA.KSDS                 -
                       DISP=RPL                          -
                       SYSOPTS=("SET FAST.LOAD.SORTED Y"  -
                                "SET XLATE ON") PNODE)
```

Allocate a VSAM Data Set and Copying a File from HP NonStop to z/OS

This Process invokes Sterling Connect:Direct DMRTAMS for VSAM file allocation. DMRTAMS (using IDCAMS commands specified in the Process) first deletes the old data set if it exists. It then defines and allocates it for use in the subsequent COPY statement. The file transmitted is an HP NonStop key-sequenced file.

```
VSAM01  PROCESS    SNODE=SC.OS390.RSMITH PNODE=CDCLX             -
                   SNODEID=(RMITH,RSMITH)                        -
                   SACCT='CHARGE TO ACCOUNTING'
STEP1   RUN TASK  (PGM=DMRTAMS,                                  -
                   PARM=(C"FREE=CLOSE,RETURN=(DD) SYSOUT=X"      -
                   C" DELETE (RSMITH.VSAM01.OUTPUT) CLUSTER  ",  -
                   C" DEFINE CLUSTER                        -",  -
                   C"   (NAME(RSMITH.VSAM01.OUTPUT)         -",  -
                   C"    RECORDS(2500 100)                  -",  -
                   C"    VOLUMES(M80003)                    -",  -
                   C"    INDEXED                            -",  -
                   C"    FREESPACE(0 0)                     -",  -
                   C"    NOIMBED                            -",  -
                   C"    KEYS (8 6)                         -",  -
                   C"    RECORDSIZE(262 880)                -",  -
                   C"    NOREPLICATE                        -",  -
                   C"    SHAREOPTIONS(2))                   -",  -
                   C"   DATA                                -",  -
                   C"    (CONTROLINTERVALSIZE(4096)         -",  -
                   C"     NAME(RSMITH.VSAM01.OUTPUT.DATA2)  -",  -
                   C"   INDEX                               -",  -
                   C"    (CONTROLINTERVALSIZE(512)          -",  -
                   C"     NAME(RSMITH.VSAM01.OUTPUT.INDEX2))"))  -
                   SNODE
STEP2   COPY FROM (DSN=$B.CDFILES.KDSDFIL DISP=SHR               -
                   PNODE)                                        -
             TO   (DSN=RSMITH.VSAM01.OUTPUT DISP=SHR             -
                   SNODE)
```

Copy to an Entry-Sequenced File (HP NonStop to HP NonStop)

In this multi-step Process, STEP01 will execute FUP to purge $B.FILERESO.A11025 on the PNODE. A message will be sent to the spooler ($S.#FUPTEST) that indicates whether FUP executed successfully.

STEP02 will copy $B.FILEDATA.ETYPE from the PNODE to the entry-sequenced file, $B.FILERESO.A110257, at the SNODE. Specifying the SYSOPTS subparameter TYPE=E ensures that file attribute defaults defined in the type file E will be used when creating the file.

```
A110257  PROCESS     PNODE=CD.TANA                  -
                     SNODE=CD.TANB
STEP01   RUN TASK    (PGM=FUP                       -
                     SYSOPTS=('/OUT $S.#FUPTEST/',  -
                              'VOLUME $B.FILERESO', -
                              'PURGE A110257  ')    -
                     PNODE)
STEP02   COPY  FROM  (DSN=$B.FILEDATA.ETYPE         -
                     PNODE                          -
                     DISP=SHR)                      -
               TO    (DSN=$B.FILERESO.A110257       -
                     SNODE                          -
                     DISP=NEW                       -
                     SYSOPTS="SET TYPE E")
```

Copy Files Between HP NonStop Spooler Systems

This Process selects a spooler job by job number and copies it to another HP NonStop spooler system at a remote node. The HP NonStop file being copied has a job number of 3722 and a location (name) of #SPLFILA. The SPOOLER command is used to specify a supervisor other than the default of $SPLS.

```
SPLPROC  PROCESS      PNODE=CD.TANA                -
                      SNODE=CD.TANB
STEP01   COPY  FROM   (DSN=\SYSA.$S.#SPLFILA       -
                      SYSOPTS=("SET SPOOLER=$SPLA" -
                      "SET SPOOLNUM=3722")         -
                      DISP=SHR PNODE)              -
                TO    (DSN=\SYSB.$S.#SPLFILB       -
                      DISP=NEW SNODE)
```

Copy Text Files from HP NonStop to UNIX

This Process, submitted from the Sterling Connect:Direct for HP NonStop node, will copy datafile in $B.smith to /payroll/monthly/jan on the Sterling Connect:Direct for UNIX node. For Sterling Connect:Direct for UNIX nodes, the security user IDs and passwords are case sensitive.

```
unix1  process      snode=unix.node snodeid=(user,user1)
step1  copy  from  (file=$B.smith.datafile              -
                    pnode)                              -
             to    (file='/payroll/monthly/jan '        -
                    snode                               -
                    sysopts=":datatype=text:"           -
                    disp = rpl)                         -
                    ckpt=128K
```

Copy Binary Files from HP NonStop to UNIX

This Process, submitted from the Sterling Connect:Direct for HP NonStop node, will copy a file in $user.unixdata.cdcom to a binary file on the Sterling Connect:Direct for UNIX node.

```
TOBIN01  PROCESS     SNODE=cd.v1200              -
                     snodeid=(user,pswd)
STEP01   COPY  FROM  (DSN=$user.unixdata.cdcom    -
                     PNODE                       -
                     SYSOPTS=('SET XLATE OFF')   -
                     DISP=SHR)                   -
               TO    (dsn='tdata/cdcomu'         -
                     SNODE                       -
                     SYSOPTS=":datatype=binary:" -
                     DISP=RPL)
```

Copy Binary Files from HP NonStop (OSS) to UNIX

This Process illustrates how to ensure data integrity when copying between a Sterling Connect:Direct HP NonStop (OSS) node and a UNIX node. The destination in STEP01 and source in STEP02 both have sysopts specified to denote a binary transfer.

```
OUOBIN     PROCESS   SNODE=qa160sun3600                          -
                     SNODEID=(qatest,qatest)
           SYMBOL    &FILE1='/home/qatest/input/ndmsrvr'
           SYMBOL    &FILE2='/export/home/users/qatest/tantest/binary'
           SYMBOL    &FILE3='/home/qatest/output/unix/binary'
RUNUNX     RUN TASK  (PGM="UNIX")    SNODE                       -
                      SYSOPTS=("rm -rf &FILE2")
STEP01     COPY TO   (DSN=&FILE2     SNODE                       -
                      DISP = RPL                                  -
                      SYSOPTS=":datatype=binary:")                -
           FROM      (DSN=&FILE1     PNODE                       -
                      DISP = SHR)
STEP02     COPY FROM (DSN=&FILE2     SNODE                       -
                      DISP = SHR                                  -
                      SYSOPTS=":datatype=binary:")                -
           TO        (DSN=&FILE3     PNODE                       -
                      DISP = RPL)
```

Copy Files from HP NonStop to VM

This Process illustrates the transmission of a file from an HP NonStop node to a VM node. The Process is submitted on the HP NonStop node. EBCDIC-to-ASCII translation is requested with the SYSOPTS parameter SET XLATE ON. All SYSOPTS keyword values must be enclosed in parentheses, and the entire SYSOPTS string must be enclosed in double quotation marks.

Use this Process when you copy files from HP NonStop to VM.

```
TANTOVM  PROCESS     PNODE=DALL.TX                               -
                     SNODE=CD.VM.BOSTON                          -
                     SNODEID=(IDXXXX,PASSWD)                     -
                     SACCT='TRANSFERRING FROM HP NONSTOP TO VM.'
STEP01   COPY  FROM  (DSN=$B.SMITH.DATAFILE                      -
                     DISP=(SHR)                                  -
                     SYSOPTS=("SET XLATE ON")                    -
                     PNODE)                                      -
               TO    (DSN='TEST FILE'                            -
                     LINK=(TRA,WPSWD,W,191)                      -
                     DISP=(RPL)                                  -
                     DCB=(RECFM=F, LRECL=80, BLKSIZE=80)         -
                     SNODE)
```

Copy an HP NonStop Key-Sequenced File to a Sterling Connect:Direct for OpenVMS Node

This Process, submitted from the Sterling Connect:Direct for HP NonStop node, copies a key-sequenced file to the Sterling Connect:Direct for OpenVMS node. When Processes are submitted from Sterling Connect:Direct for HP NonStop to Sterling Connect:Direct for OpenVMS nodes, OpenVMS file names must be in single quotation marks. Note that COMPRESS is coded between the FROM and TO clauses of the COPY statement.

```
SEND1   PROCESS      PNODE=CD.HPNONSTOP                        -
                     SNODE=CD.VMS
STEP01  COPY  FROM  (PNODE FILE=$B.TESTF.KSDS                  -
                     DISP=SHR)                                 -
                     COMPRESS PRIMECHAR=C'*'                   -
              TO    (SNODE FILE='[DUC4:-DATA.FILE]KSDS1.DAT' -
                     DCB=(DSORG=KSDS                           -
                     LRECL=100                                 -
                     KEYLEN=10                                 -
                     RECFM=F)                                  -
                     DISP=RPL)
```

Copy Files from HP NonStop to VSE

This Process illustrates the transmission of files from an HP NonStop node to a VSE node. The parameter XLATE must be set to ON and positioned in the Process where the HP NonStop file is specified. XLATE translates the file from ASCII to EBCDIC. Symbolics will be resolved at submission.

```
SEND2VSE  PROCESS      PNODE=BOSTON.NODE                                  -
                       SNODE=CD.VSE.NODE                                  -
                       SNODEID=(IDXXXX,PSWD)                              -
                       SACCT='CHARGE TO GROUP 199'
TAN_VSE   COPY  FROM  (DSN=&FROM                                          -
                       SYSOPTS=("SET XLATE ON")                           -
                       PNODE)                                             -
                TO    (DSN=&TO                                           -
                       UNIT=DLBLONLY                                      -
                       DISP=(OLD)                                         -
                       DCB=(RECFM=FB, LRECL=80, BLKSIZE=80, DSORG=PS)     -
                       SNODE)
```

Copy Binary Files from Microsoft Windows to HP NonStop (OSS)

This Process illustrates how to ensure data integrity when copying between a Microsoft Windows system and an HP NonStop (OSS) node. The destination in STEP01 and source in STEP02 both have sysopts specified to denote a binary transfer.

```
BINARY     PROCESS   SNODE=W2S.4100.Cdwops8                        -
                     SNODEID=(qatest,qatest)
           SYMBOL    &FILE1='/home/qatest/input/ndmsrvr'
           SYMBOL    &FILE2='c:\output\binary'
           SYMBOL    &FILE3='/home/qatest/output/win/binary'
RUNWIN     RUN TASK  (PGM="WindowsNT")              SNODE        -
                      SYSOPTS="CMD(ERASE &FILE2)"
STEP01     COPY TO   (DSN=&FILE2     SNODE                        -
                      DISP = RPL                                 -
                      SYSOPTS="datatype(binary)")                -
           FROM      (DSN=&FILE1     PNODE                        -
                      DISP = SHR)
STEP02     COPY FROM (DSN=&FILE2     SNODE                        -
                      DISP = SHR                                 -
                      SYSOPTS="datatype(binary)")                -
           TO        (DSN=&FILE3     PNODE                        -
                      DISP = RPL)
```

Copy a File from HP NonStop to VM

This example copies a file from HP NonStop to VM. It uses conditional statements to check the completion code from STEP01. If the completion code equals 0, then a message is issued to the operator that indicates that the transfer was successful. If the completion code is any value other than 0, then a message is issued to the operator that indicates that the transfer failed.

```
PROC01     PROCESS     SNODE=CD.VM
                       PNODE=CD.HP NONSTOP
STEP01     COPY  FROM  (FILE=$B.ABC.FILEA PNODE
                       SYSOPTS='SET XLATE ON')
                 TO    DSN=JKL.FILEA
                       DISP=(RPL), LINK=(CDVM,MULT,MW,551))
STEP02     IF          (STEP01 EQ 0) THEN
NOTIFY1    RUN TASK    (PGM=DMNOTFY2
                       SYSOPTS=("GOOD,'TRANS FOR COMPANY A COMPLETED DISK 502',
                       OPERATOR"))
                       SNODE
           EXIT
           ELSE
NOTIFY2    RUN TASK    (PGM=DMNOTFY2
                       SYSOPTS=("FAIL,'TRANS FOR COMPANY A FAILED DISK 502',
                       OPERATOR"))
                       SNODE
           EIF
           EXIT
```

Copy a Data Set from a Spooler File on Sterling Connect:Direct HP NonStop to Sterling Connect:Direct z/OS

This example statement copies a data set from a spooler file on a Sterling Connect:Direct for HP NonStop node to a Sterling Connect:Direct z/OS node. In this example, the specified spooler supervisor name overrides the default of $SPLS. Job=253 specifies the spooler file uniquely.

```
STEP1      COPY  FROM    (DSN=$S.#FILE2
                          DISP=SHR PNODE
                          SYSOPTS=("SET SPOOLER=$SPLA"
                                   "SET SPOOLNUM=253"))
                 TO      (DSN=MVSSITE.DESTFILE
                          DISP=NEW SNODE)
```

Copy a File From HP NonStop to HP NonStop and Overriding SENDOPENFILE with OPENFILEXMT

```
COPY  FROM  (DSN=$dev.datain.openes                    -
             SYSOPTS="SET OPENFILEXMT=Y"               -
             PNODE    DISP=SHR)                        -
      TO    (DSN=$dev.dataout.openfres SNODE DISP=RPL) -
             CKPT=20k
```

i5/OS

Copy an HP NonStop File to an i5/OS Node

This Process copies an HP NonStop file to a member of a physical data base file on i5/OS. The i5/OS file is created with maximum members specified as 100. The HP NonStop file is translated from ASCII to EBCDIC.

```
TESTPROC  PROCESS      PNODE=CD400               -
                       SNODE=CDTAN               -
                       SNODEID=(USER1,PASSWRD)
STEP01    COPY  FROM  (DSN=$SYSTEM.SOURCE.FILE   -
                       DISP=SHR PNODE)           -
                       SYSOPTS=("SET XLATE ON")  -
                TO    (DSN='CDTAN/SOURCE(FILE)' -
                       DISP=NEW SNODE            -
                       SYSOPTS=("TYPE(MBR)       -
                       MAXMBRS(100)"))
```

Microsoft Windows

Copy a File from Microsoft Windows to z/OS

This Process copies a text file from a remote Microsoft Windows system (where Sterling Connect:Direct is not installed) to a z/OS partitioned data set (PDS). When copying a file to or from a remote Microsoft Windows computer, you must use the Universal Naming Convention (UNC) method for specifying the file name. Do not use a drive letter. The UNC consists of two backslashes, the computer name, a single backslash, and the share name. The share, ROOT_C, must be accessible to Sterling Connect:Direct for Microsoft Windows server. In this example, the computer name is WIN_SYS1 and the share name is ROOT_C.

```
NT2OS390  PROCESS      SNODE=SS.OS390/*$OS390$*/
                       HOLD=NO
                       CLASS=1
                       PRTY=10
                       EXECPRTY=10
                       RETAIN=NO
STEP01    COPY  FROM  (FILE=\WIN_SYS1\ROOT_C\DATA\OUT\SALESJAN.DAT
                       PNODE/*$WINDOWS$*/
                       SYSOPTS="datatype(text)")
                TO    (SNODE/*$OS390$*/
                       FILE=SALES.DATA.JAN(MBR99)
                       DISP=(RPL,CATLG))
 PEND
```

Copy a File from Microsoft Windows to HP NonStop

This Process copies a text file from a Microsoft Windows system to an HP NonStop node. Each system operation is enclosed in single quotation marks. Double quotation marks enclose the entire SYSOPTS statement.

```
NT2TAN  PROCESS      SNODE=SS.TAN/*$HP NONSTOP$*/
                     HOLD=NO
                     CLASS=1
                     PRTY=10
                     EXECPRTY=10
                     RETAIN=NO
                     PNODEID=(USR1,PSWD1)
                     SNODEID=(SS.USER01,PSWD01)
STEP01  COPY  FROM  (FILE=C:\OUTPUT\BINARY\SALES.JAN
                     PNODE/*$WINDOWS$*/
                     SYSOPTS="DATATYPE(BINARY)")
              TO    (SNODE/*$HP NONSTOP$*/
                     FILE=$SALES.DATA.JAN)
                     DISP=(RPL, ,DELETE)
                     SYSOPTS="'SET CODE O' 'SET TYPE U' 'SET EXT(700 300'
                              'SETBLOCK 4096' 'SETMAXEXTENTS600'")
                     COMPRESS PRIMECHAR=X'20'
 PEND
```

Wildcard Copies from Microsoft Windows to UNIX

In the following example, a Sterling Connect:Direct for Microsoft Windows PNODE directory C:\financial\accounts contains the files customer1, customer2, customer3, supplier1, and supplier2. A Sterling Connect:Direct for UNIX SNODE has the directory /financial/accounts. The following wildcard copy command copies the files called customer1, customer2, and customer3 from the PNODE to the /financial/accounts directory on the SNODE. The source file names and the destination file names are identical.

```
WILDCOPY  COPY
                FROM  (FILE=C:\financial\accounts\customer?)
                TO    (FILE=/financial/accounts/
                       DISP=RPL)
```

When specifying a path and filename on Microsoft Windows, you can use the standard Microsoft Windows format when sending from C: or D: as show in the FROM parameter above. If you send from any other drive, you must use the UNC format such as \servername\financial\accounts\customer?.

You must include the ending forward slash (/) for the destination directory.

The following wildcard copy step copies customer1, customer2, customer3, supplier1, and supplier2 into the /financial/accounts directory on the SNODE. The source file names and the destination file names are identical.

```
WILDCOPY  COPY
                FROM  (FILE=C:\financial\accounts\*)
                TO    (FILE=/financial/accounts/
                       DISP=RPL)
```

Wildcard Copy from Microsoft Windows to Microsoft Windows

To copy send to a Sterling Connect:Direct for Microsoft Windows node, you must include an ending backslash (\) in the **TO FILE=** parameter. Following is an example:

```
WILDCOPY  COPY
                FROM  (FILE=C:\financial\accounts\customer?)
                TO    (FILE=\server1\financial\accounts\
                       DISP=RPL)
```

You can use the standard Microsoft Windows format when sending from C: or D: as show in the FROM parameter above. If you send from any other drive, you must use the UNC format, as shown in the TO parameter above.

You must include the ending backslash (\) for the destination directory.

Wildcard Copy from Microsoft Windows to a z/OS Node

To copy send to sequential files on a Sterling Connect:Direct for z/OS node, you must include an ending period (.) in the **TO FILE=** parameter. Following is an example:

```
WILDCOPY COPY
      FROM (FILE=C:\financial\accounts\*)
      TO   (FILE=FINANCIAL.ACCOUNTS.
            DISP=RPL)
```

To copy send to a PDS on a Sterling Connect:Direct for z/OS node, you must use an asterisk (*) for the PDS member name. Following is an example:

```
WILDCOPY COPY
      FROM (FILE=C:\financial\records\*)
      TO   (FILE=FINANCIAL.RECORDS(*)
            DISP=RPL)
```

When specifying a path and filename on Microsoft Windows, you can use the standard Microsoft Windows format when sending from C: or D: as show above. If you send from any other drive, you must use the UNC format such as \servername\financial\accounts\customer?.

Wildcard Copy from Microsoft Windows to a Node with Download Restrictions

To copy send to a Sterling Connect:Direct node that enforces download restrictions, use an asterisk (*) for the **TO FILE=** parameter if the destination directory is the download directory. Following is an example:

```
WILDCOPY  COPY
                FROM  (FILE=C:\financial\records\*)
                TO    (FILE=*
                       DISP=RPL)
```

Copy a File from Microsoft Windows to z/OS

This Process copies a file from a Microsoft Windows node to a z/OS node, specifying that the data type is text and blanks are to be left in the file. The comments document the node. The file name on the from side uses the UNC format.

```
copyseq     process     snode=OS390.v4200 /*$OS390$*/
                        hold=no
                        class=1
                        prty=10
                        execprty=10
                        retain=no
                        pnodeid=(user01,pw01)
                        snodeid=(user01,pw01)
step01      copy  from  (
                        file=\srv-one\c_drive emp\ntfile.txt
                        pnode/*$WindowsNT$*/
                        sysopts="datatype(text) strip.blanks(no)
                        xlate(yes)")
                        ckpt=1k
                        compress extended
                  to    (
                        snode /*$OS390$*/
                        file=user01.newfile2
                        disp=(rpl,catlg))
```

Use Symbolics In a Microsoft Windows Copy

This Process (named copyseq) copies a file from Microsoft Windows to z/OS. It uses symbolics to specify the file names to be copied. It also specified accounting data for the pnode and snode.

```
copyseq     process     snode=OS390.dallas
                        localacct="dept-59"
                        remoteacct="dept-62"
step01      copy  from  (file="&homefile")
                  to    (file="dalfile"
                        snode)
```

This submit command submits the copyseq Process and specifies the file names to use in the copy.

```
submit     file=copyseq
           &homefile="c:\mydir\myfile.txt"
           &dalfile="user01.newfile"
```

CODEPAGE Conversion During a File Copy (Microsoft Windows to z/OS)

The following example shows how to specify the CODEPAGE SYSOPTS parameter when copying a file from an ANSI Latin Sterling Connect:Direct for Microsoft Windows system (UTF-8 codepage) to a U.S. or Canadian Sterling Connect:Direct for z/OS system (IBM-1047 codepage).

```
CODEPGNT  PROCESS      SNODE=SNBEACH.OS390
                       HOLD=NO
CODEPGCP  COPY  FROM  (PNODE
                       DSN='C:\SRN_TESTDATA\MIXED.TXT'
                       DISP=SHR
                       SYSOPTS="CODEPAGE=(1252,UTF-8)")
                       CKPT=5M
                TO    (SNODE
                       DSN=SNBEAC1.DATAFILE.MIXED
                       DISP=RPL
                       SYSOPTS="CODEPAGE(UTF-8,IBM-1047)")
```

OpenVMS

Copy PDS Members from z/OS to OpenVMS

This Process copies a PDS and all of its members from z/OS to a text library on OpenVMS. Note that the string of SYSOPTS parameters is enclosed in double quotation marks.

```
PDSCPY1  PROCESS      SNODE=CD.OS390.NODE
STEP01   COPY  FROM  (DSN=SMITH.INPUT83.MEMBERS SNODE)        -
               TO    (DSN=DUA0:[RJONES.CDTEST]PDSTEST.TLB     -
                      DISP=OLD                                -
                      DCB=(DSORG=PO)                          -
                      SYSOPTS="LIBRARY='TEXT' REPLACE" PNODE)
```

Copy a File from Disk to Tape (OpenVMS)

The following example copies an OpenVMS file from disk to tape. Specifying the /OVERRIDE qualifier causes the name of the tape volume to be ignored. The /OVERRIDE qualifier can be added either to the MOUNT command or to the tape label parameter. The example shows the qualifier added to the parameter.

```
T1      PROCESS      SNODE=SC.VMS.JOEUSER  SNODEID=(JOEUSER,USER_PASSWORD)
STEP01  COPY  FROM  (DSN=DUXX:<DIRECTORY>TESTFILE.DAT)                    -
              TO    (DSN=MUXX:TESTFILE.DAT                                -
                     SYSOPTS="MOUNT='MUXX: TAPE /OVERRIDE=ID' ")
```

Copy a File from Tape to Disk (OpenVMS)

The following example copies an OpenVMS file from tape to disk. Specifying the /OVERRIDE qualifier causes the name of the tape volume to be ignored.

```
T2      PROCESS      SNODE=SC.VMS.JOEUSER SNODEID=(JOEUSER,USER_PASSWORD)
STEP01  COPY  FROM  (DSN=MUXX:TESTFILE.DAT                               -
                     SYSOPTS="MOUNT='MUXX: TAPE /OVERRIDE=ID' ")         -
              TO    (DSN=DUXX:<DIRECTORY>TESTFILE.DAT)
```

Copy from z/OS to OpenVMS and Specifying a User-Defined Translation Table

This Process, submitted from the OpenVMS node, copies from a Sterling Connect:Direct for z/OS node to a Sterling Connect:Direct for OpenVMS node. It illustrates how a user can specify a user-defined translation table. Using the

XLATE keyword in a Process overrides the default translation table; the Sterling Connect:Direct system will extract a module from the OpenVMS text library CD_1.TBL.

```
RECV_VB  PROCESS      SNODE=CD.OS390.NODE
STEP01   COPY  FROM  (DSN=SMITH.CDTEST SNODE                -
                      DISP=SHR )                            -
               TO    (DSN=$DISK:[JONES.DATA]ACCT.TXT         -
                      SYSOPTS="NOBINARY                     -
                      XLATE='$DSK:[JS.TXT]CD_1.TBL{JS.DEF}'  -
                      PROTECTION='S:RW,W:RWED'              -
                      LIBRARY='TEXT' STREAM" PNODE)
```

Copy a Single Entry from the OpenVMS Text Library to a z/OS Member

This Process sends a single entry from an OpenVMS text library to a member of a PDS on z/OS.

```
SINGENT1  PROCESS      SNODE=CD.OS390.NODE
STEP01    COPY  FROM  (DSN=DUA0:[SMITH.CDTEST]PDSTEST.TLB PNODE  -
                       SELECT=(BOOLEAN)                         -
                       SYSOPTS="LIBRARY='TEXT'"                 -
                       DISP=OLD                                 -
                       DCB=(DSORG=PO))                          -
                TO    (DSN=SMITH.MEMBERS(DATA)                 -
                       DISP=RPL SNODE)
```

Copy All Entries from an OpenVMS Text Library to z/OS

This Process copies all the modules of a text library to respective members of a PDS on z/OS. Note the use of the asterisk (*) with the SELECT parameter.

```
ALLENT1  PROCESS      SNODE=CD.OS390.NODE
STEP01   COPY  FROM  (DSN=DUA0:[JSMITH.CDTEST]PDSTEST.TLB PNODE  -
                      SELECT=(*)                                -
                      SYSOPTS="LIBRARY='TEXT'"                  -
                      DISP=OLD                                  -
                      DCB=(DSORG=PO))                           -
               TO    (DSN=JSMITH.MEMBERS                       -
                      DISP=RPL SNODE)
```

Copy a Data Set from a z/OS Node to an Executable File on an OpenVMS Node

This Process, submitted from the OpenVMS node, copies the z/OS data set RSMITH.ACCTJAN to file specification DUC4:[ACCT.COM]JAN.EXE at the OpenVMS node. For the appropriate file attribute information, the SYSOPTS parameter TYPE=IMAGE must be specified. Also, BINARY must be specified as part of the SYSOPTS parameter so that EBCDIC to ASCII translation will not occur.

```
PROCESS1  PROCESS      SNODE=CD.OS390                    -
                       SNODEID=(RSMITH,ROGER)
STEP01    COPY  FROM  (DSN=RSMITH.ACCTJAN SNODE  )       -
                TO    (DSN=DUC4:[ACCT.COM]JAN.EXE PNODE  -
                       SYSOPTS="TYPE='IMAGE' BINARY"     -
                       DISP=RPL)
```

Copy an Executable File from an OpenVMS Node to a z/OS Node

This Process, submitted from the OpenVMS node, copies the OpenVMS file JAN.EXE in directory [ACCT.COM] on device DUC4 to data set RSMITH.ACCTJAN at the z/OS node. BINARY must be specified as part of the SYSOPTS parameter so that ASCII to EBCDIC translation will not occur.

```
PROCESS1  PROCESS      SNODE=CD.OS390                    -
                       SNODEID=(RSMITH,ROGER)
STEP01    COPY  FROM  (DSN=DUC4:[ACCT.COM]JAN.EXE PNODE  -
                       SYSOPTS="BINARY")                 -
                TO    (DSN=RSMITH.ACCTJAN               -
                       DISP=RPL SNODE)
```

To submit this Process from the z/OS node (the PNODE is a z/OS node), the following syntax changes must be made:

- Enclose the OpenVMS file specification between single or double quotation marks to allow special characters to be passed to the OpenVMS node.
- Change the brackets ([ ]) to less than and greater than signs (< >).

The modified Process is as follows:

```
PROCESS1  PROCESS      SNODE=CD.VMS                          -
                       SNODEID=(RSMITH,ROGER)
STEP01    COPY  FROM  (DSN='DUC4:<ACCT.COM>JAN.EXE' SNODE  -
                       SYSOPTS="BINARY")                   -
                TO    (DSN=RSMITH.ACCTJAN                 -
                       DISP=RPL PNODE)
```

Copy a Text File from OpenVMS to Microsoft Windows and Back to OpenVMS

In this example, a text file (TIME_TEST.TXT) is being copied from the OpenVMS platform to Microsoft Windows and being renamed VMS_TEST.TXT. In the next COPY statement, the VMS_TEST.TXT file is copied from the Microsoft Windows platform back to OpenVMS and being renamed WIN_TIME_TEST.TXT.

```
WINTXT   PROCESS      SNODE=W2S.4200.CDWOPS7  snodeid=(userid,pwd)
VMS2WIN  COPY  FROM  (DSN=DISK$AXP:[NDM_3400.INPUT]TIME_TEST.TXT DISP=SHR pnode)      -
               TO    (DSN="C:\output\vms_test.txt" DISP=RPL snode)
WIN2VMS  COPY  FROM  (DSN="C:\output\vms_test.txt" DISP=SHR snode)                    -
               TO    (DSN=DISK$AXP:[NDM_3400.OUTPUT]WIN_TIME_TEST.TXT  DISP=RPL pnode)
```

Copy between the z/OS and OpenVMS Platforms

In this example, the PNODE resides on an OpenVMS system while the SNODE is on a z/OS system. This is a two-step Process where in step 1 a file (CSDQA1.TESTFILE.BENCH.M100) is being copied from the SNODE to a file on the PNODE and being renamed test100.dat. In step 2 the OpenVMS file, test100.dat, is copied to a new z/OS dataset called CSDQA2.TEST.MB100. The SUB1 statement submits a Process that resides on the z/OS system.

```
vms2mvsr process      snode=Q1B.ZOS.V4600 SNODEID=(JWARD1,APR1951)              -
                      pnodeid=(jward1,ward1949) pnode=q3a.alpha.v3400
*
*  test copy, ckpt, comp and sub job
Step1    copy  from  (file=CSDQA1.TESTFILE.BENCH.M100 DISP=(OLD)  snode)       -
                      ckpt = 10M compress                                      -
               to    ( pnode file=disk$data:[qaalpha.q3a]test100.dat DISP=(RPL) -
                      sysopts="nobinary")
Step2    copy  from  (file=disk$data:[qaalpha.q3a]test100.dat disp=(SHR))      -
                      ckpt = 20M compress                                      -
               to    (file=CSDQA2.TEST.MB100 snode                            -
                      disp=(RPL,CATLG,DELETE)                                  -
                      dcb=(RECFM=FB,LRECL=80,BLKSIZE=6160)                     -
                      space=(cyl,(20,10),rlse))
SUB1     SUBMIT       DSN='CSDQA1.MANUAL.PROCESS(MVS2VMSR)'                    -
                      SUBNODE=SNODE CASE=YES
```

Copy and Compare Files on OpenVMS

In this example Processes, two files are being copied and then compared. If the two files check out to be the same, the user is notified that the compare was good; if not, the user is notified that the compare failed.

```
BENM1SEC  PROCESS      SNODE=SHEMPHILL-DT SNODEID=(SHEMPHILL,cayanne)
itP2S     COPY  FROM  (DSN=DISK$DATA:[QADATA.INPUT]BENCHM5.dat  pnode)      -
                       compress                                             -
                       ckpt=1M                                              -
                TO    (file=c:\BENCHM1SEC.TXT snode)
itS2P     COPY  FROM  (file=c:\BENCHM1SEC.TXT snode)                        -
                TO    (DSN=DISK$DATA:[QADATA.TEMP]BENCHM5.dat   pnode)     -
                       compress                                             -
                       ckpt=1M
CMPR      RUN TASK    (pgm=vms) pnode                                       -
                       sysopts = "output='DISK$DATA:[qadata.logs]diff.log'  -
                       cmd = 'DIFF DISK$DATA:[qadata.input]BENCHM5.dat      -
                       DISK$DATA:[qadata.temp]BENCHM5.dat'"
RCZERO    IF (CMPR = 0) then
                       run task (pgm=vms) pnode sysopts = "                 -
                       cmd = 'reply/user=SHEMP1 BENCHM1.dat-GOOD'"
                else
                       run task (pgm=vms) pnode sysopts = "                 -
                       cmd = 'reply/user=SHEMP1 BENCHM1.dat-COMPARE_FAILED'"
          EIF
```

Copy HFS and Text Files Back and Forth Between z/OS and OpenVMS

In this example. the PNODE is OpenVMS while the SNODE is z/OS. For each node, userids and passwords have been included. Step up1 copies a HFS text file from the z/OS USS to a text file on the OpenVMS system called hfsout.txt. The OpenVMS directory is disk$data:[qaalpha.q3a]. Step back1 is just the opposite but it names the HFS file unix2.txt. The sysopts="permission" gives read write authority to HFS files. The sysopts="binary" indicates the data is not to be converted from ASCII to EBCDIC.]

```
VMS2HFS  PROCESS      SNODE=Q1B.ZOS.V4600 SNODEID=(JWARD1,MAY1975)     -
                      PNODE=Q3A.ALPHA.V3400 PNODEID=(jward1,ward1949)
up1      COPY                                                         -
               TO    ( DSN=DISK$DATA:[qaalpha.q3a]hfsout.txt disp=rpl  -
                       SYSOPTS="binary" )                             -
               FROM  (file='/u/jward1/unix.txt' disp=shr SNODE        -
                       SYSOPTS="PERMISS=777")
back1    COPY  TO    (file='/u/jward1/unix2.txt' disp=rpl SNODE       -
                       SYSOPTS="PERMISS=777")                         -
                       FROM  ( DSN=DISK$DATA:[qaalpha.q3a]hfsout.txt  -
                       DISP=SHR                                       -
                       SYSOPTS="binary")
```

Copy a Text File from OpenVMS to UNIX and Back to OpenVMS

In this example, a text file (TIME_TEST.TXT) is being copied from the OpenVMS platform to UNIX. In the next COPY statement, the TIME_TEST.TXT file is copied from the UNIX platform back to OpenVMS and being renamed UX_TIME_TEST.TXT.

```
UXTXT   PROCESS       SNODE=Qasol103700 Snodeid(qatest,qatest)
VMS2UX  COPY  FROM  ( DSN=DISK$AXP:[NDM_3400.INPUT]TIME_TEST.TXT DISP=SHR pnode )     -
              TO    ( DSN="/tmp/TIME_TEST.TXT" DISP=RPL snode )
UX2VMS  COPY  FROM  ( DSN="/tmp/TIME_TEST.TXT" DISP=SHR snode )                       -
              TO    ( DSN=DISK$AXP:[NDM_3400.OUTPUT]UX_TIME_TEST.TXT DISP=RPL pnode )
```

Copy a FB Text File to a z/OS PDS File and Pull Back to OpenVMS

In this example, a text file with fixed-length records on an OpenVMS node is being copied to a z/OS PDS file and then copied from the z/OS secondary node and pulled back to the OpenVMS primary node as a fixed-block text file.

```
ZOSPDS1  PROCESS      SNODE=Q1B.ZOS.V4600                                    -
                      SNODEID=(USERID,PWD)
STEP01   COPY  FROM  (DSN=DISK$AXP:[NDM_3400.INPUT]A80_80FB.dat DISP=SHR PNODE)  -
               TO    (DSN=CSDQA1.O.TESTFILE.PDS(a8080) DISP=RPL SNODE)
STEP01   COPY  FROM  (DSN=CSDQA1.O.TESTFILE.PDS(a8080) DISP=SHR SNODE)        -
               TO    (DSN=DISK$AXP:[NDM_3400.OUTPUT]zos_80.dat                -
                      DCB=(DSORG=PS,RECFM=FB)                                 -
                      DISP=RPL  PNODE )
```

Copy a Binary File from OpenVMS to Microsoft Windows and Back

In this example, a binary file (TEST.EXE) is being copied from the OpenVMS platform to Microsoft Windows. In the next COPY statement, the TEST.TXT file is copied from the Microsoft Windows platform back to OpenVMS.

```
WINBIN   PROCESS       SNODE=W2S.4200.CDWOPS8 snodeid=(userid,pswd)
VMS2WIN  COPY  FROM  ( DSN=DISK$AXP:[NDM_3400.INPUT]TEST.EXE DISP=SHR PNODE )  -
               TO    ( DSN="C:\output est.exe"                                 -
                       SYSOPTS="datatype(binary)" DISP=RPL snode )
WIN2VMS  COPY  FROM  ( DSN="C:\output est.exe"                                 -
                       SYSOPTS="datatype(binary)" DISP=SHR snode )             -
               TO    ( DSN=DISK$AXP:[NDM_3400.OUTPUT]test.exe                  -
                       DCB=(DSORG=PS,RECFM=FB,LRECL=512)  DISP=RPL  PNODE )
```

Copy a Binary File from OpenVMS to UNIX and Back

In this example, a binary file (TEST.EXE) is being copied from the OpenVMS platform to UNIX. In the next COPY statement, the TEST.TXT file is copied from the UNIX platform back to OpenVMS (ux_test.exe).

```
UXBIN   PROCESS      SNODE=qasol103700 Snodeid(userid,pwd)
VMS2UX  COPY  FROM  (DSN=DISK$AXP:[NDM_3400.INPUT]TEST.EXE DISP=SHR PNODE)           -
              TO    (DSN="/tmp/test.exe" SYSOPTS=":datatype=binary:" DISP=RPL SNODE)
UX2VMS  COPY  FROM  (DSN="/tmp/test.exe" SYSOPTS=":datatype=binary:" DISP=SHR SNODE)  -
              TO    (DSN=DISK$AXP:[NDM_3400.OUTPUT]ux_test.exe                       -
                     DCB=(DSORG=PS,RECFM=FB,LRECL=512) DISP=RPL PNODE)
```

Copy an OpenVMS Key-Sequenced File to an HP NonStop Node

This Process, submitted from the OpenVMS node, will copy a key-sequenced file to the Sterling Connect:Direct for HP NonStop node. When copying files from Sterling Connect:Direct for OpenVMS to Sterling Connect:Direct for HP NonStop nodes, include SET XLATE ON in the SYSOPTS parameter of the TO clause of the COPY statement. Because the Sterling Connect:Direct for OpenVMS system translates ASCII characters to EBCDIC, the XLATE subparameter will turn on the text conversion utility and translate from EBCDIC to ASCII. The FASTLOAD option is used to reduce disk I/O overhead.

```
VAXSND  PROCESS     PNODE=CD.VMS                                  -
                    SNODE=CD.HPNONSTOP SNODEID=(GRP.USR,PASWRD)
STEP1   COPY  FROM  (DSN=[USER.DATA]KSDS1.DAT)                    -
              TO    (DSN=$B.DATA.KSDS                             -
                     DCB=(DSORG=K                                 -
                     BLKSIZE=4096                                 -
                     LRECL=100                                    -
                     KEYLEN=10)                                   -
                     DISP=RPL                                     -
                     SYSOPTS="SET XLATE ON FAST.LOAD Y")
```

Copy an OpenVMS File from Disk to Tape

The following example copies an OpenVMS file from disk to tape. Specifying the /OVERRIDE qualifier causes the name of the tape volume to be ignored. The /OVERRIDE qualifier can be added either to the MOUNT command or to the tape label parameter. The example shows the qualifier added to the parameter.

```
T1      PROCESS      SNODE=SC.VMS.JOEUSER  SNODEID=(JOEUSER,USER_PASSWORD)
STEP01  COPY  FROM  (FILE=DUXX:[DIRECTORY]TESTFILE.DAT)                  -
              TO    (FILE=MUXX:TESTFILE.DAT                              -
                     SYSOPTS="MOUNT='MUXX: TAPE /OVERRIDE=ID' ")
```

Copy an OpenVMS File from Tape to Disk

The following example copies an OpenVMS file from tape to disk. Specifying the /OVERRIDE qualifier causes the name of the tape volume to be ignored.

```
T2      PROCESS      SNODE=SC.VMS.JOEUSER  SNODEID=(JOEUSER,USER_PASSWORD)
STEP01  COPY  FROM  (FILE=MUXX:TESTFILE.DAT                              -
                     SYSOPTS="MOUNT='MUXX: TAPE /OVERRIDE=ID' ")         -
              TO    (FILE=DUXX:[DIRECTORY]TESTFILE.DAT)
```

Use Symbolics in an OpenVMS COPY Statement

This example shows the basic use of symbolics in a Process. Both the FROM and TO files, as well as the file disposition, are resolved at Process submission.

```
SYMBOL1  PROCESS      SNODE=REMOTE_NODE
STEP01   COPY  FROM  (FILE=&FROM PNODE)    -
               TO    (FILE=&TO DISP=&DISP)
```

The Process, SYMBOL1, can be submitted with the following command issued in DCL command format:

```
$ NDMUI SUBMIT SYMBOL1               -
/SYMBOLICS=("FROM=VMS_FILENAME.TYPE", -
            "TO=OS390.DATASET.NAME",  -
           "DISP=RPL")
```

Copy an OpenVMS Sequential File to a Text Library

This example shows the format for copying a sequential file to a text library.

```
COPY01  PROCESS      SNODE=CD.VMS.DATAMOVER
DO_A    COPY  FROM  (FILE=VMSFILE)                    -
              TO    (FILE=VMSLIBRARY.TLB(VMSFILE)      -
                     SYSOPTS="LIBRARY='TEXT' REPLACE"  -
                     DISP=RPL                          -
                     DCB=(DSORG=PO))
```

Copy a File from OpenVMS to z/OS and Back to OpenVMS

These example COPY statements (STEP01 and STEP02) copy an executable file from a Sterling Connect:Direct for OpenVMS node to a Sterling Connect:Direct for z/OS node and then back to the Sterling Connect:Direct for OpenVMS node. Because BINARY is specified as part of the SYSOPTS parameter, ASCII-to-EBCDIC translation does not occur. Enclose the SYSOPTS string in double quotation marks.

```
STEP01     COPY FROM    (PNODE DSN='$DISK1:[USER.FILE]TEST.EXE'
                        SYSOPTS="BINARY"
                        TYPE=IMAGE
                        DISP=SHR)
                TO      (SNODE DSN=OS390.FILE
                        DISP=RPL)
STEP02     COPY FROM    (SNODE DSN=OS390.FILE
                        DISP=SHR)
                TO      (PNODE DSN='$DISK1:[USER.FILE]TEST2.EXE'
                        TYPE=IMAGE
                        SYSOPTS="BINARY"
                        DISP=RPL)
```

UNIX

Copy Files Between UNIX and z/OS

This Process copies a file from UNIX to z/OS. The Process was initiated from the UNIX node. The **ckpt** parameter specifies that no checkpoints will be taken. The **ckpt** parameter is generally coded between the FROM and TO clauses of the COPY statement.

```
OS390xx  process      snode=OS390.node
step01   copy  from  (file=file1
                      pnode)
                      ckpt=no
               to    (file=file2
                      dcb=(dsorg=PS,
                      recfm=FB,
                      lrecl=80,
                      blksize=2400)
                      space=(TRK,(1,,))
                      snode
                      disp=rpl)
         pend
```

This Process, submitted from the z/OS node, copies a text file from UNIX to z/OS. The DSN and SYSOPTS strings for the UNIX system must be in the proper case for UNIX and enclosed in double quotation marks.

```
PROC2   PROCESS      SNODE=UNIX.NODE
STEP01  COPY  FROM  (DSN="/company/payroll/monthly/jan"           -
                     SYSOPTS=":datatype=text:"                    -
                     SNODE)                                       -
              TO    (DSN=OS390.PAYROLL.MONTHLY.JANUARY           -
                     DISP=(NEW,CATLG)                             -
                     SPACE=(23040,(2,1))                          -
                     DCB=(RECFM=U,LRECL=0,BLKSIZE=23040,DSORG=PS) -
                     PNODE)
```

Copy Files and Using sysopts (UNIX to UNIX)

This Process copies a file between two UNIX nodes. The **sysopts** parameter specifies to remove trailing blank characters from a line of text before writing it to a text file. The **sysopts** subparameters are a series of field names and values, each of which is delimited by a colon and enclosed in double quotation marks.

```
strip  process    snode=unix.node
step01 copy from (file=blank.dat
                  sysopts=":datatype=text:"
                  snode)
            to   (file=blank_no
                  sysopts=":datatype=text:strip.blanks=yes:"
                  pnode)
 pend
```

Copy Files and Use the Checkpointing Feature (UNIX to UNIX)

This Process copies a file between two UNIX nodes. The **ckpt** parameter specifies that checkpoints will be taken at 128K intervals. If the COPY operation is interrupted, the Sterling Connect:Direct system will restart that COPY step at the last checkpoint. Code the **ckpt** parameter between the FROM and TO clauses of the COPY statement.

```
ckpt01  process     snode=unix.node
step01  copy from  (file=file1
                    snode)
                    ckpt=128k
             to    (file=file2
                    disp=new
                    pnode)
 pend
```

Copy Files and Use the Compression Feature (UNIX to UNIX)

This Process shows the syntax of the **compress** parameter. The **compress** parameter specifies that data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file is automatically decompressed at the destination. Code the **compress** parameter between the FROM and TO clauses of the COPY statement.

Compression activities for each step are as follows:

- Step01 specifies use of hex **20**, the default, as a compression character.
- Step02 specifies use of character **1** as a compression character.
- Step03 specifies use of hex **11** as a compression character.
- Step04 specifies use of the extended compression method. CMP specifies the compression level. WIN specifies the window size. MEM specifies the memory level.

Use this Process when you copy files from UNIX to UNIX using the compress parameter.

```
comp    process      snode=unix.node
                     snodeid=(userid,passwrd)
step01  copy  from  (file=text2 snode)
                     compress
              to    (file=file3 pnode)
step02  copy  from  (file=text2 snode)
                     compress primechar=c'1'
              to    (file=file3 pnode)
step03  copy  from  (file=text2 snode)
                     compress primechar=x'11'
              to    (file=file3 pnode)
step04  copy  from  (file=text2 snode)
                     compress extended [=(CMP=1
                     WIN=9
                     MEM=1
                    )
                    ]
              to    (file=file3 pnode)
 pend
```

Archive Files Using the Sterling Connect:Direct for UNIX Pipe I/O Function

This Process changes the **pnode** directory to the se subdirectory, archives all the *.c files in the **se** subdirectory using the **tar** command, and then transfers the archive to the **snode**. No checkpointing occurs when **pipe=yes** is specified.

```
pipe_ex1  process      snode=testcdu
step01    copy  from  (file="cd se; tar -cf - *.c"
                       pnode sysopts=":pipe=yes:")
                to    (file=unix.se.tar snode disp = rpl)
 pend
```

Restore Files Using the Sterling Connect:Direct for UNIX Pipe I/O Function

This Process copies a **tar** archive from the **snode** and extracts files from the archive. No checkpointing occurs when **pipe=yes** is specified.

```
pipe_ex2  process      snode=testcdu
step01    copy  from  (file=unix.se.tar snode)
                to    (file="cd se; tar -xf -" pnode sysopts=":pipe=yes:")
 pend
```

Archive and Restore Files in a Single Step Using the Sterling Connect:Direct for UNIX Pipe I/O Function

This Process changes the **pnode** directory to the se subdirectory, archives all the *.c files in the **se** subdirectory using the **tar** command, then transfers the archive to the **snode**. At the **snode**, this Process changes the directory to the **testdir** subdirectory and extracts the *.c files from the archive using the **tar -xf** command. No checkpointing occurs when **pipe=yes** is specified.

```
pipe_ex3  process      snode=testcdu
step01    copy  from  (file="cd se; tar -cf - *.c"
                       pnode sysopts=":pipe=yes:")
                to    (file="cd testdir; tar -xf -"
                       snode sysopts=":pipe=yes:")
 pend
```

Copy Files from UNIX to a Member on i5/OS

This Process copies an ASCII file from UNIX to a member on the i5/OS.

```
* COPY TO MEMBER *
copy01  process      snode=os400
                     snodeid=(userid,passwrd)
step01  copy  from  (file=/cd/file1
                     pnode
                     sysopts=":datatype=text:xlate=yes:")
              to    (file="LIB/FILENAME(MBR_NAME)"
                     sysopts="TYPE( MBR )"
                     disp=rpl)
  pend
```

Copy Files from UNIX to a Spool File on i5/OS

This Process copies an ASCII file from UNIX to a spool file on the i5/OS.

See the *IBM Sterling Connect:Direct for i5/OS User's Guide* for the spool file parameters.

```
* COPY TO SPOOL FILE *
copy01  process      snode=os400
                     snodeid=(userid,passwrd)
step01  copy  from  (file=/cd/file1
                     pnode
                     sysopts=":datatype=text:xlate=yes:")
              to    (file=FILE2
                     snode
                     sysopts="TYPE( SPLF ) PRTQLTY( *NLQ )"
                     disp=rpl)
  pend
```

Copy Save Files from i5/OS to UNIX

This Process copies a save file from i5/OS to UNIX.

```
* COPY SPECIFYING DCB INFORMATION *
copy01   process      snode=os400
                      snodeid=(userid,passwrd)
step01A  copy
               from  (file="URGRSSSV1/SAVEFILE1"
                      snode
                      sysopts="TYPE(OBJ)")
                      compress
               to    (file=/cd/usavefile1
                      sysopts=":datatype=binary:permiss=774:"
                      pnode
                      disp=new)
  pend
```

Copy Save Files from UNIX to i5/OS

This Process copies a save file from UNIX to i5/OS. Set **datatype=binary** for save files. Specify DCB information to copy a save file to i5/OS.

```
* COPY SPECIFYING DCB INFORMATION *
copy02   process      snode=os400
                      snodeid=(userid,passwrd)
step02B  copy
               from  (file=/cd/usavefile1
                      sysopts=":datatype=binary:permiss=774:"
                      pnode)
                      compress
               to    (file="URGRSSSV1/SAVEFILE1"
                      snode
                      sysopts="TYPE(OBJ) MAXRCDS(*NOMAX)"
                      disp=new)
  pend
```

Copy Executables from UNIX to i5/OS

This Process copies an executable from UNIX to i5/OS. Specify FILETYPE(*DATA) in the **sysopts** parameter.

```
* COPY UNIX EXECUTABLE TO i5/OS *
copy01  process      snode=os400
                     snodeid=(userid,passwrd)
step01  copy  from  (file=/cd/xdt3
                     sysopts=":datatype=binary:permiss=777:"
                     pnode)
              to    (snode
                     file="CD/BINARY(UDESKTOP)"
                     sysopts="TYPE(MBR) FILETYPE(*DATA)"
                     disp=new)
  pend
```

Copy a File from UNIX to Microsoft Windows

This Process copies a binary file from a UNIX node to a Microsoft Windows node. A TCP/IP address is specified instead of the Sterling Connect:Direct node name for the SNODE.

```
ux2nt   process      snode=111.11.11.111
                     hold=no
                     retain=no
copy1   copy  from  (file=/usr/data/out/invoi01.dat
                     pnode
                     sysopts=":datatype=binary:" )
              to    (file=d:\users\data\in\invoi01.dat
                     snode
                     sysopts="datatype(binary)" )
 pend
```

The following Process is a variation on the previous example. In this example, the file names are defined as symbolic variables in the COPY statement (&file1 and &file2) and are resolved at the time the Process is submitted.

```
proc1  process      snode=111.11.11.111
                    &file1="/usr/data/out/invoi01.dat"
                    &file2="d:\users\data\in\invoi01.dat"
copy1  copy  from  (file=&file1
                    pnode
                    sysopts=":datatype=binary:")
             to    (file=&file2
                    sysopts="datatype(binary)"
                    snode)
  pend
```

Copy a File from a UNIX Node to a z/OS Node

In this example, the Process copies a binary file from a local UNIX node to a z/OS node:

```
copyseq  process      snode=dallas
/* When copying, make sure the datatype is set to binary. */
step01   copy  from  (file=a.out
                      pnode
                      sysopts=":datatype=binary:")
                      ckpt=64k
                      compress extended= (CMP=1
                                          WIN=9
                                          MEM=1)
              to    (file=TESTAOUT
                      snode
                      disp=(rpl))
/*  If step01 succeeds, CD will copy the same file back to UNIX.  */
step02   if  (step01 > 4 ) then
              exit
         eif
/*  Before copying the file back, delete it first.  */
step03   run task
                      pnode
                      sysopts="rm -f a.out"
step04   if  (step03 > 4 ) then
              exit
         eif
/*  Copy the file from OS390 to UNIX.  */
/*  When copying, make sure the datatype is set to binary.  */
step05   copy  from  (file=TESTAOUT
                      snode)
                      ckpt=64k
              to    (file=a.out
                      pnode
                      disp=(rpl)
                      sysopts=":datatype=binary:")
 pend
```

Wildcard Copies from UNIX to Microsoft Windows

In the following example, a Sterling Connect:Direct for UNIX PNODE directory /financial/accounts contains the files customer1, customer2, customer3, supplier1, and supplier2. A Sterling Connect:Direct for Microsoft Windows SNODE has the directory C:/financial/accounts. The following wildcard copy command copies the files called customer1, customer2, and customer3 from the PNODE to the C:/financial/accounts directory on the SNODE. The source file names and the destination file names are identical.

```
WILDCOPY  COPY
               FROM  (FILE=/financial/accounts/customer?)
               TO    (FILE=C:\financial\accounts\
                      DISP=RPL)
```

You must include the ending backslash (\) for the destination directory.

The following wildcard copy step copies customer1, customer2, customer3, supplier1, and supplier2 into the C:\financial\accounts directory on the SNODE. The source file names and the destination file names are identical.

```
WILDCOPY  COPY
               FROM  (FILE=/financial/accounts/*)
               TO    (FILE=J:\financial\accounts\
                      DISP=RPL)
```

Wildcard Copy from UNIX to UNIX

To copy send to a Sterling Connect:Direct for UNIX node, you must include an ending forward slash (/) in the **TO FILE=** parameter. Following is an example:

```
WILDCOPY  COPY
                FROM  (FILE="/financial/accounts/customer?")
                TO    (FILE=/financial/accounts/
                       DISP=RPL)
```

Wildcard Copy from UNIX to a z/OS Node

To copy send to sequential files on a Sterling Connect:Direct for z/OS node, you must include an ending period (.) in the **TO FILE=** parameter. Following is an example:

```
WILDCOPY  COPY
                FROM  (FILE=/financial/accounts/*)
                TO    (FILE=FINANCIAL.ACCOUNTS.
                       DISP=RPL)
```

To copy send to a PDS on a Sterling Connect:Direct for z/OS node, you must use an asterisk (*) for the PDS member name. Following is an example:

```
WILDCOPY  COPY
                FROM  (FILE=/financial/records/*)
                TO    (FILE=FINANCIAL.RECORDS(*)
                       DISP=RPL)
```

Wildcard Copy from UNIX to a Node with Download Restrictions

To copy send to a Sterling Connect:Direct node that enforces download restrictions, use an asterisk (*) for the **TO FILE=** parameter if the destination directory is the download directory. Following is an example:

```
WILDCOPY COPY
      FROM (FILE=/financial/records/*)
      TO   (FILE=*
            DISP=RPL)
```

VM/ESA

Copy a VSAM File from VM to an Entry-Sequenced HP NonStop File

The following Process copies a VM VSAM file to an entry-sequenced HP NonStop file. The entry-sequenced file with extents of 100 pages each is created as indicated by the SYSOPTS parameter. Note that the VM file name is not enclosed in single quotation marks. If the VM file name is not placed between quotation marks, the Sterling Connect:Direct system assumes the file is a VSAM file.

```
HPNONSTP  PROCESS      PNODE=CD.VM.DALLAS HOLD=YES       -
                       SNODE=BOSTON.01 NOTIFY=%USER      -
                       SNODEID=(127.200,JONES)
SEND01    COPY  FROM  (DSN=ABC.RPTFILE                  -
                       LINK=(IVVB6,RIVVB6,RR,200))       -
                TO    (DSN=$C.JONES.VSAME SNODE         -
                       SYSOPTS=\"'SET EXT(100 100)'\ ||  -
                               \ 'SET TYPE E'"\          -
                       DISP=(RPL))
```

Copy a VSAM File from VM to a Key-Sequenced HP NonStop File

The following Process copies a VM VSAM file to a key-sequenced HP NonStop file. The key-sequenced file with extents of 100 pages each is created as indicated by the SYSOPTS parameter. Because the VM file name is not placed between quotation marks, the Sterling Connect:Direct system assumes the file is a VSAM file.

```
HPNNSTP6  PROCESS    PNODE=CD.VM.DALLAS                -
                     SNODE=BOSTON.01 NOTIFY=%USER      -
                     SNODEID=(127.210,SMITH)
SEND01  COPY  FROM  (DSN=ABC.TST                       -
                     LINK=(IVVB6,RIVVB6,RR,200))       -
              TO    (DSN=$C.ABC.VSAMERR SNODE         -
                     SYSOPTS=\"'SET EXT(100 100)'\ ||  -
                             \ 'SET KEYLEN 8'\ ||      -
                             \ 'SET REC 880'\ ||       -
                             \ 'SET TYPE K'"\          -
                     DISP=(RPL))
```

Copy a VM File to VM Spool

This Process copies a file to the VM reader of user RJONES. Because a copy to VM spool does not involve writing to disk, you do not need to specify link information.

```
VM2RDR2  PROCESS      SNODE=CD.VM.JSMITH NOTIFY=RJONES
STEP01   COPY  FROM  (DSN='JIM SCRIPT'                      -
                      LINK=(RRT,READPW,RR,191))             -
                TO    (DSN='!SPOOL RJONES TEST DATA'        -
                       DCB=(DSORG=PS,RECFM=F,LRECL=80,BLKSIZE=80))
```

Copy an Entire VM Minidisk

This example shows the COPY statement of a Process that copies an entire minidisk to another minidisk (301).

```
STEP01  COPY  FROM  (GROUP='* *'                -
                     LINK=(MDSKI,RMDSKI,RR,199) -
                     DISP=SHR)                  -
              TO    (GROUP='%1% %2%'            -
                     LINK=(N4100,WN14100,W,301) -
                     DISP=RPL)
```

Copy from VM Disk to Tape

This Process copies a VM disk file from the 191 disk of IVVB8 to tape.

```
TAPE    PROCESS      SNODE=CD.VM.NODE           -
                     NOTIFY=CDA8
STEP01  COPY  FROM  (DSN='TEST FILE'            -
                     LINK=(IVVB8,RIVVB7,RR,191) -
                     DISP=SHR)                  -
              TO    (DSN=TEST.EXPDT.ONE        -
                     UNIT=TAPE                  -
                     LABEL=(1,SL,EXPDT=900967)  -
                     SNODE                      -
                     DISP=RPL)
```

VM to VM Group File Copy

This Process illustrates a VM group file copy, which copies source modules from one minidisk to a minidisk at another site. Notice that a source file name, as well as a group name, is specified on the FROM clause of the COPY statement. This causes the Sterling Connect:Direct for VM system to send members of the group beginning with that source file name instead of beginning with the first member of the group; therefore, members of a group are excluded from the transfer.

```
GROUP9  PROCESS      SNODE=CD.VM.NODE
STEP01  COPY  FROM  (DSN='ACF2C A*'             -
                     GROUP='* A*'               -
                     LINK=(IVVB7,RIVVB7,RR,191) -
                     DISP=SHR)                  -
              TO    (GROUP='%1% A%2%'          -
                     LINK=(IVVB6,WIVVB6,W,301)  -
                     DISP=RPL)
```

The parameter GROUP on the TO clause contains the special symbols %1% and %2%, which are used to build the destination name. Each symbol is replaced by characters from the name determined to be in that source group.

The source disk, CDA7 191, contains the following:

```
ACF2A ASSEMBLE    ARMMOD ASSEMBLE
ACF2B ASSEMBLE    DMCXRJ ASSEMBLE
ACF2C ASSEMBLE   DMDPTR ASSEMBLE
ALOEXIT ASSEMBLE    DMFPTR ASSEMBLE
ASMSAMP ASSEMBLE    DMGFTR ASSEMBLE
ASMTASK ASSEMBLE    DMMF ASSEMBLE
```

After the transfer completes, the CDA6 300 disk contains:

```
CF2C ASSEMBLE   DMCXRJ ASSEMBLE
ALOEXIT ASSEMBLE   DMDPTR ASSEMBLE
ASMSAMP ASSEMBLE   DMFPTR ASSEMBLE
ASMTASK ASSEMBLE   DMGFTR ASSEMBLE
ARMMOD ASSEMBLE   DMMF ASSEMBLE
```

ACF2A and ACF2B are excluded from the COPY because the DSN parameter indicated that the group file copy should start with the ACF2C A* file.

Copy VM files to a Shared File System (SFS)

This multi-step Process copies several different types of VM files to a SFS. In each step, if the file exists, Sterling Connect:Direct replaces it. If the file does not exist, the Sterling Connect:Direct system creates it as indicated by the DISP=RPL parameter. All of the files (input and output) are fixed length 80 byte records. Each step performs the following task:

- STEP1 copies a CMS file in a SFS to another SFS.
- STEP2 copies a CMS file from a Minidisk to a SFS.
- STEP3 copies a VSAM RRDS to a sequential file in a SFS.
- STEP4 copies a VSAM KSDS to a sequential file in a SFS.
- STEP5 copies a VSAM ESDS to a sequential file in a SFS.

```
SFSPROC  PROCESS                                                -
             &PROCESS=SFSPROC                                   -
             &CKPT=0K                                           -
             &COMPRESS=COMPRESS                                 -
             &EXT=,                                             -
             &CUU1=0199                                         -
             &CUU2=0195                                         -
             &DIR1='MYSFS:USER01.MYSFS'                         -
             &DIR2='COSFS:USER02.COSFS'                         -
             &INUSER=USER01                                     -
             &INUSERP=RPASS                                     -
             SNODEID=(USERID,PASSWD)                            -
             &SNODE=CD.VM.NODE1                                 -
             SNODE=&SNODE
STEP1    COPY  FROM   (PNODE                                    -
                       SFSDIR=("&DIR1")                         -
                       DSN='MYINPUT FILE'                       -
                       DISP=SHR )                               -
                       CKPT=&CKPT &COMPRESS  &EXT               -
                 TO    (SNODE                                   -
                       SFSDIR=("&DIR2")                         -
                       DSN=\xd5 FILETEST\&PROCESS.1\xd5 \       -
                       DCB=(LRECL=80,RECFM=F)                   -
                       DISP=RPL )
STEP2    COPY  FROM   (PNODE                                    -
                       LINK=(&INUSER,&INUSERP,RR,&CUU1)         -
                       DSN='MYINPUT FILE2'                      -
                       DCB=(LRECL=80,RECFM=F,DSORG=PS)          -
                       DISP=SHR )                               -
                       CKPT=&CKPT   &COMPRESS  &EXT             -
                  TO   (SNODE                                   -
                       SFSDIR=("&DIR2")                         -
                       DSN=\xd5 FILETEST\&PROCESS.2\xd5 \       -
                       DCB=(LRECL=80,RECFM=F)                   -
                       DISP=RPL  )
STEP3    COPY  FROM   (PNODE                                    -
                       LINK=(&INUSER,&INUSERP,RR,&CUU2)         -
                       DSN=MYHLQ.TESTFILE.VRRDS.FB80            -
                       DCB=(DSORG=VSAM)                         -
                       DISP=SHR )                               -
                       CKPT=&CKPT   &COMPRESS  &EXT             -
                 TO    (SNODE                                   -
                       SFSDIR=("&DIR2")                         -
                       DSN=\xd5 FILETEST\&PROCESS.3\xd5 \       -
                       DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)     -
                       DISP=RPL )
STEP4    COPY  FROM   (PNODE                                    -
                       LINK=(&INUSER,&INUSERP,RR,&CUU2)         -
                       DSN=MYHLQ.TESTFILE.VKSDS.FB80            -
                       DCB=(DSORG=VSAM)                         -
                       DISP=SHR )                               -
                       CKPT=&CKPT   &COMPRESS  &EXT             -
```

```
                TO      (SNODE                                          -
                        SFSDIR=("&DIR2")                                -
                        DSN=\xd5 FILETEST\&PROCESS.4\xd5 \               -
                        DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)            -
                        DISP=RPL  )
STEP5    COPY   FROM    (PNODE                                          -
                        LINK=(&INUSER,&INUSERP,RR,&CUU2)                -
                        DSN=MYHLQ.TESTFILE.VESDS.FB80                   -
                        DCB=(DSORG=VSAM)                                -
                        DISP=SHR  )                                     -
                        CKPT=&CKPT   &COMPRESS  &EXT                    -
                TO      (SNODE                                          -
                        SFSDIR=("&DIR2")                                -
                        DSN=\xd5 FILETEST\&PROCESS.5\xd5 \               -
                        DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)            -
                        DISP=RPL)
```

Extract an SFS File and Placing the File on the VM Reader Spool

In this example, one file is being extracted from the CDSFS filepool to be placed upon a VM reader spool. Note that the format of the SFSDIR statement must end with a period when only the main directory is referenced.

```
TESTSFS  PROCESS      SNODE=CSD.VM.NODE NOTIFY=USER1
STEP01   COPY  FROM  (DSN='PROFILE EXEC'              -
                      SFSDIR=('CDSFS:USER1.')         -
                TO    (DSN='!SPOOL USER1 TSET DATA')
```

Copy Files from VM to OpenVMS

This Process copies an existing VM file to an existing OpenVMS file.

```
VMSTEST  PROCESS      SNODEID=(RGL,UNISEF)
COPY01   COPY  FROM  (DSN='TEST FILE'             -
                      LINK=(SE9GWWT,TOM,RR,191)   -
                      DISP=SHR                    -
                      PNODE)                      -
                TO    (DSN='DISK:<RGL>VMTEST.DAT'  -
                      DISP=OLD)
```

Copy a VM Sequential File to a CA-DYNAM/T Tape File (VSE)

This Process copies a sequential file to a DYNAM/T volume tape file. Note that DCB information and the disposition of RPL are specified. If the VSE file does not exist, RPL specifies the Process is to allocate the file NEW using the DCB information.

```
AMFTAPE  PROCESS      SNODE=CD.VSE.NODE NOTIFY=%USER
STEP01   COPY  FROM  (DSN='TESTA INPUT'                          -
                      LINK=(IVB4100,WIVB4100,RR,202)             -
                      DISP=SHR)                                  -
                TO    (DSN='DYNAM.TAPE***'                       -
                      DCB=(DSORG=PS BLKSIZE=3200 LRECL=80 RECFM=F) -
                      DISP=RPL                                   -
                      UNIT=TNOASGN                               -
                      SNODE)
```

Copy VM Sequential Files to CA-DYNAM/D Files (VSE)

This multi-step Process shows various ways of copying sequential files to DYNAM/D.

- STEP01 and STEP02 copy to a DYNAM/D file that has not been defined to DYNAM/D.
- STEP03 copies to a DYNAM/D file using a TYPEKEY containing DCB information.
- STEP04 copies to an existing DYNAM/D file that has not been defined to DYNAM/D.
- STEP05 copies a sequential file to a DYNAM/D file (not previously defined). The DYNAM/D file has different DCB information specified.

```
NODEFINE  PROCESS      SNODE=CD.VSE.NODE
STEP01    COPY  FROM  (DSN='TESTA INPUT'                              -
                       LINK=(IVB4100,RIVB4100,RR,202)                 -
                       PNODE DISP=SHR)                                -
                TO    (DSN='VSE.NODEF.STEP01'                         -
                       DCB=(DSORG=PS BLKSIZE=3200 LRECL=80 RECFM=F)   -
                       UNIT=DNOASGN                                   -
                       VOL=SER=POOLNAME                               -
                       SNODE DISP=RPL)
STEP02    COPY  FROM  (DSN='CDFILE INPUT'                             -
                       LINK=(IVB4100,RIVB4100,RR,202)                 -
                       PNODE DISP=SHR)                                -
                TO    (DSN='VSE.NODEF.STEP02'                         -
                       DCB=(DSORG=PS BLKSIZE=3200 LRECL=80 RECFM=F)   -
                       UNIT=DNOASGN                                   -
                       VOL=SER=POOLNAME                               -
                       SNODE DISP=RPL)
STEP03    COPY  FROM  (DSN='TESTA INPUT'                              -
                       LINK=(IVB4100,RIVB4100,RR,202)                 -
                       PNODE DISP=SHR)                                -
                TO    (DSN='VSE.NODEF.STEP03'                         -
                       TYPE=DYNAMD                                    -
                       SNODE DISP=RPL)
STEP04    COPY  FROM  (DSN='TESTA INPUT'                              -
                       LINK=(IVB4100,RIVB4100,RR,202)                 -
                       PNODE DISP=SHR)                                -
                TO    (DSN='VSE.NODEF.STEP03'                         -
                       DCB=(DSORG=PS BLKSIZE=3200 LRECL=80 RECFM=F)   -
                       UNIT=DNOASGN                                   -
                       SNODE DISP=RPL)
STEP05    COPY  FROM  (DSN='TESTC INPUT'                              -
                       LINK=(SMI4100,RSMI4100,RR,202)                 -
                       PNODE DISP=SHR)                                -
                TO    (DSN='VSE.NODEF.STEP05'                         -
                       DCB=(DSORG=PS BLKSIZE=1000 LRECL=100 RECFM=F)  -
                       UNIT=DNOASGN                                   -
                       VOL=SER=POOLNAME                               -
                       SNODE DISP=RPL)
```

Copy Files from VM to i5/OS

This Process copies a sequential file from a Sterling Connect:Direct for VM node to an i5/OS node. The SYSOPTS parameter specifies that the data is to be copied to the Sterling Connect:Direct for i5/OS® node as a member of a physical database file. CKPT=0K turns off the checkpointing feature.

```
TEST      PROCESS      SNODE=OS400                                    -
                       SNODEID=(USER1,PASSWD1)
************************************************************
* STEP 1 WILL COPY SEQUENTIAL TO SEQUENTIAL ( VM TO OS400)
************************************************************
STEP1000  COPY  FROM  (PNODE                                         -
                       LINK=(QACD,RQACD,RR,191)                      -
                       DSN='TESTFILE'                                -
                       DCB=(LRECL=80,RECFM=F,DSORG=PS,BLKSIZE=80)    -
                       DISP=SHR                                      -
                       CKPT=0K                                       -
                TO    (SNODE                                         -
                       DSN='TEST/PDS(STEP1000)'                      -
                       SYSOPTS=\"\                                   -
                               \ TYPE ( MBR ) \                      -
                               \ MAXMBRS ( *NOMAX ) \                -
                               \ RCDLEN ( 92 ) \                     -
                               \ TEXT ( 'ADDED BY PROCESS TEST \     -
                               \ IN STEP1000' ) \                    -
                               \"\                                   -
                       DISP=RPL)
```

Copy VSAM Files from VM to i5/OS

This Process copies a VSAM file from a Sterling Connect:Direct for VM node to a sequential file on a Sterling Connect:Direct for i5/OS node. The DSN parameter on the COPY TO side specifies the destination object name based on the i5/OS standard file naming conventions and must be in single quotation marks. CKPT=0K turns off the checkpointing feature.

```
TEST1     PROCESS      SNODE=OS400                                   -
                       SNODEID=(USER1,PASSWD1)
*************************************************************
* STEP 2000 WILL COPY VSAM TO SEQUENTIAL  (VM TO OS400)
*************************************************************
STEP2000  COPY  FROM  (PNODE                                         -
                       LINK=(QACD,RQACD,RR,192)                      -
                       DSN=SCQA1.TESTFILE                            -
                       DCB=(DSORG=VSAM,LRECL=80)                     -
                       DISP=SHR)                                     -
                       CKPT=0K                                       -
                TO    (SNODE                                         -
                       DSN='TEST1/PDS(STEP2000)'                     -
                       SYSOPTS=\"\                                   -
                               \ TYPE ( FILE ) \                     -
                               \ MAXMBRS ( *NOMAX ) \                -
                               \ RCDLEN ( 92 ) \                     -
                               \ TEXT ( 'ADDED BY PROCESS TEST1 \    -
                               \IN STEP2000' ) \                     -
                               \"\                                   -
                       DISP=RPL)
```

VM-Initiated Copy from i5/OS to VM

This Process is initiated from the Sterling Connect:Direct for VM/ESA to copy a file from an i5/OS node to a VM node. The contents of the SYSOPTS parameter specifies that the object being copied is to be transferred in save object format.

```
TEST2     PROCESS      SNODE=OS400                                        -
                       SNODEID=(USER1,PASSWD1)
*********************************************************************
 * COPY PROCESS FROM OS400 TO VM (VM INITIATED)
 *********************************************************************
STEP4000  COPY  FROM  (SNODE                                             -
                       DSN=' TEST2  / SAVEFILE1 . FILE '                 -
                       SYSOPTS=\"\                                       -
                               \ TYPE ( OBJ ) \                          -
                               \"\)                                      -
                TO    (DSN='IAOB001 REGRESS1'                            -
                       LINK=(QACD,WQACD,W,192)                           -
                       DCB=(RECFM=FB,LRECL=528,BLKSIZE=5280,DSORG=PS)    -
                       DISP=NEW                                          -
                       PNODE)
```

VM-Initiated Copy VM to i5/OS Spool

This Process is initiated from the Sterling Connect:Direct for VM/ESA to copy a file from a VM node to a spooled output file on an i5/OS node.

```
TEST3     PROCESS      SNODE=OS400                    -
                       SNODEID=(USER1,PASSWD1)
*********************************************************************
* VM TO OS400 COPY OF A FILE TO OS400 SPOOL (VM INITIATED)
*********************************************************************
STEP5000  COPY  FROM  (PNODE                          -
                       LINK=(QACD,RQACD,RR,191)       -
                       DSN='OS400 REP1'               -
                       DISP=SHR)                      -
                TO    (SNODE                         -
                       DSN=TEST                       -
                       DISP=RPL                       -
                       SYSOPTS=\"\                    -
                               \ TYPE ( SPLF ) \      -
                               \ CTLCHAR ( *FCFC ) \  -
                               \ PRTQLTY( *NLQ ) \    -
                               \"\)
```

VM-Initiated Copy from i5/OS to VM Spool

This Process is initiated from the Sterling Connect:Direct for VM/ESA to copy a physical database file member from a Sterling Connect:Direct for i5/OS node to a VM spool file.

```
TEST3     PROCESS      SNODE=CD.DALLAS                                -
                       SNODEID=(USER1,PASSWD1)
*********************************************************************
* OS400 TO VM COPY OF A OS400 FILE TO VM SPOOL (VM INITIATED)
*********************************************************************
STEP6000  COPY  FROM  (SNODE                                         -
                       DSN=' TEST / CD01 (VMTEST) '                  -
                       SYSOPTS=\"\                                   -
                               \ TYPE ( MBR ) \                      -
                               \"\                                   -
                       DISP=SHR)                                     -
                TO    (DSN='!SPOOL  VMTEST TESTFILE'                -
                       DCB=(DSORG=PS,RECFM=F,LRECL=80,BLKSIZE=80))   -
                       PNODE)
```

Copy a VM VSAM file to z/OS

This Process copies a VSAM Disk from a VM/ESA node to a sequential disk file on a z/OS node. If the file exists, Sterling Connect:Direct replaces it. If the file does not exist, Sterling Connect:Direct creates it as indicated by the DISP parameter.

```
VSMtozOS  PROCESS  SNODE=CD.OS390.NDOEY
          COPY  FROM  (DSN=MYNAME.TESTFILE.VESDS.FB80S              -
                       LINK=(VSAMDSK,RPASWD,,RR,195)                -
                       DCB=(DSORG=VSAM,LRECL=80)                    -
                       DISP=SHR                                     -
                       PNODE)                                       -
                TO    (DSN=HLQ.VSAMTST.FILE                        -
                       UNIT=SYSDA                                   -
                       DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120) -
                       DISP=RPL                                     -
                       SNODE)
```

Copy a VM CMS Disk File to a z/OS Node

This Process copies a CMS Disk file from a VM/ESA node to a disk on a Sterling Connect:Direct for z/OS node. If the file exists, Sterling Connect:Direct replaces it. If the file does not exist, Sterling Connect:Direct creates it as indicated by the DISP parameter.

```
VMTOZOS  PROCESS      SNODE=CD.OS390.NODEX         -
                      NOTIFY=USERID
STEP01   COPY  FROM  (DSN='VMCMS FILE'             -
                      LINK=(USER1,RPASWWD,RR,191)  -
                      DISP=SHR                     -
                      PNODE)                       -
               TO    (DSN=HLQ.VMTEST.FILE         -
                      UNIT=SYSDA                   -
                      DISP=RPL                     -
                      SNODE)
```

Copy a VM CMS Sequential File to UNIX

This Process copies a VM CMS Sequential file to UNIX in binary format.

```
PROC01   PROCESS                                                    -
               &USER=MYUSR1                                         -
               &COMPRESS=COMPRESS                                   -
               &EXT=PRIME=X'40'                                     -
               &USERPW=ALL                                          -
               &SNODE=cd.unix.node                                  -
               SNODEID=(userx,passwdx)                              -
               PNODEID=(myuser1,passwd)                             -
               SNODE=&SNODE
STEP1    COPY FROM                                                  -
               ( PNODE                                              -
                 DSN='TESTFILE STRESS01'                            -
                 LINK=(&USER,&USERPW,RR,192)                        -
               )                                                    -
                 &COMPRESS         &EXT                           -
             TO                                                     -
              ( SNODE                                               -
                DSN='/tmp_mnt/home/fremont/mfinc1/hello'            -
                DISP=(RPL)                                          -
                SYSOPTS=":STRIP.BLANKS=NO:DATATYPE=BINARY:"         -
             )
```

Copy a CMS Sequential File from VM to Microsoft Windows

This Process copies a CMS Sequential file from VM to Microsoft Windows.

```
PROC01   PROCESS                                                -
               &CKPT=0K                                         -
               &COMPRESS=,                                      -
               &EXT=,                                           -
               &CUU1=0192                                       -
               &USER=MYUSR1                                     -
               &USERPW=ALL                                      -
               &SNODE=WINNT                                     -
               SNODE=&SNODE                                     -
               SNODEID=(MYUSER)
STEP1    COPY  FROM                                            -
               (                                                -
                 PNODE                                          -
                 LINK=(&USER,&USERPW,RR,&CUU1)                  -
                 DSN='TESTFILE STRESS04'                        -
                 DISP=SHR                                       -
               )                                                -
               CKPT=&CKPT  &COMPRESS  &EXT                      -
               TO                                               -
               (                                                -
                 SNODE                                          -
                 DSN='C:\OUTPUT\VM\OUT04'                       -
                 DISP=RPL                                       -
               )
```

Copy a VM CMS File to Microsoft Windows

This Process copies a VM CMS file to Microsoft Windows.

```
PROC01  PROCESS SNODE=CD.NT.V1300   SNODEID=(USERID,PASSWD)
STEP1    COPY                                                 -
               FROM                                           -
                     ( PNODE                                  -
                      LINK=(MYUSR1,ALL,RR,192)                -
                      DSN='TESTFILE FB80S'                    -
                      DISP=SHR                                -
                     )                                        -
               TO    (SNODE DSN='C:\OUTPUT\MYDATA' DISP=RPL)
```

Copy a DBCS File from Microsoft Windows to VMESA Using the KSCXEBC Translation Table

The following PC-to-host DBCS translation uses the supplied translation table KSCXEBC This sample COPY statement copies a data set from a PC to a host Sterling Connect:Direct for VM/ESA node.

```
/*********************************************************************/
/*           PC to HOST DBCS translation using table KSCXEBC         */
/*********************************************************************/
PCTOHOST  PROCESS SNODE=HOSTNODE                                      -
                  HOLD=CALL
STEP01    COPY                                                        -
                TO    (PNODE                                          -
                       DSN='hlq.PCFILE'                               -
                       DISP=(RPL,CATLG)                               -
                       UNIT=SYSDA                                     -
                       DCB=(RECFM=VB,LRECL=254,BLKSIZE=4096,DSORG=PS) -
                       SPACE=(254,(1000,100))                         -
                       SYSOPTS="DBCS=KSCXEBC"                         -
                      )                                               -
                FROM  (SNODE                                          -
                       DSN=PCFILE                                     -
                       DISP=SHR                                       -
                      )
```

- The copy step is named STEP01.
- The input data set is cataloged after successful completion of the Process.
- The DCB attributes specified in the TO clause of the COPY statement are used for file allocation.
- The SYSOPTS attribute specified in the TO clause of the COPY statement is used to define the supplied translation table KSCXEBC.
- UNIT has been specified on the PNODE only.

Copy a DBCS File from VMESA to Microsoft Windows Using the EBCXKSC Translation Table

The following host-to-PC DBCS translation uses the supplied translation table EBCXKSC. This example COPY statement copies a data set from a host Sterling Connect:Direct for VM/ESA to a PC node.

```
/*********************************************************************/
/*           HOST to PC DBCS translation using table EBCXKSC      */
/*********************************************************************/
HOSTTOPC  PROCESS SNODE=PCNODE                -
                  HOLD=CALL
STEP01    COPY                                -
                FROM  (PNODE                  -
                       DSN='hlq.HOSTFILE'     -
                       SYSOPTS="DBCS=EBCXKSC" -
                       DISP=(SHR)             -
                      )                       -
                TO    (SNODE                  -
                       DSN=PCFILE             -
                       DISP=RPL               -
                      )
```

- The copy step is named STEP01.
- The SYSOPTS attribute is specified in the FROM clause of the COPY statement is used to define the default translation table EBCXKSC.

Copy a DBCS File From UNIX to VMESA Using the EBCXKSC Translation Table

The following UNIX-to-host DBCS translation uses the default translation table EBCXKSC. This example COPY statement copies a data set from a UNIX to a host Sterling Connect:Direct for VM/ESA node.

```
/*****************************************************************/
/*           UNIX to HOST DBCS translation using table EBCXKSC    */
/*****************************************************************/
STEP01    COPY                                                    -
                  FROM (PNODE                                     -
                        DSN='hlq.UNIXFILE'                        -
                        SYSOPTS="DBCS=EBCXKSC"                    -
                        DISP=(SHR)                                -
                        )                                         -
                  TO   (SNODE                                     -
                        DSN='/unixfile'                           -
                        SYSOPTS=":xlate=no:strip.blanks=no:"      -
                        DISP=RPL                                  -
                        )
```

- The copy step is named STEP01.
- The SYSOPTS attribute is specified in the TO clause of the COPY statement is used to define the default translation table EBCXKSC.
- The SYSOPTS parameter on the FROM clause of the COPY statement is required.

Copy a DBCS File From VMESA to UNIX Using the KSCXEBC Translation Table

The following host-to-UNIX DBCS translation uses the default translation table KSCXEBC. This COPY statement copies a data set from a host Sterling Connect:Direct for VM/ESA to a UNIX node.

```
/*****************************************************************/
/*          HOST to UNIX DBCS translation using table KSCXEBC       */
/*****************************************************************/
STEP02  COPY                                                      -
              TO    (PNODE                                        -
                     DSN='hlq.HOSTFILE'                           -
                     SYSOPTS="DBCS=KSCXEBC"                       -
                     DISP=(RPL,CATLG)                             -
                     UNIT=SYSDA                                   -
                     DCB=(RECFM=VB,LRECL=254,BLKSIZE=4096,DSORG=PS) -
                     SPACE=(254, (1000,100))                      -
                    )                                             -
              FROM  (SNODE                                        -
                     DSN='/unixfile'                              -
                     SYSOPTS=":xlate=no:strip.blanks=no:"         -
                     DISP=SHR                                     -
                    )
```

- The copy step is named STEP02.
- The SYSOPTS attribute specified in the TO clause of the COPY statement is used to define the default translation table KSCXEBC.
- The DCB attributes specified on the TO clause of the COPY statement are used for file allocation.
- Unit is specified on the PNODE.
- The SYSOPTS parameter on the FROM clause of the COPY statement is required.

Copy a Non-VSAM File on VMESA

The following Process copies a non-VSAM file from a node named Chicago to a node named Minneapolis.

```
COPYSEQ  PROCESS  PNODE=CHICAGO SNODE=MINNEAPOLIS
STEP01   COPY  FROM  (DSN='MYFILE TEXT'          -
                      LINK=(VMID1,PASS1,RR,125))  -
               TO    (DSN='YOURFILE TEXT'        -
                      LINK=(VMID2,PASS2,W,126))
```

Copy All Files In a Group to a Destination with Fn Ft Unchanged on VMESA

This example illustrates copying all files in a group to a destination with Fn Ft unchanged. Suppose the following files were on a disk:

```
ABC ASSEMBLE
AAA ASSEMBLE
ABCD ASSEMBLE
SOURCE1 FILE
SOURCE2 FILE
```

Also, assume the following:

```
FROM GROUP='* *'
TO   GROUP='%1% %2%'
```

The group name specified, * * has two special pattern-matching characters (two asterisks), so the destination pattern supplied has two replacement symbols (**%1%** and **%2%**). Replacement symbol **%1%** is replaced by the characters that correspond to the first asterisk in all names determined to be in the group. Replacement symbol **%2%** is replaced by the characters that correspond to the second asterisk in all names determined to be in the group.

The following transfers occur:

```
FROM                TO
ABC ASSEMBLE  ----> ABC     ASSEMBLE
AAA ASSEMBLE  ----> AAA     ASSEMBLE
ABCD ASSEMBLE ----> ABCD    ASSEMBLE
SOURCE1 FILE  ----> SOURCE1 FILE
SOURCE2 FILE  ----> SOURCE2 FILE
```

A more efficient way to copy all the files from one minidisk to another is to specify the FROM group as * and the TO group as **%1%**.

Copy Selected Files from a Group to a Destination with Fn Ft Unchanged on VMESA

This example illustrates copying selected files from a group to a destination with Fn Ft unchanged.

The group name specified **a* *** includes all names that begin with an **a** or an **A**. If the specified destination pattern is **%1% %2%**, then the leading **a** from each file in the group is dropped when the destination name is built. This action occurs because the first asterisk in the group name corresponds to all the characters that follow but do not include the first **a**.

```
For    FROM  GROUP='a* *'
       TO    GROUP='a%1% %2%'
```

The following transfers occur:

```
FROM                TO
ABC ASSEMBLE  ----> ABC  ASSEMBLE
AAA ASSEMBLE  ----> AAA  ASSEMBLE
ABCD ASSEMBLE ----> ABCD ASSEMBLE
SOURCE1 FILE
SOURCE2 FILE
```

Copy Selected Files from a Group to a Destination with Added Characters In Fn Ft on VMESA

This example illustrates copying selected files from a group to a destination with added characters in Fn Ft.

The group name specified **a* *** includes all names that begin with an **a** or an **A**. Because the destination pattern specified includes the leading a specified in the group name and one additional **a**, the destination names built begin with **aa**.

```
For    FROM  GROUP='a* *'
       TO    GROUP='aa%1% %2%'
```

The following transfers occur:

```
FROM                TO
ABC ASSEMBLE  ----> AABC  ASSEMBLE
AAA ASSEMBLE  ----> AAAA  ASSEMBLE
ABCD ASSEMBLE ----> AABCD ASSEMBLE
SOURCE1 FILE
SOURCE2 FILE
```

Copy Selected Files from a Group to a Destination with Characters Stripped from Fn Ft on VMESA

This example illustrates copying selected files from a group to a destination with characters stripped from Fn Ft.

```
For    FROM  GROUP='s* *'
       TO    GROUP='%1% %2%'
```

The following transfers occur:

```
FROM                    TO

ABC      ASSEMBLE
AAA      ASSEMBLE
ABCD     ASSEMBLE
SOURCE1  FILE      ---->  SOURCE1  FILE
SOURCE2  FILE      ---->  SOURCE2  FILE
```

The leading s from each name found in the group is dropped from the names built with the supplied destination pattern.

Copy Selected Files with Equal Length Names from a Group to a Destination on VMESA

This example illustrates copying selected files with equal length names from a group to a destination.

```
For    FROM  GROUP='a?? *'
       TO    GROUP='a%1%%2% %3%'
```

The following transfers occur:

```
FROM                    TO
ABC      ASSEMBLE ----> ABC  ASSEMBLE
AAA      ASSEMBLE ----> AAA  ASSEMBLE
ABCD     ASSEMBLE
SOURCE1  FILE
SOURCE2  FILE
```

Copy Selected Files from a Group to a Destination with Fn Ft Formatting on VMESA

This example illustrates copying selected files from a group to a destination with Fn Ft formatting.

```
For    FROM GROUP='* *'
       TO   GROUP='%1%.%2%'
```

The following transfers occur:

```
FROM                     TO
ABC      ASSEMBLE  ---->  ABC.ASSEMBLE
AAA      ASSEMBLE  ---->  AAA.ASSEMBLE
ABCD     ASSEMBLE  ---->  ABCD.ASSEMBLE
SOURCE1  FILE      ---->  SOURCE1.FILE
SOURCE2  FILE      ---->  SOURCE2.FILE
```

Copy Selected Files from a Group to a Destination with Fn Ft Reversed and Formatted on VMESA

This example illustrates copying selected files from a group to a destination with Fn Ft reversed and formatted.

```
For   FROM GROUP='* *'
      TO   GROUP='%2%.%1%'
```

The following transfers occur:

```
FROM                     TO
ABC      ASSEMBLE  ---->  ASSEMBLE.ABC
AAA      ASSEMBLE  ---->  ASSEMBLE.AAA
ABCD     ASSEMBLE  ---->  ASSEMBLE.ABCD
SOURCE1  FILE      ---->  FILE.SOURCE1
SOURCE2  FILE      ---->  FILE.SOURCE2
```

The special symbols used in the destination pattern are specified in reverse order, which causes the destination names that are built to appear reversed.

CAUTION:
Do not make group file copies from disks linked R/W. The results are unpredictable.

Use SYSOPTS for DBCS in VMESA

The SYSOPTS statement declares that a Process is transferring a DBCS file. Include this statement on the host node COPY statement.

Support for multiple transfers with multiple translation tables is possible. All Processes support compression and checkpointing.

The following example uses the table name EBCXKSC and the default values x'0E', for so, and x'0F' for si.

```
SYSOPTS="DBCS=(EBCXKSC,0E,0F)"
```

The following example uses the table name KSCXEBC and the default values x'0E', for so, and x'0F' for si.

```
SYSOPTS="DBCS=(KSCXEBC,0E,0F)"
```

The following example uses the table name EBCXKSC and the NOSO value x'00' for so and si.

```
SYSOPTS="DBCS=(EBCXKSC,00,00)"
```

The following example uses the table name EBCXKSC and takes the defaults for so and si.

```
SYSOPTS="DBCS=(EBCXKSC)"
```

The following example uses the table name USERTAB and takes the defaults for so and si. USERTAB is a user-defined, customized translation table.

```
SYSOPTS="DBCS=USERTAB"
```

VSE

Copy a File from z/OS to VSE (DYNAM/T Tape Files)

This multi-step Process copies various sequential files from a z/OS node (the SNODE) to VSE DYNAM/T tape files at the PNODE. Different record formats are specified in the steps. Note that a TYPE file record is specified in STEP03. This record contains the DCB and UNIT information for this file. The TYPE record must be in the TYPE file at the destination (VSE node).

```
DYMTVSE  PROCESS     SNODE=CD.VSE.DALLAS  NOTIFY=%USER
STEP01   COPY  FROM (DSN=$ABC.FDATA                                  -
                     DISP=SHR)                                       -
                TO   (DSN=FDATA.DYMT.STEP01                          -
                     DCB=(DSORG=PS BLKSIZE=80 LRECL=80 RECFM=FB)     -
                     DISP=RPL                                        -
                     LABEL=(RETPD=0007)                              -
                     UNIT=TNOASGN                                    -
                     SNODE)
STEP02  COPY  FROM  (DSN=$ABC.SOURCE                                 -
                     DISP=SHR)                                       -
                TO   (DSN=SOURCE.DYMT.STEP02                         -
                     DCB=(DSORG=PS BLKSIZE=3120 LRECL=80 RECFM=FB)   -
                     UNIT=TNOASGN                                    -
                     LABEL=(RETPD=0007)                              -
                     DISP=RPL                                        -
                     SNODE)
STEP03  COPY  FROM  (DSN=$ABC.REPORT                                 -
                     DISP=SHR)                                       -
                TO   (DSN=REPORT.DYMT.STEP03                         -
                     TYPE=OS390DYMT                                  -
                     LABEL=(RETPD=0007)                              -
                     DISP=RPL                                        -
                     SNODE)
STEP04  COPY  FROM  (DSN=$ABC.UDATA                                  -
                     DISP=SHR)                                       -
                TO   (DSN=UDATA.DYMT.STEP04                          -
                     DCB=(DSORG=PS BLKSIZE=80 LRECL=0 RECFM=U)       -
                     UNIT=TNOASGN                                    -
                     DISP=RPL                                        -
                     LABEL=(RETPD=0007)                              -
                     SNODE)
STEP05  COPY  FROM  (DSN=$ABC.VDATA                                  -
                     DISP=SHR)                                       -
                TO   (DSN=VDATA.DYMT.STEP05                          -
                     DCB=(DSORG=PS BLKSIZE=88 LRECL=84 RECFM=V)      -
                     UNIT=TNOASGN                                    -
                     LABEL=(RETPD=0007)                              -
                     DISP=RPL  SNODE)
```

Copy a z/OS PDS Member to a New VSE File in a DYNAM Pool

This Process copies a PDS member on z/OS to a DYNAM-controlled file on VSE. The file on VSE will be dynamically allocated by DYNAM in a virtual disk pool defined to DYNAM as POOL01.

```
VSEOS390  PROCESS      PNODE=SC.VSE.NODE1 SNODE=SC.OS390.NODE1
STEP01    COPY  FROM  (DSN=OS390.PDS.DATASET(TESTDATA)         -
                      DISP=(SHR,KEEP)                         -
                      SNODE)                                  -
                TO    (DSN=VSE.DYNAM.FILE                     -
                      DISP=NEW                                -
                      UNIT=DNOASGN                            -
                      VOL=SER=POOL01                          -
                      SPACE=(1,(300))                         -
                      PNODE)
```

Copy a z/OS BSAM File to a VSE-Controlled Disk Data Set

Use this Process to transfer a z/OS/ESA BSAM file into a VSE CA-DYNAM/D or CA-EPIC controlled disk data set. The disk data set has already been defined to the appropriate system catalog. You do not need to specify DCB attributes for the output data set, these will be taken from the input data set.

This Process was written with symbolics for substitution.

```
ZOS2DYD1  PROC   SNODE=SC.OS390.NODE                                    -
                 PNODE=SC.VSE.NODE                                     -
                 &VSEDSN=USER01.TEST.COPYFILE
STEP0001  COPY                                                         -
          FROM ( SNODE                                                 -
               DSN=USER01.TEST.OS390PRT01                              -
               DISP=SHR                                                -
               )                                                       -
          TO   ( PNODE                                                 -
               DSN=&VSEDSN                                             -
               UNIT=DLBLONLY                                           -
               DCB=(DSORG=PS,RECFM=FBA,LRECL=121,BLKSIZE=27951)        -
               )                                                       -
               COMPRESS
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                            -
                       PARM=('GOOD',&VSEDSN))                          -
                       PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                            -
                       PARM=('FAIL',&VSEDSN))                          -
                       PNODE
         EIF
```

Copy a z/OS Sequential Data Set or PDS to a VSE-Controlled Tape Data Set

This Process copies either a z/OS sequential data set or a z/OS partitioned data set into a CA-DYNAM/T or CA-EPIC non-controlled tape data set. The input resides on z/OS and the output file is written to tape (or cartridge) on VSE.

```
ZOS2DYT1 PROC    SNODE=SC.OS390.NODE0                                  -
                 PNODE=SC.VSE21.USER01
STEP0001 COPY  FROM ( SNODE                                            -
                    DSN=RPITT1.LARGE.OS390.FILE                        -
                    DISP=SHR                                           -
                    )                                                  -
               TO   ( PNODE                                            -
                    DSN=USER01.TEST.NONDYNAM                           -
                    UNIT=TNOASGN                                       -
                    LABEL=(1,SL)                                       -
                    DISP=(NEW,CATLG)                                   -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=1600)      -
                    )
```

Copy a z/OS PDS Member to a VSE BSAM Sublibrary Member

This Process copies a z/OS PDS library member into a VSE BSAM sublibrary member. This sample Process is coded with symbolic parameters to allow you to use a generic Process to move any type of members (Dump, OBJ, Phase, Source, Processes, JCL) Use this Process when you are moving files from z/OS to VSE with the Process running on VSE.

The disk data set has already been defined to the appropriate system catalog. This Process was written with symbolics for substitution. When you reference BSAM libraries in a Sterling Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

When you copy data into a VSE BSAM library, you must add either RECFM=F or RECFM=V to your DCB parameter. This specification depends on the type of input file. If you do not include the RECFM, the Process fails with the message SVSJ122I.

```
ZOS2LIB1  PROC   SNODE=SC.OS390.NODE                                    -
                 PNODE=SC.VSE.NODE                                      -
                 &MEMBER=,                                              -
                 &OS390PDS=USER01.PROCESS.LIB                           -
                 &VSELIB=CONN.DIRECT.LIBRARIES                          -
                 &VSESUB=USER01                                         -
                 &VSETYP=N                                              -
                 &VSEVOL=USER03
STEP0001 COPY  FROM ( SNODE                                             -
                      DSN=&OS390PDS                                     -
                      DISP=SHR                                          -
                      SELECT=(&MEMBER)                                  -
                      )                                                 -
               TO   ( PNODE                                             -
                      DSN=&VSELIB                                       -
                      DISP=SHR                                          -
                      UNIT=DISK                                         -
                      DCB=(DSORG=PS,RECFM=F)                            -
                      VOL=SER=&VSEVOL                                   -
                      LIBR=(REPLACE=YES                                 -
                            SLIBDISP=SHR                                -
                            SUBLIB=&VSESUB                              -
                            TYPE=&VSETYP)                               -
                      )                                                 -
STEP0002 IF    (STEP0001 EQ 0) THEN
                      RUN TASK (PGM=DMNOTIFY,                           -
                          PARM=('GOOD',&VSELIB))                        -
                          PNODE
         ELSE
                      RUN TASK (PGM=DMNOTIFY,                           -
                          PARM=('FAIL',&VSELIB))                        -
                          PNODE
         EIF
```

Copy a z/OS PDS Member to a VSE VSAM Sublibrary Member

This Process copies a z/OS PDS library member into a VSE VSAM sublibrary member. This sample Process is coded with symbolic parameters to allow you to use a generic Process to move any type of members (Dump, OBJ, Phase, Source, Processes, JCL). Use this Process when you are moving files from z/OS to VSE with the Process running on VSE.

For VSAM libraries you must specify the DSN and DSORG parameters on the FROM statement. You can optionally specify the catalog parameter if needed.

```
ZOS2LIB2 PROC    SNODE=SC.OS390.NODE                                    -
                 PNODE=SC.VSE.NODE                                      -
                 &MEMBER=,                                              -
                 &OS390PDS=USER01.TEST.JCLLIB                           -
                 %VSELIB=CONN.DIRECT.LIB1                               -
                 &VSESUB=RXUSER01                                       -
                 &VSETYP=JCL
STEP0001 COPY  FROM ( SNODE                                             -
                      DSN=&OS390PDS                                     -
                      DISP=SHR                                          -
                      SELECT=(&MEMBER)                                  -
                      )                                                 -
                TO   ( PNODE                                            -
                      DSN=&VSELIB                                       -
                      DISP=SHR                                          -
                      DCB=(DSORG=VSAM)                                  -
                      LIBR=(REPLACE=YES                                 -
                            SLIBDISP=SHR                                -
                            SUBLIB=&VSESUB                              -
                            TYPE=&VSETYP)                               -
                      )
```

Copy a z/OS Sequential Data Set or z/OS PDS to a VSE-Controlled Tape Data Set

This Process copies either a z/OS sequential data set or a z/OS partitioned data set into a CA-DYNAM/T or CA-EPIC controlled tape data set. The input is from z/OS with the Process running on VSE. The disk data set has already been defined to the appropriate system catalog. This Process was written with symbolics for substitution.

```
ZOS2TAP1  PROC   SNODE=SC.OS390.NODE                                    -
                 PNODE=SC.VSE.NODE                                      -
                 &VSECUU=CART                                           -
                 &VSEDSN=TEST.TAPE.FILE
STEP0001  COPY FROM ( SNODE                                             -
                      DSN=RPITT1.LARGE.OS390.FILE                       -
                      DISP=SHR                                          -
                      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=800)      -
                      )                                                 -
          TO          ( PNODE                                           -
                      DSN=&VSEDSN                                       -
                      UNIT=&VSECUU                                      -
                      LABEL=(1,SL)                                      -
                      DISP=(NEW,KEEP)                                   -
                      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=16000)    -
                      )
STEP0002 IF    (STEP0001 EQ 0) THEN
                      RUN TASK (PGM=DMNOTIFY,                           -
                          PARM=('GOOD',&VSEDSN))                        -
                          PNODE
         ELSE
                      RUN TASK (PGM=DMNOTIFY,                           -
                          PARM=('FAIL',&VSEDSN))                        -
                          PNODE
         EIF
```

Copy an HP NonStop File to a VSE VSAM File

This Process copies a file from an HP NonStop node to a VSE VSAM file. The transfer is initiated by the VSE node. The SYSOPTS SET XLATE ON parameter enables ASCII to EBCDIC translation.

```
VSE2TAN  PROCESS      PNODE=SC.VSE.NODE1 SNODE=HPNONSTOP.NODE -
                      SNODEID=(HPNONSTOP.NODE)                -
                      SNODEID=(123.456,TANUSR)
STEP01   COPY  FROM  (DSN='\CLX.$SUP1.XFILES.SENDTEST'        -
                      DISP=SHR                                -
                      SYSOPTS="SET XLATE ON"                  -
                      SNODE)                                  -
               TO    (DSN=VSE.TEST.VSAM                       -
                      DISP=RPL                                -
                      DCB=(DSORG=VSAM)                        -
                      PNODE)
```

Copy a VSE VSAM File to an HP NonStop Node

This Process copies a VSE VSAM file to an HP NonStop ESDS file. The transfer is initiated by the VSE node. The SYSOPTS SET XLATE ON parameter enables EBCDIC to ASCII translation.

```
VSE2TAN  PROCESS      PNODE=SC.VSE.NODE1 SNODE=HPNONSTOP.NODE -
                      SNODEID=(123.456,TANUSR)
STEP01   COPY  FROM  (DSN=VSE.TEST.VSAM                       -
                      DISP=RPL                                -
                      DCB=(DSORG=VSAM)                        -
                      PNODE)                                  -
               TO    (DSN='\$SUP1.TSTPROC.VSETEST'            -
                      DISP=NEW                                -
                      SYSOPTS=\"'SET XLATE ON\                -
                              \ , TYPE=E\                     -
                              \ , REC=100\                    -
                              \ , EXT= (100,100)'"\)          -
                      SNODE)
```

Copy a VSE DYNAM-Controlled File to a VM Node

This Process copies a DYNAM-controlled file from a VSE node to a VM node. The file is copied to the CDUSR CMS ID.

```
VSE2VM  PROCESS      PNODE=SC.VSE.NODE1 SNODE=SC.VM.NODE1
STEP01  COPY  FROM  (DSN=VSE.DYNAM.FILE                          -
                     DISP=SHR                                    -
                     UNIT=DLBLONLY                               -
                     DCB=(DSORG=PS,LRECL=80,BLKSIZE=8000,RECFM=FB) -
                     PNODE)                                      -
              TO    (DSN='VMDYND TEMP02'                        -
                     DISP=RPL                                    -
                     LINK=(CDUSR,XCDUSR,RR,301)                  -
                     SNODE)
```

Use a Typekey to Copy a VSE DYNAM-Controlled File to a VM Node

This Process copies a DYNAM-controlled file from a VSE node to a VM node. DCB and UNIT information is supplied through a TYPEKEY named DYNAMD.

```
VSE2VM  PROCESS      PNODE=SC.VSE.NODE1 SNODE=SC.VM.NODE1
STEP01  COPY  FROM  (DSN=VSE.DYNAM.FILE                    -
                     TYPE=DYNAMD                           -
                     PNODE)                                -
              TO    (DSN='VMDYND TEMP02'                   -
                     DISP=RPL                              -
                     LINK=(CDUSR,XCDUSR,RR,301)            -
                     SNODE)
```

Copy a VSE Sequential File to an OpenVMS Node

This Process copies a VSE sequential file to an OpenVMS node.

```
PROC01  PROCESS      PNODE=SC.VSE.NODE1 SNODE=SC.VMS.NODE2         -
                     SNODEID=(VMSUSR,PASSWD)
STEP01  COPY  FROM  (DSN=VSE.TEST.DATA                             -
                     DCB=(BLKSIZE=2400,DSORG=PS,LRECL=80,RECFM=FB) -
                     UNIT=241                                      -
                     SPACE=(10620,(45))                            -
                     DISP=SHR)                                     -
                     CKPT=0K                                       -
              TO    (DSN='$SUP:<VMSUSR>DATA.TST'                  -
                     SNODE)
```

Copy a VSE Sequential File to Another VSE Sequential File

This Process copies a sequential file from one VSE node to another VSE node, with checkpointing at 128K intervals.

```
PROC01  PROCESS      PNODE=SC.VSE.NODE1 SNODE=SC.VSE.NODE2          -
                     PNODEID=(SUPERUSR,SUPERUSR)
STEP01  COPY  FROM  (DSN=VSE.TEST.DATA                             -
                     DCB=(BLKSIZE=2400,DSORG=PS,LRECL=80,RECFM=FB) -
                     UNIT=DISK                                     -
                     VOL=SER=123456                                -
                     SPACE=(10620,(15))                            -
                     DISP=SHR)                                     -
                     CKPT=128K                                     -
              TO    (DSN=VSE.CKPT.TEST                            -
                     DCB=(BLKSIZE=24000,DSORG=PS,LRECL=80,RECFM=FB) -
                     UNIT=243                                      -
                     SPACE=(10530,(45))                            -
                     DISP=(NEW,KEEP))
```

Copy a VSE Non-Labeled Tape to a VSE Sequential File

This Process copies the second data set on a non-labeled tape from one VSE node to a sequential file on another VSE node.

```
VSETAPE  PROCESS      PNODE=SC.VSE.NODE1 SNODE=SC.VSE.NODE2
STEP01   COPY  FROM  (DSN=VSE.NLTAPE                                -
                      UNIT=CART                                    -
                      DCB=(DSORG=PS,LRECL=80,BLKSIZE=18000,RECFM=FB) -
                      LABEL=(2,NL)                                 -
                      VOL=SER=123456                               -
                      PNODE)                                       -
               TO    (DSN=VSE.SEQFILE                              -
                      UNIT=TAPE                                    -
                      SPACE=(3645,(60))                            -
                      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=14400) -
                      DISP=NEW                                     -
                      SNODE)
```

Copy the Sterling Connect:Direct Message File to SL Tape on VSE

This Process copies the Sterling Connect:Direct for z/OS VSAM message file to a standard label tape device on another VSE node.

```
VSETAPE  PROCESS      PNODE=SC.VSE.NODE1 SNODE=SC.VSE.NODE2
COPY01   COPY  FROM  (DSN=ABC.MSG                                    -
                      DISP=SHR                                       -
                      DCB=(DSORG=VSAM)                               -
                      PNODE)                                         -
               TO    (DSN=ABC.MSG.TAPE                               -
                      DISP=NEW                                       -
                      UNIT=TAPE                                      -
                      DCB=(DSORG=PS,LRECL=80,BLKSIZE=12000,RECFM=VB) -
                      LABEL=(1,SL)                                   -
                      DISP=NEW                                       -
                      SNODE)
```

Copy a Non-managed Disk Data Set into Another Non-managed CKD Disk Data Set (VSE)

Use this Process to copy a non-managed disk data set into another non-managed disk data set residing on a CKD device. This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing and uses symbolic values.

```
DSK2DSK1 PROC  PNODE=SC.VSE.NODE                                    -
               SNODE=SC.VSE.NODE                                    -
               &VSEDSN=GGREG1.TEST.NODYNAM1
STEP0001 COPY  FROM ( PNODE                                         -
                      DSN=LRR.LREC480.ADDX                          -
                      DISP=(SHR)                                    -
                      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=480)  -
                      VOL=SER=USER01                                -
                      )                                             -
               TO    ( SNODE                                       -
                     DSN=&VSEDSN                                    -
                     DISP=(RPL)                                     -
                     VOL=SER=USER02                                 -
                     SPACE=(6055,(25))                              -
                     DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=13600) -
                     )
STEP0002 IF    (STEP0001 EQ 0) THEN
                     RUN TASK (PGM=DMNOTIFY,                        -
                         PARM=('GOOD',&VSEDSN))                     -
                         PNODE
         ELSE
                     RUN TASK (PGM=DMNOTIFY,                        -
                         PARM=('FAIL',&VSEDSN))                     -
                         PNODE
         EIF
```

Copy a Non-controlled Disk Data Set to a Managed CKD Disk Data Set (VSE)

This Process copies a non-DYNAM/D or non-EPIC controlled disk data set into a DYNAM/D or EPIC managed CKD disk data set. The disk data set has already

been defined to the appropriate system catalog. This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing and uses symbolic values.

```
DSK2DYD1 PROC  PNODE=SC.VSE.NODE                                          -
               SNODE=SC.VSE.NODE                                          -
               &VSEDSN=USER01.TEST.GDGCOPY1
STEP0001 COPY  FROM ( PNODE                                               -
                    DSN=LRR.LREC480.ADDX                                  -
                    DISP=SHR                                              -
                    VOL=SER=USER01                                        -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=800)          -
                    )                                                     -
               TO   ( SNODE                                               -
                    DSN=&VSEDSN                                           -
                    UNIT=DLBLONLY                                         -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=27920)        -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('GOOD',&VSEDSN))                            -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('FAIL',&VSEDSN))                            -
                        PNODE
         EIF
```

Copy a Non-managed Disk File into a Start Track 1 FBA Non-controlled Data Set (VSE)

Use this Process to copy a Non-managed disk file into a DYNAM/D or EPIC start-track 1 FBA non-controlled data set.

This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing and uses symbolic values. CA-DYNAM/D or CA-EPIC will perform the disk allocation for Sterling Connect:Direct but since the data set is allocated as a start-track 1 data set with a vol=ser it will not be a managed data set.

```
DSK2DYD2 PROC  PNODE=SC.VSE.NODE                                          -
               SNODE=SC.VSE.NODE                                          -
               &VSEDSN=USER01.TEST.FILENAME
STEP0001 COPY  FROM ( PNODE                                               -
                    DSN=LRR.LREC480.ADDX                                  -
                    DISP=SHR                                              -
                    VOL=SER=USER01                                        -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=800)          -
                    )                                                     -
               TO   ( SNODE                                               -
                    DSN=&VSEDSN                                           -
                    VOL=SERUSER04                                         -
                    UNIT=DNOASGN                                          -
                    LABEL=(,,,EXPDT=99365)                                -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=27920)        -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('GOOD',&VSEDSN))                            -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('FAIL',&VSEDSN))                            -
                        PNODE
         EIF
```

Copy to Non-TMS Controlled Tapes (VSE)

This Process copies two non-TMS controlled tapes. The input tape is 3480/3490 cartridge and the output tape is reel (3420). This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing. This Process uses symbolic values.

```
TAP2TAP1 PROC  PNODE=SC.VSE.USER01                                        -
               SNODE=SC.VSE.USER01                                        -
               &FCUU=CART                                                 -
               &TCUU=TAPE
STEP001  COPY  FROM ( PNODE                                               -
                      DSN=TEST.TAPE.FILE                                  -
                      UNIT=&FCUU                                          -
                      LABEL=(1,SL)                                        -
                      VOL=SER=(807012)                                    -
                      DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)                 -
                      )                                                   -
               TO     ( SNODE                                            -
                      DSN=NON.DYNAM.TAPE                                  -
                      UNIT=&TCUU                                          -
                      LABEL=(1,SL,,,EXPDT=99365)                          -
                      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=16000)      -
                      )                                                   -
                      COMPRESS
```

Copy a Non-managed Disk File to a CA-DYNAM/D or CA-EPIC Start Track 1 FBA Non-controlled Data Set (VSE)

Use this Process to copy a non-managed disk file into a DYNAM/D or EPIC start-track 1 FBA non-controlled data set. This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing.

This Process uses symbolic values. CA-DYNAM/D or CA-EPIC will perform the disk allocation for Sterling Connect:Direct. The data set is allocated as a start-track **1** data set with a VOL=SER it will not be a managed data set.

```
DSK2DYD3 PROC  PNODE=SC.VSE.NODE                                          -
               SNODE=SC.VSE.NODE                                          -
               &VSEDSN=USER01.TEST.NONDYD                                 -
               &VOLSER=FBA001
STEP0001 COPY  FROM ( PNODE                                               -
                    DSN=LRR.LREC480.ADDX                                  -
                    DISP=SHR                                              -
                    VOL=SER=USER01                                        -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=800)          -
                    )                                                     -
               TO   ( SNODE                                               -
                    DSN=&VSEDSN                                           -
                    UNIT=DNOASGN                                          -
                    VOL=SER=&VOLSER                                       -
                    SPACE=(1,(5),RLSE)                                    -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=27920)        -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('GOOD',&VSEDSN))                            -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('FAIL',&VSEDSN))                            -
                        PNODE
         EIF
```

Print a Managed Disk Data Set (VSE)

Use this Process to print the contents of a CA-DYNAM/D or CA-EPIC managed disk data set. The output will become a LST queue member under the name of &JBNAME. The disk data set has already been defined to the appropriate system catalog.

```
DYD2LST1 PROC  SNODE=SC.VSE.NODE                                          -
               PNODE=SC.VSE.NODE                                          -
               &JBNAME=GGGDYD00                                           -
               &JBNUMB=0000                                               -
               &VSEDSN=USER01.TEST.GDGPOWR1
STEP0001 COPY  FROM ( PNODE                                               -
                    DSN=&VSEDSN                                           -
                    DISP=SHR                                              -
                    UNIT=DLBLONLY                                         -
                    DCB=(RECFM=FBM,LRECL=133,BLKSIZE=1330)                -
                    )                                                     -
               TO   ( SNODE                                               -
                    DSN=&JBNAME..&JBNUMB                                  -
                    LST=(                                                 -
                      CC=M                                                -
                      CLASS=Q                                             -
                      COPIES=2                                            -
                      DISP=L)                                             -
                    )
```

The previous Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing and uses symbolic values. You must specify the input DCB parameter (RECFM, LRECL); this information will be copied to the output data set.

Copy a Non-controlled Sequential File to a MSAM File (VSE)

This Process copies a non-controlled BSAM (sequential) file into a MSAM (VSAM Managed SAM) file. The disk data set has already been defined to the appropriate system catalog (the default ESDS model). This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing and uses symbolic values.

When you reference BSAM libraries in a Sterling Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

```
DSK2MSM1 PROC PNODE=SC.VSE.NODE                                          -
              SNODE=SC.VSE.NODE                                          -
              &VSEDSN=USER01.TEST.MSAMFIL1
STEP0001 COPY  FROM ( PNODE                                              -
                    DSN=LRR.LREC480.ADDX                                 -
                    DISP=SHR                                             -
                    VOL=SER=USER01                                       -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=800)         -
                    )                                                    -
               TO   ( SNODE                                              -
                    DSN=&VSEDSN                                          -
                    DISP=RPL                                             -
                    UNIT=DISK                                            -
                    VOL=SER=USER06                                       -
                    SPACE=(80,(500,300))                                 -
                    VSAMCAT=(VSE.COMMON.CATALOG,X,X,,123)                -
                    DCB=(DSORG=MSAM,RECFM=FB,LRECL=80,BLKSIZE=16000)     -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                              -
                        PARM=('GOOD',&VSEDSN))                           -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                              -
                        PARM=('FAIL',&VSEDSN))                           -
                        PNODE
         EIF
```

Copy a Non-controlled Tape Data Set to a Controlled Disk File (VSE)

This Process copies a non-CA-DYNAM/T or CA-EPIC controlled tape data set into a controlled disk file. The disk data set has already been defined to the appropriate system catalog. This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing.

```
TAP2DYD1 PROC        PNODE=SC.VSE.NODE                                   -
                     SNODE=SC.VSE.NODE
STEP001  COPY  FROM ( PNODE                                              -
                    (DSN=TEST.TAPE.FILE                                  -
                     UNIT=5A0                                            -
                     LABEL=(1,NL)                                        -
                     VOL=SER=(777777)                                    -
                     DCB=(RECFM=FB,LRECL=1500,BLKSIZE=22500)             -
                    )                                                    -
               TO   ( SNODE                                              -
                    DSN=USER01.TEST.GDGCOPY1                             -
                    UNIT=DLBLONLY                                        -
                    LABEL=(EXPDT=99365)                                  -
                    DCB=(RECFM=FB,LRECL=1500,BLKSIZE=22500)              -
                    )
```

Copy Non-managed Disk Data Set to a Non-managed Tape Data Set (VSE)

This Process copies a non-managed disk data set into a non-managed tape data set and runs on the same Sterling Connect:Direct node using PNODE=SNODE processing. This Process uses symbolic values.

```
DSK2TAP1 PROC  PNODE=SC.VSE.NODE                                          -
               SNODE=SC.VSE.NODE                                          -
               &VSECUU=CART                                               -
               &VSEDSN=TEST.TAPE.FILE
STEP0001 COPY  FROM ( PNODE                                               -
                    DSN=USER01.TEST.NODYNAM1                              -
                    DISP=(SHR)                                            -
                    UNIT=DISK                                             -
                    VOL=SER=DOSRES                                        -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=8000)         -
                    )                                                     -
               TO   ( SNODE                                               -
                    DSN=&VSEDSN                                           -
                    UNIT=&VSECUU                                          -
                    LABEL=(1,SL)                                          -
                    DISP=(NEW,KEEP)                                       -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=800)          -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('GOOD',&VSEDSN))                            -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('FAIL',&VSEDSN))                            -
                        PNODE
         EIF
```

Copy a Managed Disk Data Set to Another Managed Data Set (VSE)

Use this Process to copy either a CA-DYNAM/D or CA-EPIC managed disk data set into another DYNAM/D or EPIC managed data set and reblock the output data set. The disk data set has already been defined to the appropriate system catalog.

```
DYD2DYD1 PROC  PNODE=SC.VSE.NODE                                          -
               SNODE=SC.VSE.NODE                                          -
               &VSEDSN=USER01.TEST.GDGCOPY2
STEP0001 COPY  FROM ( PNODE                                               -
                    DSN=USER01.TEST.GDGCOPY1                              -
                    DISP=(SHR)                                            -
                    UNIT=DLBLONLY                                         -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=16000)        -
                    )                                                     -
               TO   ( SNODE                                               -
                    DSN=&VSEDSN                                           -
                    UNIT=DLBLONLY                                         -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=27920)        -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('GOOD',&VSEDSN))                            -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('FAIL',&VSEDSN))                            -
                        PNODE
         EIF
```

Copy a Managed Generation Disk Data Set to Another Managed Data Set (VSE)

Use this Process to copy either a CA-DYNAM/D or CA-EPIC managed disk data set into another DYNAM/D or EPIC managed data set. The input data set is CKD and the output data set is FBA. The disk data set has already been defined to the appropriate system catalog.

This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing. This Process uses symbolic values.

```
DYD2DYD2 PROC  PNODE=SC.VSE.NODE                                          -
               SNODE=SC.VSE.NODE                                          -
               &CKDDSN=USER01.TEST.GDGCOPY2                               -
               &FBADSN=USER01.TEST.FBACOPY1
STEP0001 COPY  FROM ( PNODE                                               -
                    DSN=&CKDDSN                                           -
                    UNIT=DLBLONLY                                         -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=16000)        -
                    )                                                     -
               TO   ( SNODE                                               -
                    DSN=&FBADSN                                           -
                    UNIT=DLBLONLY                                         -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=27920)        -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('GOOD',&FBADSN))                            -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('FAIL',&FBADSN))                            -
                        PNODE
         EIF
```

Copy a Controlled Disk Data Set to a Controlled Tape Output File (VSE)

Use this Process to copy a CA-DYNAM/D or CA-EPIC controlled disk data set to a CA-DYNAM/T or CA-EPIC controlled tape output file on the same Sterling Connect:Direct node by using PNODE=SNODE. All of the disk and tape data set names have been predefined to the appropriate system catalog.

```
DYD2DYT1 PROC  PNODE=SC.VSE.NODE                                          -
               SNODE=SC.VSE.NODE                                          -
               &VSEDSN=USER01.TEST.TAPE1
STEP0001 COPY  FROM ( PNODE                                               -
                      DSN=USER01.TEST.COPYFILE                            -
                      UNIT=DLBLONLY                                       -
                      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=16000)      -
                      )                                                   -
               TO     ( SNODE                                             -
                      DSN=&VSEDSN                                         -
                      UNIT=TNOASGN                                        -
                      LABEL=(1,SL)                                        -
                      DISP=(NEW,CATLG)                                    -
                      DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=32000)      -
                      )
STEP0002 IF    (STEP0001 EQ 0) THEN
                     RUN TASK (PGM=DMNOTIFY,                              -
                         PARM=('GOOD',&VSEDSN))                           -
                         PNODE
         ELSE
                     RUN TASK (PGM=DMNOTIFY,                              -
                         PARM=('FAIL',&VSEDSN))                           -
                         PNODE
         EIF
```

Copy a Controlled BSAM Data Set to a MSAM Output Data Set (VSE)

This Process copies from a CA-DYNAM/D or CA-EPIC controlled BSAM data set into a MSAM (VSAM Managed SAM) output data set. The disk data set has already been defined to the appropriate system catalog (the default ESDS model).

When you reference BSAM libraries in a Sterling Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

You can ignore the Sterling Connect:Direct information message: SVSG501I VSAM OPEN ERROR='A0'. ASSUMING ESDS. RETRYING OPEN.

```
DYD2MSM1 PROC PNODE=SC.VSE.NODE                                          -
              SNODE=SC.VSE.NODE                                          -
              &VSEDSN=USER01.TEST.MSAMFIL1
STEP0001 COPY  FROM ( PNODE                                              -
                    DSN=USER01.TEST.GDGCOPY1                             -
                    DISP=(SHR)                                           -
                    UNIT=DLBLONLY                                        -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=16000)       -
                    )                                                    -
               TO   ( SNODE                                              -
                    DSN=&VSEDSN                                          -
                    DISP=RPL                                             -
                    UNIT=DISK                                            -
                    VOL=SER=USER06                                       -
                    SPACE=(80,(500,300))                                 -
                    VSAMCAT=(VSE.COMMON.CATALOG,X,X,,123)                -
                    DCB=(DSORG=MSAM,RECFM=FB,LRECL=80,BLKSIZE=16000)     -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                              -
                        PARM=('GOOD',&VSEDSN))                           -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                              -
                        PARM=('FAIL',&VSEDSN))                           -
                        PNODE
         EIF
```

This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing. This Process uses symbolic values.

Copy a Controlled Tape Data Set to a Controlled FBA Disk Output Data Set (VSE)

Use this Process to copy a CA-DYNAM/D or CA-EPIC controlled tape data set to CA-DYNAM/D or CA-EPIC controlled FBA disk output data set. The disk data set has already been defined to the appropriate system catalog.

This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing and uses symbolic values.

```
DYT2DYD1 PROC  PNODE=SC.VSE.NODE                                         -
               SNODE=SC.VSE.NODE                                         -
               &VSEDSN=USER01.TEST.FBAFILE1
STEP0001 COPY  FROM  ( PNODE                                             -
                     DSN=USER01.TEST.TAPE1                               -
                     UNIT=TNOASGN                                        -
                     DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=32000)      -
                     )                                                   -
               TO    ( SNODE                                             -
                     DSN=&VSEDSN                                         -
                     DISP=(SHR)                                          -
                     UNIT=DLBLONLY                                       -
                     )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                              -
                        PARM=('GOOD',&VSEDSN))                           -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                              -
                        PARM=('FAIL',&VSEDSN))                           -
                        PNODE
         EIF
```

Copy a Controlled CKD Disk Data Set to a non-controlled Tape Data Set (VSE)

This Process copies a CA-DYNAM/D or CA-EPIC controlled CKD disk data set to a non-CA-DYNAM/T or CA-EPIC controlled tape data set. The disk data set has already been defined to the appropriate system catalog. The output data set DCB attributes will be propagated from the input file attributes. This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing and uses symbolic values.

```
DYD2TAP1 PROC  PNODE=SC.VSE.NODE                                          -
               SNODE=SC.VSE.NODE                                          -
               &VSECUU=CART                                               -
               &VSEDSN=TEST.TAPE.FILE
STEP0001 COPY  FROM ( PNODE                                               -
                    DSN=USER01.TEST.GDGCOPY1                              -
                    UNIT=DLBLONLY                                         -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=8000)         -
                    )                                                     -
               TO   ( SNODE                                               -
                    DSN=&VSEDSN                                           -
                    UNIT=&VSECUU                                          -
                    LABEL=(1,SL)                                          -
                    DISP=(NEW,KEEP)                                       -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('GOOD',&VSEDSN))                            -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                               -
                        PARM=('FAIL',&VSEDSN))                            -
                        PNODE
         EIF
```

Copy a VSE Sublibrary Member from a BSAM Sublibrary to a Controlled Disk Data Set

This Process copies a VSE sublibrary member from a BSAM sublibrary to a CA-DYNAM/D or CA-EPIC controlled disk data set on the same Sterling Connect:Direct node using PNODE=SNODE processing. All of the disk and tape data set names have been predefined to the appropriate system catalog. This Process was written with symbolic parameters to allow for a generic Process, so you must modify to your standards.

When you reference BSAM libraries in a Sterling Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

```
LIB2DYD1 PROC  SNODE=SC.VSE.NODE                                        -
               PNODE=SC.VSE.NODE                                        -
               &MEMBER=LIB2DYT1                                         -
               &VSEDSN=USER01.TEST.LIBCOPY1                             -
               &VSELIB=CONN.DIRECT.LIBRARIES                            -
               &VSESUB=USER01                                           -
               &VSETYP=N                                                -
               &VSEVOL=USER03
STEP0001 COPY  FROM ( SNODE                                             -
                    DSN=&VSELIB                                         -
                    DISP=SHR                                            -
                    VOL=SER=&VSEVOL                                     -
                    DCB=(DSORG=PS)                                      -
                    LIBR=(                                              -
                      SELMEM=&MEMBER                                    -
                      SELSLIB=&VSESUB                                   -
                      SELTYPE=&VSETYP)                                  -
                    )                                                   -
               TO   ( PNODE                                             -
                    DSN=&VSEDSN                                         -
                    UNIT=DLBLONLY                                       -
                    DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)               -
                    DISP=(NEW,CATLG)                                    -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('GOOD',&VSEDSN))                          -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('FAIL',&VSEDSN))                          -
                        PNODE
         EIF
```

Copy a VSE Sublibrary Member from a BSAM Sublibrary to a Controlled Tape Data Set

This member copies a VSE sublibrary member from a BSAM sublibrary to a CA-DYNAM/D or CA-EPIC controlled tape data set on the same Sterling Connect:Direct node using PNODE=SNODE processing.

All of the disk and tape data set names have been predefined to the appropriate system catalog. You do not have to specify output DCB parameters, these will be copied from the input library DCB parameters.

When you reference BSAM libraries in a Sterling Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

This Process was written with symbolic parameters to allow for a generic Process. You must modify to your standards.

```
LIB2DYT1 PROC  SNODE=SC.VSE.NODE                                        -
               PNODE=SC.VSE.NODE                                        -
               &MEMBER=LIB2DYT1                                         -
               &VSEDSN=USER01.TEST.TAPE1                                -
               &VSELIB=CONN.DIRECT.LIBRARIES                            -
               &VSESUB=USER01                                           -
               &VSETYP=JCL                                              -
               &VSEVOL=USER03
STEP0001 COPY  FROM ( SNODE                                             -
                    DSN=&VSELIB                                         -
                    DISP=SHR                                            -
                    VOL=SER=&VSEVOL                                     -
                    DCB=(DSORG=PS)                                      -
                    LIBR=(                                              -
                      SELMEM=&MEMBER                                    -
                      SELSLIB=&VSESUB                                   -
                      SELTYPE=&VSETYP)                                  -
                    )                                                   -
               TO   ( PNODE                                             -
                    DSN=&VSEDSN                                         -
                    UNIT=TNOASGN                                        -
                    LABEL=(1,SL)                                        -
                    DISP=(NEW,CATLG)                                    -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('GOOD',&VSEDSN))                          -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('FAIL',&VSEDSN))                          -
                        PNODE
         EIF
```

Copy a VSE/POWER LST Queue Member to a Controlled Disk Data Set

This Process extracts a VSE/POWER LST queue member (where: DSN=power.jobname) and places the data into a CA-DYNAM/D or CA-EPIC controlled disk data set. The disk data set has already been defined to the appropriate system catalog.

```
LST2DYD1 PROC  PNODE=SC.VSE.NODE                                        -
               SNODE=SC.VSE.NODE                                        -
               CLASS=5                                                  -
               &JBNAME=,                                                -
               &JBDISP=D                                                -
               &JBCLASS=A                                               -
               &VSEDSN=USER01.TEST.GDGPOWR1
STEP0001 COPY FROM ( PNODE                                              -
                    DSN=&JBNAME                                         -
                    LST=(CLASS=&JBCLASS DISP=&JBDISP)                   -
                    )                                                   -
             TO    ( SNODE                                              -
                    DSN=&VSEDSN                                         -
                    UNIT=DLBLONLY                                       -
*                   DCB=(DSORG=PS,RECFM=VM,LRECL=133,BLKSIZE=137)       -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                RUN TASK (PGM=DMNOTIFY,                                 -
                PARM=('GOOD',&VSEDSN))                                  -
                PNODE
         ELSE
                RUN TASK (PGM=DMNOTIFY,                                 -
                PARM=('FAIL',&VSEDSN))                                  -
                PNODE
         EIF
```

You do not need to specify an output DCB parameter. This information will be obtained from the LST queue entry. The Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing and uses symbolic values.

Copy a BSAM VSE Sublibrary to a New VSE BSAM Library

This Process sends an entire BSAM VSE sublibrary into a new VSE BSAM library to be allocated on the SNODE. This Process uses symbolic values. You must specify all of the shown below, on the FROM side to Process BSAM libraries.

When you reference BSAM libraries in a Sterling Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

When you copy data into a VSE BSAM library, you must add either RECFM=F or RECFM=V to your DCB parameter. This specification depends on the type of input file. If you do not include the RECFM, the Process fails with the message SVSJ122I.

```
LIB2LIB1 PROC  SNODE=SC.VSE.NODE                                        -
               PNODE=SC.VSE.NODE                                        -
               &NEWLIB=USER01.TEST.FBALIB01                             -
               &VSELIB=CONN.DIRECT.LIBRARIES                            -
               &VSESUB=USER01                                           -
               &VSETYP=*
STEP0001 COPY  FROM ( PNODE                                             -
                    DSN=&VSELIB                                         -
                    DISP=SHR                                            -
                    LIBR=(*)                                            -
                    VOL=SER=USER03                                      -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=80)                    -
                    )                                                   -
               TO   ( SNODE                                             -
                    DSN=&NEWLIB                                         -
                    UNIT=FBA                                            -
                    DISP=NEW LIBR=(*)                                   -
                    VOL=SER=FBA001                                      -
                    DCB=(DSORG=PS,RECFM=F)                              -
                    SPACE=(100,(4000),RLSE)                             -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('GOOD',&NEWLIB))                          -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('FAIL',&NEWLIB))                          -
                        PNODE
         EIF
```

Copy a BSAM VSE Sublibrary to a New z/OS PDS

This Process sends an entire BSAM VSE sublibrary into a new z/OS partitioned data set. The Process runs on the PNODE (VSE) and sends the data to the SNODE (z/OS). This Process uses symbolic values.

When you reference BSAM libraries in a Sterling Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

You must specify all of the parameters in this sample on the FROM side to Process BSAM libraries.

```
LIB2ZOS4 PROC  SNODE=SC.OS390.NODE                                          -
               PNODE=SC.VSE.NODE                                           -
               &OS390LIB=USER01.TEST.VSELIB                                -
               &VSELIB=CONN.DIRECT.LIBRARIES                               -
               &VSESUB=PROCESS                                             -
               &VSETYP=N
STEP0001 COPY  FROM ( PNODE                                                -
                    DSN=&VSELIB                                            -
                    DISP=SHR                                               -
                    VOL=SER=USER03                                         -
                    DCB=(DSORG=PS)                                         -
                    LIBR=(SELSLIB=&VSESUB                                  -
                          SELTYPE=&VSETYP                                  -
                          REPLACE=YES)                                     -
                    )                                                      -
               TO   ( SNODE                                                -
                    DSN=&OS390LIB                                          -
                    DISP=RPL                                               -
                    UNIT=SYSDA                                             -
                    SPACE=(CYL,(5,1,100),RLSE)                             -
                    DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=27920)         -
                    )                                                      -
                    COMPRESS
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                                -
                        PARM=('GOOD',&OS390LIB))                           -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                                -
                        PARM=('FAIL',&OS390LIB))                           -
                        PNODE
         EIF
```

Copy a MSAM Data Set to a Controlled BSAM Data Set (VSE)

Use this Process to copy a MSAM (VSAM Managed SAM) data set into a CA-DYNAM/D or CA-EPIC controlled BSAM data set.

```
MSM2DYD1 PROC PNODE=SC.VSE.NODE                                            -
              SNODE=SC.VSE.NODE                                            -
              &VSEDSN=USER01.TEST.GDGCOPY1
STEP0001 COPY  FROM ( PNODE                                                -
                    DSN=USER01.TEST.MSAMFIL1                               -
                    DISP=OLD                                               -
                    UNIT=DISK                                              -
                    VOL=SER=USER06                                         -
                    VSAMCAT=(VSE.COMMON.CATALOG,X,X,,123)                  -
                    DCB=(DSORG=MSAM,RECFM=FB,LRECL=80,BLKSIZE=16000)       -
                    )                                                      -
               TO   ( SNODE                                                -
                    DSN=&VSEDSN                                            -
                    DISP=NEW                                               -
                    UNIT=DLBLONLY                                          -
                    DCB=(DSORG=PS,RECFM=F,LRECL=80,BLKSIZE=16000)          -
                    )                                                      -
                    COMPRESS
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                                -
                        PARM=('GOOD',&VSEDSN))                             -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                                -
                        PARM=('FAIL',&VSEDSN))                             -
                        PNODE
         EIF
```

In the previous example, the disk data set has already been defined to the appropriate system catalog. This Process runs on the same Sterling Connect:Direct node using PNODE=SNODE processing. This Process uses symbolic values.

When you reference BSAM libraries in a Sterling Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

When you copy data into a VSE BSAM library, you must add either RECFM=F or RECFM=V to your DCB parameter. This specification depends on the type of input file. If you do not include the RECFM, the Process fails with the message SVSJ122I.

Copy a VSE VSAM to an i5/OS PDS Member

This Process copies a VSAM file from a VSE node to an i5/OS PDS member. The VSAM file resides on a catalog other than IJSYSUC, so a VSAMCAT parameter is coded.

```
PROC01  PROCESS      PNODE=SC.VSE.NODE1 SNODE=OS400.NODE2   -
                     SNODEID=(OS400ND,PWD400)
STEP01  COPY  FROM  (DSN=VSE.TEST.VSAM                      -
                     DCB=(DSORG=VSAM)                       -
                     VSAMCAT=(VSESP.USER.CATALOG,1,1,,111)  -
                     DISP=SHR)                              -
              TO    (DSN='OS400X/PDSLIB(TSTDATA)'           -
                     SYSOPTS=\"TYPE(MBR)\                   -
                             \RCDLEN(100)\                  -
                             \FILETYPE(*DATA)"\             -
                     DISP=RPL                               -
                     SNODE)
```

Copy a VSE VSAM File to an i5/OS Spooled File

This Process copies a VSAM file from a VSE node to an i5/OS spooled file. Page size is optional and dependent on the printer device.

```
PROC01  PROCESS      PNODE=SC.VSE.NODE1 SNODE=OS400.NODE2  -
                     SNODEID=(OS400ND,PWD400)
STEP01  COPY  FROM  (DSN=VSE.TEST.VSAM                     -
                     DCB=(DSORG=VSAM)                      -
                     DISP=SHR)                             -
              TO    (DSN='DATAFF'                         -
                     SYSOPTS=\"TYPE(SPLF)\                 -
                             \CTLCHAR(*NONE)\              -
                             \PAGESIZE(66 378)\            -
                             \SPOOL(*YES)"\                -
                     SNODE)
```

Copy a VSE Librarian BSAM Member to a Preallocated z/OS PDS Member

This Process copies a VSE Librarian BSAM member into a preallocated z/OS partitioned data set (PDS) member. The disk data set has already been defined to the appropriate system catalog. This Process was written with symbolics for substitution.

When you reference BSAM libraries in a Sterling Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

```
LIB2ZOS1 PROC  PNODE=SC.VSE.NODE                                        -
               SNODE=SC.OS390.NODE                                      -
               &OS390LIB=USER01.TEST.VSEPROC                            -
               &VSELIB=CONN.DIRECT.LIBRARIES                            -
               &VSEMEM=,                                                -
               &VSESUB=USER01                                           -
               &VSETYP=N                                                -
               &VSEVOL=USER03
STEP0001 COPY  FROM ( PNODE                                             -
                    DSN=&VSELIB                                         -
                    DISP=SHR                                            -
                    UNIT=DISK                                           -
                    DCB=(DSORG=PS)                                      -
                    VOL=SER=&VSEVOL                                     -
                    LIBR=(SELMEM=&VSEMEM                                -
                          SELSLIB=&VSESUB                               -
                          SELTYPE=&VSETYP)                              -
                    )                                                   -
               TO   ( SNODE                                             -
                    DSN=&OS390LIB                                       -
                    DISP=SHR                                            -
                    )
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('GOOD',&OS390LIB))                        -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('FAIL',&OS390LIB))                        -
                        PNODE
        EIF
```

Copy a VSAM VSE Library Member to a Preallocated z/OS PDS Member

Use the Process to copy a VSAM VSE Library member into a preallocated z/OS partitioned data set (PDS) member.

The disk data set has already been defined to the appropriate system catalog. This Process was written with symbolics for substitution. When referencing VSAM libraries in a Sterling Connect:Direct Process you must specify the DSORG and DISP parameters.

```
LIB2ZOS2 PROC  PNODE=SC.VSE.NODE                                        -
               SNODE=SC.OS390.NODE                                      -
               &MEMBER=DITTODVT                                         -
               &OS390LIB=USER01.PROCESS.LIB                             -
               &VSELIB=CONN.DIRECT.LIB1                                 -
               &VSESUB=RXUSER01                                         -
               &VSETYP=JCL
STEP0001 COPY  FROM ( PNODE                                             -
                    DSN=&VSELIB                                         -
                    DISP=SHR                                            -
                    DCB=(DSORG=VSAM)                                    -
                    LIBR=(SELMEM=&MEMBER                                -
                          SELTYPE=&VSETYP                               -
                          SELSLIB=&VSESUB                               -
                          REPLACE=YES)                                  -
                    )                                                   -
               TO   ( SNODE                                             -
                    DSN=&OS390LIB                                       -
                    DISP=SHR                                            -
                    )                                                   -
               COMPRESS EXT
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('GOOD',&OS390LIB))                        -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('FAIL',&OS390LIB))                        -
                        PNODE
         EIF
```

Copy a VSE/POWER LST Queue Member to a Preallocated z/OS PDS

Use this Process to copy a VSE/POWER LST queue member into a preallocated z/OS partitioned data set (PDS). The member name is submitted when the Process is submitted by overriding the symbolic &PINUMB.

The disk data set has already been defined to the appropriate system catalog. Verify that your job class in the LST queue matches the Process job class (&JBCLASS); otherwise the Process will end and not copy the data set. This Process uses symbolic values.

```
LST2ZOS1 PROC  PNODE=SC.VSE.NODE                                        -
               SNODE=SC.OS390.NODE                                      -
               CLASS=8                                                  -
               &JBNAME=GGG3200                                          -
               &JBNUMB=0000                                             -
               &JBDISP=L                                                -
               &JBCLASS=Q                                               -
               &PINUMB=
STEP0001 COPY FROM ( PNODE                                              -
                   DSN=&JBNAME                                          -
                   LST=(CLASS=&JBCLASS,DISP=&JBDISP)                    -
                   )                                                    -
              TO   ( SNODE                                              -
                   DSN=USER01.TEST.VSEDUMPS(&PINUMB)                    -
                   DISP=RPL                                             -
                   )                                                    -
                   COMPRESS
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('GOOD',&PINUMB))                          -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('FAIL',&PINUMB))                          -
                        PNODE
         EIF
```

Copy a File from UNIX HP to a Controlled Disk Data Set on VSE Using LU6.2

Use this Process to copy a file from UNIX HP into a CA-DYNAM/D or CA-EPIC controlled disk data set using the LU 6.2 protocol. The disk data set has already been defined to the appropriate system catalog. DCB information will be provided by the SNODE. This Process uses symbolic values.

```
UNX2DYD1 PROCESS PNODE=SC.VSE.USER01                                    -
                 SNODE=SC.UNIX.NODE                                     -
                 &VSEDSN=USER01.TEST.UNIXFILE
STEP0001 COPY    TO    ( PNODE                                          -
                       DSN=&VSEDSN                                      -
                       UNIT=DLBLONLY                                    -
                       )                                                -
                 FROM  ( SNODE                                          -
                       DSN='/home/fremont/ddunc1/750070/arx02.dat'      -
                       SYSOPTS=":xlate=no:strip.blanks=no:"             -
                       )                                                -
                 CKPT=1M                                                -
                 COMPRESS
STEP0002 IF    (STEP0001 EQ 0) THEN
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('GOOD',&VSEDSN))                          -
                        PNODE
         ELSE
                    RUN TASK (PGM=DMNOTIFY,                             -
                        PARM=('FAIL',&VSEDSN))                          -
                        PNODE
         EIF
```

Copy a File from HP UNIX to a VSE Controlled Disk Data Set Using TCP/IP

This Process copies a file from HP UNIX into a CA-DYNAM/D or CA-EPIC controlled disk data set using the TCP/IP protocol.

The disk data set has already been defined to the appropriate system catalog. Verify that you have updated your network map with the SNODE TCP/IP address and port number. DCB information will be provided by the SNODE. This Process was written with symbolics for substitution.

```
UNX2DYD2 PROCESS PNODE=SC.VSE.USER01                                    -
                 SNODE=199.1.4.87                                       -
                 &VSEDSN=USER01.TEST.UNIXFILE
STEP0001 COPY    TO    ( PNODE                                          -
                       DSN=&VSEDSN                                      -
                       UNIT=DLBLONLY                                    -
                       )                                                -
                 FROM  ( SNODE                                          -
                       DSN='/home/fremont/ddunc1/750070/arx02.dat'      -
                       SYSOPTS=":xlate=no:strip.blanks=no:"             -
                       )                                                -
                 CKPT=1M                                                -
                 COMPRESS
STEP0002 IF    (STEP0001 EQ 0) THEN
                     RUN TASK (PGM=DMNOTIFY,                            -
                         PARM=('GOOD',&VSEDSN))                         -
                         PNODE
         ELSE
                     RUN TASK (PGM=DMNOTIFY,                            -
                         PARM=('FAIL',&VSEDSN))                         -
                         PNODE
         EIF
```

Copy a DBCS Data Set from VSE to UNIX Using the KSCXEBC Translation Table

This COPY statement copies a data set from a VSE node to a UNIX node using the translation table KSCXEBC. Required parameters for this translation are in bold print.

```
/*********************************************************************/
/*          HOST to UNIX DBCS translation using table KSCXEBC        */
/*********************************************************************/
STEP02  COPY                                                        -
              TO    (PNODE                                          -
                     DSN='hlq.HOSTFILE'                             -
                     SYSOPTS="DBCS=KSCXEBC"                         -
                     DISP=(RPL,CATLG)                               -
                     UNIT=SYSDA                                     -
                     DCB=(RECFM=VB,LRECL=254,BLKSIZE=4096,DSORG=PS) -
                     SPACE=(254, (1000,100))                        -
                    )                                               -
              FROM  (SNODE                                          -
                     DSN='/unixfile'                                -
                     SYSOPTS=":xlate=no:strip.blanks=no:"           -
                     DISP=SHR                                       -
                    )
```

- The copy step is named STEP02.
- The SYSOPTS attribute specified in the TO clause of the COPY statement is used to define the default translation table KSCXEBC.

- The DCB attributes specified on the TO clause of the COPY statement are used for file allocation.
- Unit is specified on the PNODE.
- The SYSOPTS parameter on the FROM clause of the COPY statement is required.

Copy a DBCS Data Set from Microsoft Windows to VSE Using the KSCXEBC Translation Table

The following PC-to-host DBCS translation uses the supplied translation table KSCXEBC when copying data set from a PC to a host Sterling Connect:Direct for VSE node. Required parameters for this translation are in bold print.

```
/*********************************************************************/
/*           PC to HOST DBCS translation using table KSCXEBC         */
/*********************************************************************/
PCTOHOST  PROCESS  SNODE=HOSTNODE                                    -
                   HOLD=CALL
STEP01    COPY                                                       -
                TO    (PNODE                                         -
                       DSN='hlq.PCFILE'                              -
                       DISP=(RPL,CATLG)                              -
                       UNIT=SYSDA                                    -
                       DCB=(RECFM=VB,LRECL=254,BLKSIZE=4096,DSORG=PS) -
                       SPACE=(254,(1000,100))                        -
                       SYSOPTS="DBCS=KSCXEBC"                        -
                      )                                              -
                FROM  (SNODE                                         -
                       DSN=PCFILE                                    -
                       DISP=SHR                                      -
                      )
```

- The copy step is named STEP01.
- The input data set is cataloged after successful completion of the Process.
- The DCB attributes specified in the TO clause of the COPY statement are used for file allocation.
- The SYSOPTS attribute specified in the TO clause of the COPY statement is used to define the supplied translation table KSCXEBC.
- UNIT has been specified on the PNODE only.

Copy a DBCS Data Set from VSE to Microsoft Windows Using the EBCXKSC Translation Table

This COPY statement copies a data set from a host Sterling Connect:Direct for VSE to a PC node using the translation table EBCXKSC. Required parameters for this translation are in bold print.

```
/*****************************************************************/
/*           HOST to PC DBCS translation using table EBCXKSC     */
/*****************************************************************/
HOSTTOPC  PROCESS  SNODE=PCNODE                -
                   HOLD=CALL
STEP01    COPY     FROM  (PNODE                -
                          DSN='hlq.HOSTFILE'    -
                          SYSOPTS="DBCS=EBCXKSC" -
                          DISP=(SHR)            -
                         )                      -
                   TO    (SNODE                -
                          DSN=PCFILE            -
                          DISP=RPL              -
                         )
```

- The copy step is named STEP01.
- The SYSOPTS attribute is specified in the FROM clause of the COPY statement is used to define the default translation table EBCXKSC.

Copy a DBCS Data Set from UNIX to VSE Using the EBCXKSC Translation Table

This COPY statement copies a data set from a UNIX to a host Sterling Connect:Direct for VSE node using the translation table EBCXKSC. Required parameters for this translation are in bold print.

```
/*****************************************************************/
/*           UNIX to HOST DBCS translation using table EBCXKSC    */
/*****************************************************************/
STEP01  COPY                                                -
              FROM  (PNODE                                  -
                     DSN='hlq.UNIXFILE'                     -
                     SYSOPTS="DBCS=EBCXKSC"                 -
                     DISP=(SHR)                             -
                    )                                       -
              TO    (SNODE                                  -
                     DSN='/unixfile'                        -
                     SYSOPTS=":xlate=no:strip.blanks=no:"   -
                     DISP=RPL                               -
                    )
```

- The copy step is named STEP01.
- The SYSOPTS attribute specified in the TO clause of the COPY statement is used to define the default translation table EBCXKSC.
- The SYSOPTS parameter on the FROM clause of the COPY statement is required.

Copy a Data Set from a VSE Node to Another VSE Node

This example COPY statement copies a data set from one Sterling Connect:Direct for VSE node to another.

```
STEP1  COPY  FROM  (DSN=SRCDATA.SET          -
                   DISP=(SHR)               -
                   PNODE                    -
                   DCB=(DSORG=PS,LRECL=80,  -
                   RECFM=FB,BLKSIZE=3120)   -
                   UNIT=3380                -
                   VOL=SER=VOL003           -
                  )                         -
                   COMPRESS                 -
             TO    (                        -
                   DSN=DESTDATA.SET         -
                   DCB=(DSORG=PS,LRECL=80,  -
                   RECFM=FB,BLKSIZE=3120)   -
                   DISP=(NEW)               -
                   UNIT=242                 -
                   SPACE=(600,(300))        -
                   SNODE                    -
                  )
```

- The COPY step is named STEP1.
- The DCB attributes specified in the TO clause of the COPY statement are used for file allocation.
- Unit and volume serial number are specified on the PNODE; however, only unit is specified on the SNODE.
- Specifying COMPRESS without a subparameter indicates that blanks will be compressed during transmission and converted back to the original string during decompression.
- Space parameters for the new TO data set are explicitly specified. This will allocate the new file on track 600 of unit 242 for a length of 300 tracks.

Use SYSOPTS for DBCS in VSE

The SYSOPTS statement declares that a Process is transferring a DBCS file. Include this statement on the host node COPY statement.

Support for multiple transfers with multiple translation tables is possible. All Processes support compression and checkpointing.

File transfer with double-byte character set (DBCS) is only supported in record mode. It is not supported in block mode. The following example uses the table name EBCXKSC and the default values x'0E' for **so**, and x'0F' for **si**.

```
SYSOPTS="DBCS=(EBCXKSC,0E,0F)"
```

(Transfers attempted in block mode can produce unpredictable results in the destination file. These results can compromise data integrity. You do not receive an error message in these cases.)

The following example uses the table name KSCXEBC and the default values x'0E' for **so**, and x'0F' for **si**.

```
SYSOPTS="DBCS=(KSCXEBC,0E,0F)"
```

The following example uses the table name EBCXKSC and the NOSO value x'00' for **so** and **si**.

```
SYSOPTS="DBCS=(EBCXKSC,00,00)"
```

The following example uses the table name EBCXKSC and takes the defaults for **so** and **si**.

```
SYSOPTS="DBCS=(EBCXKSC)"
```

The following example uses the table name USERTAB and takes the defaults for **so** and **si**. USERTAB is a user-defined, customized translation table.

```
SYSOPTS="DBCS=USERTAB"
```

z/OS

Use Defaults to Allocate a File (z/OS to z/OS)

This Process designates that the destination file is placed on a specified unit and volume. The SPACE information is also supplied. Because DCB information is not specified for the destination data set, DCB information from the source data set is used.

Both checkpointing and compression are requested. If the Process receives an X37-type abend, the Process is requeued, because REQUEUE=YES is specified in the PROCESS statement. Corrective action can then be taken.

```
COPYSEQ  PROCESS       PNODE=CD.DALLAS                  -
                       SNODE=CD.CHICAGO REQUEUE=YES
STEP01   COPY   FROM  (PNODE DSN=$DAL.PSDATA)           -
                        CKPT=10M                        -
                        COMPRESS                        -
                TO    (SNODE DSN=$CHI.PSDATA            -
                        DISP=(NEW,CATLG)                -
                        UNIT=3380                       -
                        VOL=(SER=SYS009)                -
                        SPACE=(4096,(200,40),,,ROUND))
```

Use %SUBDATE and %SUBTIME for a File Name (z/OS to z/OS)

This Process shows how to use the %SUBDATE and %SUBTIME variables to construct a unique dataset name based on the date and time of a file transfer submission.

%JDATE, %SUBDATE, and %SUBTIME are exclusive to Sterling Connect:Direct for z/OS.

```
CD1     PROCESS SNODE=CD.VM.NODE                                          -
                &amp;DATE=%SUBDATE                                        -
                &amp;TIME=%SUBTIME
STEP01  COPY  FROM  (                                                     -
                     DSN=ACCTNG.DAILY.DATA                                -
                     SELECT=(PL*)                                         -
                    )                                                     -
              TO    (                                                     -
                     DSN=ACCTNG.UPDATES.D || &amp;DATE || .T || &amp;TIME  -
                     DISP=(NEW,CATLG)                                     -
                     DCB=(DSORG=PO,LRECL=80,RECFM=FB,BLKSIZE=8000)        -
                    )
```

File Allocation Using a TYPE File (z/OS to z/OS)

The following Process uses a TYPE file to allocate a data set; therefore, UNIT, VOLUME, and SPACE parameters are not specified within the COPY statement.

The destination data set is allocated using definitions specified in the TYPE record, PSFILE.

```
COPYSEQ PROCESS      PNODE=CD.DALLAS                -
                     SNODE=CD.CHICAGO REQUEUE=YES
STEP01  COPY  FROM  (DSN=DAL.PSDATA)                -
                     CKPT=10M                       -
                     COMPRESS                       -
              TO    (DSN=CHI.PSDATA                -
                     TYPE=PSFILE)
```

The following parameters make up the TYPE record, PSFILE. This TYPE record must be present at the destination node (SNODE).

```
   DISP=(NEW,CATLG)                       -
   DCB=(BLKSIZE=3120,LRECL=80,DSORG=PS)   -
   UNIT=3380                              -
   VOL=(SER=SYS009)                       -
   SPACE=(4096,(200,40),,,ROUND)
```

You can also place the IOEXIT parameter in a TYPE entry. Any parameters specified on the COPY statement take precedence over those coded in the TYPE file. See the appropriate Sterling Connect:Direct platform's documentation for details on setting up entries in the TYPE file.

Override Secure Plus Settings in an STS Protocol Environment (z/OS)

The following sample Process, SAMPLE, copies the data set TEST.INPUT.DATASET from the PNODE to the SNODE (THE.OTHER.NODE), renames it to TEST.OUTPUT.DATASET, and enables data encryption and digital signatures.

```
SAMPLE    PROCESS SNODE=THE.OTHER.NODE
*
COPYFILE  COPY    FROM ( PNODE                      -
                         DSN='TEST.INPUT.DATASET'   -
                         DISP=SHR                   -
                       )                            -
                  TO   ( SNODE                      -
                         'TEST.OUTPUT.DATASET'      -
                          DISP=(NEW.CATLG)          -
                       )                            -
                          SECURE=(ENC=Y,SIG=Y)
```

Copy a KSDS that Must be Extended

This COPY statement copies and sends a basic key-sequenced data set (KSDS) to the secondary node as a new KSDS that is required to be extended. Because the LIKE parameter is not included, the KEYLEN, KEYOFF, AND LRECL parameters had to be specified.

```
COPY FROM (PNODE DSN=EPETE1.KSDS    DISP=(SHR)   )     -
     TO   (SNODE DSN=EPETE2.KSDSE DISP=(NEW,CATLG)     -
           DSNTYPE=EXTREQ KEYLEN=8 KEYOFF=6 LRECL=80)
```

Copy a SAM File (z/OS to z/OS)

The following Process copies a SAM file from CD.DALLAS to a new file located at another site, CD.CHICAGO. The receiving node allocates the file using the same file attributes as the source data set because no file information is supplied. Both checkpointing and compression are requested within the Process. The submitting user is notified when the Process completes.

```
COPYSEQ PROCESS    PNODE=CD.DALLAS  NOTIFY=%USER  -
                   SNODE=CD.CHICAGO
STEP01  COPY FROM (DSN=DAL.PSDATA)                -
                   CKPT=10M                       -
                   DISP=SHR                       -
                   COMPRESS                       -
             TO   (DSN=CHI.PSDATA)                -
                   DISP=RPL
```

Copy a DFDSS Volume Dump (z/OS to z/OS)

This Process transmits a z/OS Data Facility Data Set Services (DFDSS) volume dump to tape. Note that the RECFM must be U.

```
DFDSS    PROCESS      SNODE=CD.OS390.ALTO
COPY1    /* STEP NAME  */                                -
         COPY  FROM  (DSN=DAT.P34ECH.DEC91                -
                      DISP=OLD  DCB=(RECFM=U, BLKSIZE=32760))  -
                      COMPRESS                          -
               TO    (DSN=DAT.P44ECH.DEC91              -
                      DISP=(NEW,CATLG) LABEL=(,SL,,,RETPD=15)  -
                      DCB=(RECFM=U, BLKSIZE=32760)      -
                      UNIT=CART SNODE)
```

Copy an Entire PDS (z/OS to z/OS)

This COPY statement transmits a PDS to a new file located at another site. Because allocation data is not included in the Process, the receiving node allocates the file

using the same file attributes as the source data set. Both checkpointing and compression are requested within the Process. Checkpointing only takes place at the member level when copying a PDS.

```
COPY  FROM  (DSN=PDS.SOURCE)   -
             CKPT=1M           -
             DISP=SHR          -
             COMPRESS          -
      TO    (DSN=PDS.DEST)     -
             DISP=RPL
```

Specify a Range and the NR Subparameter to Copy Selected PDS Members (z/OS to z/OS)

The following COPY statement shows how the NR (NOREPLACE) subparameter can be used when a range is specified. The member ABC is copied to the PDS, PDS.DEST. All members from FAA through GBB are copied if they do not already exist. The member PQR is also copied.

```
/*   USE OF NR WHEN A RANGE IS SPECIFIED   */
COPY FROM (DSN=PDS.SOURCE  -
           SELECT=(ABC,    -
           (FAA/GBB,,NR),  -
           PQR))           -
     TO   (DSN=PDS.DEST    -
           DISP=OLD)
```

Copy One Member of a PDS (z/OS to z/OS)

This example illustrates three ways to copy a single member of a PDS.

- In STEP01, the FROM and TO member names are specified in the data set names.
- In STEP02, the FROM and TO member names are specified in the data set names, with the output member name changed.
- STEP03 uses the SELECT parameter as part of the FROM clause of the COPY statement, with a new member name specified on the data set name for the destination data set.

```
STEP01  COPY  FROM  (DSN=PDS.SOURCE(MEMBER))   -
              TO    (DSN=PDS.DEST(MEMBER)      -
                     DISP=RPL)
STEP02  COPY  FROM  (DSN=PDS.SOURCE(OLDMEM))   -
              TO    (DSN=PDS.DEST(NEWNAME)     -
                     DISP=RPL)
STEP03  COPY  FROM  (DSN=PDS.SOURCE            -
                     SELECT=OLDMEM)            -
              TO    (DSN=PDS.DEST(NEWNAME)     -
                     DISP=RPL)
```

Copy a PDS and Excluding an Individual Member (z/OS to z/OS)

This COPY statement copies an entire PDS, with the exception of the member, MEM3, named in the EXCLUDE parameter.

```
COPY FROM  (DSN=PDS.SOURCE  -
            EXCLUDE=MEM3)   -
     TO    (DSN=PDS.DEST)
```

Copy a PDS and Excluding Members Generically (z/OS to z/OS)

This COPY statement shows how to exclude members generically. In this example, all members are copied, except members with names beginning with ABC.

```
COPY  FROM  (DSN=PDS.SOURCE  -
             EXCLUDE=ABC*)   -
      TO    (DSN=PDS.DEST)
```

Copy a PDS and Using a Range to Exclude PDS Members (z/OS to z/OS)

The following COPY statement copies an entire PDS but uses a range to exclude any members from AAB through BBC.

```
COPY  FROM  (DSN=PDS.SOURCE        -
             EXCLUDE=((AAB/BBC)))  -
      TO    (DSN=PDS.DEST)
```

The source file, PDS.SOURCE, contains the following members:

```
AAA, AAB, ABB, ABC, BBA, BBB, BBCA, BGG, XXX
```

Members AAA, BBCA, BGG, and XXX are copied. Members AAB, ABB, ABC, BBA, and BBB are not copied because they are within the range of members excluded in AAB/BBC. There is no member BBC to exclude; BBB was the last member in PDS.SOURCE within the specified range. Because BBCA is not part of the range to be excluded, it is copied. A generic range, like AAB*/BBC*, is not valid.

All range entries must be specified as subparameters and enclosed in parentheses.

Copy a PDS and Generically Selecting Members (z/OS to z/OS)

This COPY statement shows how to select generically all member names starting with PAID (the first four characters of the data set names). The remaining members of PDS.SOURCE are not copied.

```
COPY  FROM  (DSN=PDS.SOURCE   -
             SELECT=(PAID*))  -
      TO    (DSN=PDS.DEST)
```

Copy PDS Members Using the EXCLUDE and SELECT Parameters (z/OS to z/OS)

In this example, the data set, PDS.SOURCE, contains the following members:

```
A, AB, ABA, ABB, ABC, ABD, ABE, ACB, ACC, ACD, BAA, BAB, BAC, CDE, CDF
```

This COPY statement shows the use of the SELECT and EXCLUDE parameters:

```
COPY  FROM  (DSN=PDS.SOURCE                      -
             SELECT=((BA/BBB),A*,ABC,(CDF,ZZZ))  -
             EXCLUDE=(AC*,BAA,(ABC/AZ)))         -
      TO    (DSN=PDS.DEST)
```

Results are as follows:

- A, AB, ABA, ABB are copied because they are generically selected by A*.
- ABC is copied because a specific ABC selection overrides the EXCLUDE range (ABC/AZ).
- ABD and ABE are not copied because they are excluded by the range ABC/AZ.
- ACB, ACC, and ACD are not copied because they are excluded generically by AC*.
- BAA is not copied because the specific BAA EXCLUDE overrides the SELECT range (BA/BBB).
- BAB and BAC are copied because they are selected by the range (BA/BBB).
- CDE is not copied because it is not selected.
- CDF is copied because it is specifically selected by (CDF,ZZZ); ZZZ is the new name.

Copy a PDS Using the ALIAS Parameter with SELECT and EXCLUDE (z/OS to z/OS)

This COPY statement shows how the ALIAS parameter works when other optional parameters are specified.

```
COPY   FROM  (DSN=PDS.SOURCE  -
              ALIAS=Y         -
              EXCLUDE=C3      -
              SELECT=(A,C1))  -
       TO    (DSN=PDS.DEST)
```

In this example, the data set PDS.SOURCE has the following members and associated aliases:

Members	**Aliases**
A	A1 A2
B	
C	C1 C2 C3

The data set PDS.DEST contained no members and no aliases before the COPY operation. After the COPY operation, PDS.DEST contains the following members and aliases:

Members	**Aliases**
A	A1 A2
C	C1 C2

The explanation follows:

- A is copied because it was selected by member name.
- A1 is copied because ALIAS=Y and A1 is an alias for member A.

- A2 is copied because ALIAS=Y and A2 is an alias for member A.
- B is not copied because it was not selected.
- C is copied because ALIAS=Y and C is the true member name for C1.
- C1 is copied because it was selected by member name.
- C2 is copied because ALIAS=Y and C1 (another alias for member C) was selected by member name.
- C3 is not copied because it was specifically excluded.

Copy a PDS Member to Tape (z/OS to z/OS)

The following Process copies a PDS member to a tape device, specifying the PDS member as part of the DSN. Because you can only copy a single PDS member to tape in one copy step, DSORG=PS must be coded on the TO clause of the Sterling Connect:Direct for z/OS COPY statement.

```
COPYSEQ  PROCESS  PNODE=CHICAGO                  -
                  SNODE=MINNEAPOLIS
STEP01   COPY     FROM  (DSN=MY.PDS.FILE(MEMBER)  -
                         DISP=SHR PNODE)          -
                  TO    (DSN=TAPE.FILE            -
                         DISP=(NEW,CATLG)         -
                         UNIT=TAPE                -
                         LABEL=(,SL)              -
                         DCB=(DSORG=PS))
```

You can copy multiple PDS members by coding multiple COPY steps in a Process, and either append the additional PDS members to the first file or create separate tape files for each member. In the following example, MEMB is appended to TAPE.FILE.

```
COPYSEQ  PROCESS  PNODE=CHICAGO                  -
                  SNODE=MINNEAPOLIS
STEP01   COPY     FROM  (DSN=MY.PDS.FILE(MEMA)   -
                         DISP=SHR PNODE)          -
                  TO    (DSN=TAPE.FILE           -
                         DISP=(NEW,CATLG)         -
                         UNIT=TAPE                -
                         DCB=(DSORG=PS))
STEP02   COPY     FROM  (DSN=MY.PDS.FILE(MEMB)   -
                         DISP=SHR PNODE)          -
                  TO    (DSN=TAPE.FILE           -
                         DISP=(MOD,CATLG)         -
                         UNIT=TAPE                -
                         LABEL=(,SL)              -
                         DCB=(DSORG=PS))
```

Use the IOEXIT Parameter (z/OS to z/OS)

You can specify use of an I/O exit by the inclusion of the IOEXIT keyword on the COPY statement. The IOEXIT keyword is valid in either the FROM or TO clause of the COPY statement. You can specify a different user-written IOEXIT on each side as shown in the following example. The exit must, however, reside on the node where it is referenced.

The exit referenced in this COPY (OUEXT03) must reside in an authorized loadlib at the destination site.

In this example, INEXT01, an IOEXIT program, on the source Sterling Connect:Direct node will be invoked and passed two parameters-a character string ('DB0A05') and a hexadecimal value (X'0E'). It will pass records using Sterling Connect:Direct to OUEXT03, an IOEXIT on the destination Sterling Connect:Direct node.

```
COPY  FROM  (PNODE                              -
             IOEXIT=(INEXT01,C'DB0A05',X'0E'))  -
      TO    (SNODE                              -
             IOEXIT=OUEXT03)
```

Copy to a New SMS-Controlled Data Set (z/OS to z/OS)

This COPY statement transmits a sequential file to a new SMS-controlled file at another site. The new data set will be allocated using the DCB parameters that are specified within the DATACLAS definition on the receiving node.

```
COPY  FROM  (DSN=SEQ.SOURCE)    -
      TO    (DSN=SEQ.DEST       -
             DATACLAS=DCLASS1   -
             STORCLAS=SCLASS1   -
             MGMTCLAS=MCLASS1)
```

Copy to a New SMS Data Set Using LIKE (z/OS to z/OS)

This COPY statement transmits a sequential file to a new SMS-controlled file at another site. The new data set will be allocated using the DCB parameters from the data set specified on the LIKE parameter.

```
COPY  FROM  (DSN=SEQ.SOURCE)          -
      TO    (DSN=SEQ.DEST             -
             LIKE=MODEL.DATASET.DEST  -
             STORCLAS=SCLASS1         -
             MGMTCLAS=MCLASS1)
```

Create and Copy a PDSE Data Set (z/OS to z/OS)

This COPY statement transmits a PDS file to a new PDSE file at another site. The DSNTYPE parameter is used to specify that the receiving data set is a LIBRARY (another name for a PDSE file). The data set is also allocated with the STORCLAS parameter to ensure that the SMS controls the data set.

```
COPY  FROM  (DSN=PDS.SOURCE)    -
      TO    (DSN=PDSE.DEST      -
             DSNTYPE=LIBRARY    -
             DATACLAS=DCLASS1   -
             STORCLAS=SCLASS1)
```

Create and Copy a VSAM KSDS Data Set (z/OS to z/OS)

This COPY statement transmits a VSAM KSDS file to a new VSAM KSDS at another site. The KEYLEN parameter is used to specify the length of the key for the data set. The RECORG parameter specifies that the output VSAM data set is a KSDS. The LRECL parameter specifies the maximum length of a record in the VSAM data set. The KEYOFF parameter specifies the offset to the key within each record. Note that the offset is relative to **0**.

```
COPY  FROM  (DSN=VSAM.KSDS.SOURCE)  -
      TO    (DSN=VSAM.KSDS.DEST    -
             DISP=(NEW,CATLG)      -
             RECORG=KS             -
             KEYLEN=16             -
             KEYOFF=20             -
             LRECL=256             -
             STORCLAS=SCLASS1      -
             MGMTCLAS=MCLASS1)
```

Create and Copy a VSAM ESDS Data Set (z/OS to z/OS)

This COPY statement transmits a VSAM ESDS file to a new VSAM ESDS at another site. The RECORG parameter specifies that the target VSAM data set is an ESDS. The LRECL parameter specifies the maximum length of a record in the VSAM data set.

```
COPY  FROM  (DSN=VSAM.ESDS.SOURCE)  -
      TO    (DSN=VSAM.ESDS.DEST    -
             DISP=(NEW,CATLG)      -
             RECORG=ES             -
             LRECL=128             -
             STORCLAS=SCLASS1      -
             MGMTCLAS=MCLASS1)
```

Create and Copy a VSAM RRDS Data Set (z/OS to z/OS)

This COPY statement transmits a VSAM RRDS file to a new VSAM RRDS at another site. The RECORG parameter specifies that the target VSAM data set is an RRDS. The LRECL parameter specifies the maximum length of a record in the VSAM data set.

```
COPY  FROM   (DSN=VSAM.RRDS.SOURCE)  -
      TO     (DSN=VSAM.RRDS.DEST    -
              DISP=(NEW,CATLG)      -
              RECORG=RR             -
              LRECL=300             -
              STORCLAS=SCLASS1      -
              MGMTCLAS=MCLASS1)
```

Create and Copy a VSAM Linear Data Set (z/OS to z/OS)

This COPY statement transmits a VSAM LINEAR file to a new VSAM LINEAR file at another site. The RECORG parameter specifies that the target VSAM data set is a LINEAR file.

```
COPY  FROM  (DSN=VSAM.LINEAR.SOURCE)  -
      TO    (DSN=VSAM.LINEAR.DEST    -
             DISP=(NEW,CATLG)        -
             RECORG=LS               -
             STORCLAS=SCLASS1        -
             MGMTCLAS=MCLASS1)
```

Copy a Data Set with a Security Profile (z/OS to z/OS)

This COPY statement transmits a sequential file to a new sequential file at another site. The security profile of the model data set is used as a model to create a generic security profile for the new data set.

```
COPY  FROM  (DSN=SEQ.SOURCE)                        -
      TO    (DSN=SEQ.DEST                           -
             DISP=(NEW,CATLG)                       -
             SECMODEL=(SEQ.DATASET.DEST,GENERIC)    -
             STORCLAS=SCLASS1                       -
             MGMTCLAS=MCLASS1)
```

Copy to z/OS Nodes with Unique Member Name Allocation (AXUNIQ Exit)

The following examples demonstrate how Sterling Connect:Direct resolves member names when UNIQUE=YES is specified on the SYSOPTS parameter of the COPY TO statement.

The member name is made unique as follows:

- The member name specified in the TO DSN (or defaulted from the FROM DSN) is used as a seed for comparison. If the seed name is not unique on the receiving node, Sterling Connect:Direct will modify the specified member name to create a unique name.
- If the seed name is less than eight characters long, Sterling Connect:Direct appends a unique numeric character to the member name, starting with 1. If the member name formed by the seed and the suffix exists already, the suffix is incremented until a unique name is created.
- The suffix appended can be as long as seven digits.
- If the seed name is eight characters long, Sterling Connect:Direct will truncate the name from the right to limit the name to eight characters. This truncation will also take place if the seed name and its appended suffix are more than eight characters long.

Resolution of a Unique Member Name by Appending a Digit

This example assumes that HLQ.PDS already contains the members DATA, DATA1, DATA2, and DATA11. Because the member name in the example, DATA, already exists on the z/OS TO node, a different member name will have to be created. The AXUNIQ exit, invoked by SYSOPTS="UNIQUE=YES" does this for you by adding numeric digits to the specified member name until a unique member name is achieved.

```
COPY  FROM  (DSN=\app101\data)                        -
      TO    (DSN=HLQ.PDS(DATA) SYSOPTS="UNIQUE=YES")
```

Sterling Connect:Direct resolves the member name to DATA3.

Resolution of a Unique Member Name by Truncating and Appending a Digit

This example assumes that HLQ.PDS already contains the member DATEABLE. Because the member name in the example, DATAFILE, already exists on the z/OS TO node, a different member name will have to be specified. The AXUNIQ exit, invoked by SYSOPTS="UNIQUE=YES", does this for you by dropping the rightmost byte and adding a numeric digit to the specified member name until a unique member name is achieved.

```
COPY  FROM  (DSN=\app101\data)                            -
      TO    (DSN=HLQ.PDS(DATAFILE) SYSOPTS="UNIQUE=YES")
```

Sterling Connect:Direct resolves the member name to DATAFIL1.

Copy a Sequential File from z/OS to a Member of a Physical Data Base File on an i5/OS Node

This Process copies a sequential file from z/OS to a member of a physical data base file on the i5/OS node. The RUN TASK then sends a message notifying an i5/OS user that the Process completed.

All SYSOPTS keyword values must be enclosed in parentheses, and the entire SYSOPTS string must be enclosed in double quotation marks. Sterling Connect:Direct syntax requires backslashes to continue the SYSOPTS over multiple lines when the Process is submitted from a z/OS node. Bracketing backslashes allow for continuation of quotation marks when they begin and end on different lines.

Specifying COMPRESS without a subparameter indicates that blanks are compressed during transmission and converted back to the original string during decompression. The default for the COMPRESS parameter is PRIMEchar=X'40'.

```
PROC#01  PROCESS  SNODE=OS400.CDI1                              -
                  PNODE=SC.OS390.CD5A                           -
                  PRTY=8                                        -
                  NOTIFY=%USER                                  -
                  CLASS=4                                       -
                  SNODEID=(USERID,PSWRD)
STEP001  COPY  FROM  (PNODE                                     -
                      DSN=CD.OS400.SEQFILE1                     -
                      DISP=SHR)                                 -
                      COMPRESS                                  -
               TO    (SNODE                                     -
                      DSN='CD/OS400(TEST01)'                    -
                      SYSOPTS=\"TYPE(MBR)\                      -
                      \TEXT('CREATED BY PROC#001')\             -
                      \RCDLEN(133)"\                            -
                      DISP=RPL)
STEP002  RUN TASK    (PGM = OS400) SNODE                        -
                      SYSOPTS=\"\                               -
                      \CMD(\                                    -
                      \SNDBRKMSG\                               -
                      \MSG('PROCESS PROC#01 HAS COMPLETED')\    -
                      \TOMSGQ(DSP07)\                           -
                      \   )\                                    -
                      \"\
```

Copy a Member of a Physical Data Base File from i5/OS to a Sequential File on z/OS

This Process copies a member of a physical data base file from the i5/OS node to a sequential file on z/OS. DCB information is specified for file allocation on z/OS. The SYSOPTS keyword value is enclosed in parentheses, and the SYSOPTS string is enclosed in double quotation marks. Specifying COMPRESS EXT indicates that repetitive strings in the data will be compressed and converted to the original string during decompression.

```
PROC#001  PROCESS    SNODE=OS400.CDI1                          -
                     PNODE=SC.OS390.CD5A                       -
                     PRTY=8                                    -
                     NOTIFY=%USER                              -
                     CLASS=4                                   -
                     HOLD=NO                                   -
                     SNODEID=(RSMITH,ROGER)
STEP001   COPY FROM (SNODE                                     -
                     DSN='ABC/OS400(TEST01)'                   -
                     SYSOPTS="TYPE(MBR)"                       -
                     DISP=SHR)                                 -
                     COMPRESS EXT                              -
          TO        (PNODE                                    -
                     DSN=ABC.OS400.SEQFILE9                    -
                     DCB=(RECFM=FB,LRECL=133,BLKSIZE=23408)    -
                     DISP=RPL)
```

Copy a Data Set from z/OS to a Spooled File on i5/OS

This example copies a data set from z/OS to a spooled file on an i5/OS node. The specified SYSOPTS parameters override the defaults defined for the print device at installation. The parameter CTLCHAR(*FCFC) is always used when the source z/OS file has a record format (RECFM) of xxA, which indicates that it contains ANSI carriage control.

```
TEST01A  PROCESS       SNODE=OS400.CDI1                          -
                       PNODE=SC.OS390.CD5A                       -
                       PRTY=8                                    -
                       NOTIFY=%USER                              -
                       CLASS=4                                   -
                       HOLD=NO                                   -
                       SNODEID=(RSMITH,ROGER)
STEP001 COPY  FROM     (PNODE                                    -
                       DSN=CD.OS400.REPORTS                      -
                       DISP=SHR)                                 -
              TO       (SNODE                                    -
                       DSN=REPORTS401                            -
                       SYSOPTS=\"TYPE(SPLF)\                     -
                               \DEV(*JOB)\                       -
                               \DEVTYPE(*IPDS)\                  -
                               \CTLCHAR(*FCFC)\                  -
                               \SPOOL(*YES)\                     -
                               \PRTQLTY(*NLQ)\                   -
                               \PRTTXT('*** END OF PAGE ***')\   -
                               \JUSTIFY(100)\                    -
                               \OUTQ(*JOB)\                      -
                               \COPIES(2)\                       -
                               \MAXRCDS(999)\                    -
                               \FILESEP(2)\                      -
                               \HOLD(*YES)\                      -
                               \SAVE(*YES)\                      -
                               \OUTPTY(*JOB)\                    -
                               \USRDTA(RSMITH)"\                 -
                       DISP=RPL)
```

Copy a Member of a PDS from z/OS to a Spooled File on i5/OS

This Process copies a member of a PDS from z/OS to a spooled file on an i5/OS node. The SYSOPTS parameter PRTQLTY specifies near-letter-quality print.

```
PROC#001 PROCESS      SNODE=OS400.CDI1            -
                      PRTY=8                      -
                      NOTIFY=RSMITH               -
                      CLASS=4                     -
                      SNODEID=(RSMITH,ROGER)
STEP001  COPY  FROM  (PNODE                       -
                      DSN=CD.OS400.CNTL(LRECL80)  -
                      DISP=SHR)                   -
               TO    (SNODE                       -
                      DSN=REPORTS402              -
                      SYSOPTS=\"\                 -
                              \TYPE(SPLF)\        -
                              \PRTQLTY(*NLQ)\     -
                              \"\)
```

Copy to an Entry-Sequenced File (z/OS to HP NonStop)

This Process, submitted from Sterling Connect:Direct for HP NonStop, contains two steps:

- STEP1, which executes a FUP RUN TASK to purge a file. (Note the ! character in the PURGE command— it is required when issuing some FUP commands in BATCH mode.) The output from the FUP process will be written to spool location $s.#FUP.
- STEP2, which executes a COPY to pull a file from z/OS to HP NonStop. The HP NonStop file is created NEW, so the desired file attributes are specified in SYSOPTS. The space attributes are explicitly declared rather than allowing them to default to the space information provided by the z/OS node. Checkpointing is disabled (CKPT=0K).

```
A110257  PROCESS   SNODE=CD.OS390
STEP1    RUN TASK  PGM=FUP PNODE                          -
                   PARM=("/OUT $S.#FUP/"                  -
                         "PURGE $B.FILERESO.A110257 !")
STEP2    COPY FROM (SNODE DISP=SHR                        -
                    DSN=DATA1.SEQ.A110257)                -
              TO   (PNODE DISP=NEW                        -
                    DSN=$B.FILERESO.A110257               -
                    SYSOPTS=("SET TYPE E"                 -
                             "SET CODE 0"                 -
                             "SET REC 4000"               -
                             "SET BLOCK 4096"             -
                             "SET EXT(32,32)"             -
                             "SET MAXEXTENTS 128"         -
                             "SET XLATE ON")              -
                    CKPT=0K
```

Create a Code 101 File (z/OS to HP NonStop)

This Process, submitted from the z/OS node, copies a file from z/OS and creates a type 101 EDIT file. Space attributes are allowed to default. Checkpointing is disabled.

```
TEST  PROCESS     SNODE=CD.HPNONSTOP
STEP1 COPY FROM  (PNODE DISP=SHR                -
                  DSN=DATA1.VSAME001.DATA)      -
           TO    (SNODE DISP=NEW                -
                  DSN=$WORK02.HPDATA.UNST1      -
                  SYSOPTS=\"'SET TYPE U'   \||  -
                          \'SET CODE 101'  \||  -
                          \'SET XLATE ON'" \)   -
                  CKPT=0K
```

Create a Code 0 Oddunstructured File (z/OS to HP NonStop)

This Process, submitted from the z/OS node, copies a file from z/OS and creates a type 0 oddunstructured file. Space attributes are allowed to default. Checkpointing is disabled, as is translation.

```
TEST  PROCESS      SNODE= CD.HPNONSTOP
STEP1 COPY  FROM  (PNODE DISP=SHR                -
                   DSN= DATA1.VSAME001.DATA)     -
            TO    (SNODE DISP=NEW                -
                   DSN=$WORK02.HPDATA.UNST1      -
                   SYSOPTS=\"'SET TYPE U'   \||  -
                           \'SET ODDUNSTR'  \||  -
                           \'SET XLATE OFF'" \)  -
                   CKPT=0K
```

Copy a Sequential File from a z/OS Node to an HP NonStop Node

The following Process transfers a sequential file from z/OS to HP NonStop. The Process is submitted on the z/OS node. Because the parameter SET EXT (50 50) is specified, the HP NonStop file system allocates 50 pages (one page=2048 bytes) to the primary extent and 50 pages to all secondary extents. Using the default for MAXEXTENTS results in a file with a capacity of 16 x 50 x 2048, or 1638400 bytes.

```
HPNONSTP  PROCESS      PNODE=CD.OS390.DALL             -
                       SNODE=CD.BILL                   -
                       SNODEID=(127.202,WILLIE)
STEP01    COPY  FROM  (PNODE DSN=SMITH.FDATA           -
                       DISP=SHR)                       -
          TO          (DSN='$C.ROGER.TESTSND'         -
                       SYSOPTS=\"'SET EXT(50 50)'"   \  -
                       DISP=NEW)
```

Copy a File Submitted from z/OS to HP NonStop

The following Process, submitted on a Sterling Connect:Direct for z/OS node, copies a data set from z/OS to a remote node on an HP NonStop EXPAND network. The SNODEID field passes the userid 147.200 to the HP NonStop, along with the password MONEY.

The userid and password are validated before the Sterling Connect:Direct for z/OS system copies the JSMITH.DATA file into the TESTOUT file in the $B.JOHN volume on the \TSCIEXT system.

The following Process illustrates copying a file submitted from z/OS to HP NonStop.

```
GENSEND1 PROCESS      SNODE=ACCT.JOHN                  -
                      SNODEID=(147.200,MONEY)
STEP01   COPY  FROM  (PNODE                            -
                      DSN=JSMITH.DATAFILE              -
                      DISP=SHR)                        -
         TO          (SNODE                            -
                      DSN='\TSCIEXT.$B.JOHN.TESTOUT'   -
                      DISP=NEW                         -
                      SYSOPTS=\"'SET TYPE E'    \||    -
                              \xd5 SET XLATE ON'"  \)
```

Enclose the data set name in single quotation marks.

Copy a File from HP NonStop on an EXPAND Network to z/OS

This Process is submitted on the z/OS node to copy a file from an HP NonStop node on an EXPAND network to a z/OS node. The Sterling Connect:Direct for HP NonStop data transformation facility (DTF) (or server) is on system \SYSTEM, but the file is being copied from system \SYSEXT.

Enclose the data set name in single quotation marks, because the Process is submitted from a Sterling Connect:Direct for z/OS node.

```
GENSEND2  PROCESS      SNODE=CD.SMITH                      -
                       SNODEID=(149.205,WILLIE)
STEP01    COPY  FROM  (DSN='\SYSEXT.$A.ABC.TESTOUT' SNODE  -
                       DISP=SHR)                          -
                TO    (DSN=JSMITH.TESTDATA PNODE          -
                       DISP=RPL)
```

Use FUP in a Process Submitted on z/OS to Delete a File on HP NonStop

In this Process, the FUP utility is used to delete an HP NonStop file. The Process is submitted on z/OS.

```
PART0003  PROCESS    SNODE=HPNONSTOP.CD
STEP1     RUN TASK  (PGM=FUP                                    -
                     SYSOPTS="/OUT $S.#SFUP01/PURGE $USER.FILE01.* !"  -
                     SNODE  )
```

Use Sterling Connect:Direct to Allocate a Partitioned File on a Single System (z/OS to HP NonStop)

In this Process, the Sterling Connect:Direct system is used to allocate an HP NonStop partitioned file. All partitions reside on the same system. The Process is submitted on z/OS.

```
PART0001  PROCESS      SNODE=HPNONSTOP.CD
STEP1     COPY  FROM  (PNODE                                  -
                       DSN=DATA1.TESTFILE.SEQ.FB80L          -
                       DISP=SHR                              -
                      )                                      -
                TO    (SNODE                                 -
                       DSN=$A.TESTFILE.PCODE0                -
                       DISP=NEW                              -
                       SYSOPTS=\"'SET CODE 0'           \||  -
                               \ 'SET TYPE U'           \||  -
                               \ 'SET PART (1,$B,10,5)' \||  -
                               \ 'SET PART (2,$C,10,5)' \||  -
                               \ 'SET MAXEXTENTS 16'    \||  -
                               \ 'SET XLATE ON'"        \    -
                      )
```

SYSOPTS Syntax Conventions (z/OS to HP NonStop)

This Process will copy a file from the z/OS node to the HP NonStop node. Because the Process is submitted from the z/OS node, syntax conventions must follow those established for z/OS. In this example, bracketing backslashes are used to continue a string containing special characters across multiple lines.

```
SYSOPTS  PROCESS      PNODE=SC.OS390.NODE1          -
                      SNODE=TSCI.HPNONSTOP          -
                      SNODEID=(157.214 ABLE1)
STEP01   COPY  FROM  (DSN=DATA1.SMALLER             -
                      UNIT=SYSDA)                   -
                TO    (DSN=$C.ACCTPROC.HPNONSTOP    -
                      SYSOPTS=\"'SET EXT(10,10)\ ||  -
                              \ ,XLATE ON\ ||       -
                              \ ,TYPE E\ ||         -
                              \ ,REC 100'"\         -
                      DISP=RPL)
```

Multiple SET commands can also be specified as follows:

```
SYSOPTS  PROCESS      PNODE=SC.OS390.NODE1          -
                      SNODE=TSCI.HPNONSTOP          -
                      SNODEID=(157.214 ABLE1)
STEP01   COPY  FROM  (DSN=DATA1.SMALLER             -
                      UNIT=SYSDA)                   -
                TO    (DSN=$C.ACCTPROC.HPNONSTOP    -
                      SYSOPTS=\"'SET EXT(10,10)\ ||  -
                              \  SET XLATE ON\ ||   -
                              \  SET TYPE E\ ||     -
                              \  SET REC 100'"\     -
                      DISP=RPL)
```

Copy Files Between z/OS and UNIX

This Process copies a sequential file from a z/OS node to a UNIX node (STEP1) and back to the z/OS node (STEP2). It then submits itself to run again (STEP3).

```
CPY01  PROCESS    SNODE=den34                                    -
                  SNODEID=(test1,propty2)
STEP1  COPY                                                      -
                  FROM  (PNODE DSN=TSTFL1.IN.FILE01 DISP=SHR)    -
                  TO    (SNODE DSN='out01' DISP=SHR)
STEP2  COPY                                                      -
                  TO    (PNODE DSN=TSTFL1.IN.FILE01 DISP=SHR)    -
                  FROM  (SNODE DSN='out01' DISP=SHR)
       IF        (STEP1 EQ 0)  THEN
       RUN TASK  (PGM=DMNOTFY2,                                  -
                  PARM=(FAIL,TSTFL1.IN.FILE01,%USER))            -
                  PNODE
                  EXIT
       EIF
STEP3  SUBMIT     DSN=TSTFL1.PROCESS.MVS.TO.UNIX(CPY01)          -
                  SUBNODE=PNODE  CASE=YES
```

This example shows a file copy from a z/OS node to a UNIX node using SYSOPTS. The UNIX DSN and SYSOPTS strings are in lower case and are enclosed in double quotation marks.

```
BENCHM1  PROCESS      SNODE=kyla                                  -
                      SNODEID=(barry1,322red)
STEP1    COPY                                                    -
               FROM  (PNODE DSN=TEST.TESTFILE.BENCH.M1 DISP=SHR   -
                      CKPT=1M                                     -
                      COMPRESS EXTENDED                           -
               TO    (SNODE DSN='benchm1' DISP=MOD               -
                      SYSOPTS=":datatype=text:strip.blanks=no:")
```

Copy a File from z/OS to VM

This Process copies a z/OS data set to a file on the 191 disk of VM user IVVB. If the file exists, the Sterling Connect:Direct system replaces it. If the file does not exist, the Sterling Connect:Direct system creates it as indicated by the DISP=RPL parameter.

The following Process illustrates copying a file from z/OS to VM.

```
ZOSTOVM  PROCESS      SNODE=CD.VM.NODE1          -
                      CLASS=10
STEP01   COPY  FROM  (DSN=SMITH.FDATA            -
                      DISP=SHR                   -
                      UNIT=SYSDA)                -
               TO    (DSN='TEMP1 OUTPUT'        -
                      LINK=(IVVB,WIVVB,W,191)    -
                      DISP=(RPL))
```

Copy a z/OS PDS to a Set of Files on VM

The COPY statement in this example copies a PDS, excluding all members with names beginning with the characters DM, to files on VM with file names corresponding to the member names on z/OS. The file type is ACCTDATA.

```
STEP01  COPY  FROM (DSN=OS390.PDS.ACCT     -
                   DISP=SHR               -
                   EXCLUDE=(DM*)          -
                   UNIT=SYSDA)            -
              TO   (DSN='* ACCTDATA'      -
                   LINK=(PEI,WPEI,W,300)  -
                   DISP=(RPL))
```

Copy a z/OS File to Tape on VM

This Process copies a disk file from a Sterling Connect:Direct for z/OS node to tape at the Sterling Connect:Direct for VM node. If the file exists, the Sterling Connect:Direct system replaces it. If the file does not exist, the Sterling Connect:Direct system creates it as indicated by the DISP=RPL parameter.

```
TAPE    PROCESS      SNODE=CD.VM.NODE1         -
                     NOTIFY=USERID
STEP01  COPY  FROM  (DSN=CHELSEN.DATA.FILE     -
                     DISP=SHR)                 -
              TO    (DSN=TAPE.DATA.FILE        -
                     UNIT=TAPE                 -
                     LABEL=(1,SL,EXPDT=92067)  -
                     SNODE                     -
                     DISP=RPL)
```

Copy a z/OS File to Spool on VM

This Process copies a disk file from a Sterling Connect:Direct for z/OS node to spool at a Sterling Connect:Direct for VM node.

```
READER  PROCESS      SNODE=CD.VM.NODE1                          -
                     NOTIFY=USERID
STEP01  COPY  FROM  (DSN=RRT.SALES.DATA                         -
                     DISP=OLD)                                  -
              TO    (DSN='!SPOOL VKK SALES DATA'                -
                     DCB=(DSORG=PS,RECFM=F,LRECL=80,BLKSIZE=80))
```

Use the SYSOPTS Parameter (z/OS to OpenVMS)

This Process shows how to specify the SYSOPTS parameter with the MOUNT and PROTECTION keywords.

When one or more SYSOPTS parameters are specified and they continue across several lines, bracketing backslashes (\) and the double bar (||) concatenation characters are required.

The entire SYSOPTS string must be enclosed in double quotation marks. Sterling Connect:Direct syntax requires using backslashes to continue the SYSOPTS over multiple lines when the Process is submitted from a z/OS node.

```
SYSOPTS3  PROCESS      SNODE=CD.VMS.NODE NOTIFY=CDA1              -
                       SNODEID=(RSMITH,ROGER) HOLD=YES
STEP01    COPY  FROM  (DSN=SMITH.CNTL(IEBGENER)                  -
                       PNODE)                                   -
                TO    (DSN='MSA0:TAPE.DAT'                      -
                       DISP=RPL SNODE                           -
                       SYSOPTS=\"MOUNT='MSA0:/NOLABEL'\ ||      -
                               \ NODISMOUNT\ ||                 -
                               \ PROTECTION='S:E,O:E,G:E,W:E' "\)
```

Copy a File from z/OS to Microsoft Windows

This Process copies a file from a z/OS node to a Microsoft Windows node. The DSN (data set name) for the Microsoft Windows file uses the Universal Naming Convention (UNC) to specify the computer name \WIN-NODE and the share name \ROOT_D. This can be used as an alternative to specifying a drive letter such as C:\, and must be used when copying a file to a remote computer where the Sterling Connect:Direct server is not running. The Sterling Connect:Direct for Microsoft Windows node, WIN.1200.TCP, must have access to the share, \ROOT_D, for the copy operation to succeed. The SYSOPTS parameter DATATYPE(TEXT) indicates that the data contains text (rather than binary data) and EBCDIC to ASCII translation will be performed.

```
TONT1  PROCESS      SNODE=WIN.1200.TCP                              -
                    HOLD=NO                                         -
                    RETAIN=NO
STEP1  COPY  FROM  (DSN=TEST.DEL.DATA99)                           -
             TO    (DSN='\WIN-NODE\ROOT_D\USERS\TEST1\DATA1.TXT'   -
                    SYSOPTS="DATATYPE(TEXT)"                        -
                    DISP=(RPL))
```

The following Process is a variation on the previous example. In this example, the data set names (DSN) are defined as symbolic variables in the COPY statement (&DSN1 and &DSN2), and are resolved at the time the process is submitted. This process also uses the STARTT parameter to specify a day of the week and time when the process will execute, and the RETAIN=YES parameter to indicate that the Process should stay in the TCQ and execute again at the next scheduled start time (STARTT). This Process will execute automatically each Monday at 3:00 a.m.

```
TONT1  PROCESS      SNODE=WIN.1200.TCP                              -
                    HOLD=NO                                         -
                    RETAIN=YES                                      -
                    STARTT=(MONDAY,03:00)                           -
                    &DSN1='TEST.DEL.TEST99'                         -
                    &DSN2='\WIN-NODE\ROOT_D\USERS\TEST1\DATA1.TXT'
STEP1  COPY  FROM  (DSN=&DSN1)                                     -
             TO    (DSN=&DSN2                                      -
                    SYSOPTS="DATATYPE(TEXT)"                        -
                    DISP=(RPL))
```

Copy a File from Sterling Connect:Direct for z/OS to Sterling Connect:Direct for i5/OS

In this example COPY statement, the member FILEA in PDS USER.TESTLIB on the z/OS system is copied to the member TEST in the TESTFILES/PROCLIB file on the Sterling Connect:Direct i5/OS node. If the member TEST already exists in the file TESTFILES/PROCLIB on the Sterling Connect:Direct i5/OS node, it is replaced by this version.

```
/* COPY FROM: SOURCE DATA SET ATTRIBUTES  (z/OS) */
STEP01     COPY   FROM  (DSN=USER.TESTLIB(FILEA)            -
                        PNODE                               -
                        DISP=SHR                            -
                        )                                   -
/* COPY TO: DESTINATION DATA SET ATTRIBUTES (i5/OS) */
                  TO    (SNODE                              -
                        DSN='TESTFILES/PROCLIB(TEST)'       -
                        DISP=RPL                            -
                        SYSOPTS=\"\                         -
                                \TYPE(MBR)\                 -
                                \TEXT('COPIED FROM FILEA')\  -
                                \"\                         -
                        )
```

Copy a File From z/OS to i5/OS

In this example, the member FILEA in PDS USER.TESTLIB on a z/OS system is copied to the member TEST in the TESTFILES/PROCLIB file on a Sterling Connect:Direct for i5/OS node. If the member TEST already exists in the file TESTFILES/PROCLIB on the Sterling Connect:Direct for i5/OS node, it is replaced by this version.

```
/* COPY FROM: SOURCE DATA SET ATTRIBUTES  (z/OS) */
STEP01     COPY   FROM  (DSN=USER.TESTLIB(FILEA)            -
                        PNODE                               -
                        DISP=SHR                            -
                        )                                   -
/* COPY TO: DESTINATION DATA SET ATTRIBUTES (i5/OS) */
                  TO    (SNODE                              -
                        DSN='TESTFILES/PROCLIB(TEST)'       -
                        DISP=RPL                            -
                        SYSOPTS=\"\                         -
                                \TYPE(MBR)\                 -
                                \TEXT('COPIED FROM FILEA')\  -
                                \"\                         -
                        )
```

Copy DBCS Data Sets Using Translation Tables in z/OS

In this example consisting of six COPY steps, six different translation tables are being used to convert from one format to another while each file is being transferred from the PNODE to the SNODE. Each translation table is specified using the SYSOPTS DBCS parameter in the TO clause.

```
DBCSTST1 PROCESS SNODE=CD.MAINFRAME                                 -
SNODEID=(USERID,PASSWRD)
************************************************************************
* DBCS CHINESE NEW HOST CODE TO CHINESE BIG 5 (NHCXBG5)
************************************************************************
STEP01                                                              -
        COPY  TO    ( SNODE                                         -
                      DSN=CD.O.DBCS.NHCXBG5                         -
                      DISP=(RPL,CATLG)                              -
                      UNIT=SYSDA                                    -
                      DCB=(RECFM=VB,LRECL=254,BLKSIZE=4096,DSORG=PS) -
                      SPACE=(254,(1000,100))                        -
                      SYSOPTS="DBCS=NHCXBG5"                        -
                    )                                               -
              FROM  ( PNODE                                         -
                      DSN=CD.DBCS.NHC                               -
                      DISP=SHR                                      -
                    )
```

```
*********************************************************************
* DBCS TABLE BG5XNHC
*********************************************************************
STEP02                                                              -
        COPY  TO    ( SNODE                                         -
                      DSN=CD.O.DBCS.BG5XNHC                         -
                      DISP=(RPL,CATLG)                              -
                      UNIT=SYSDA                                    -
                      DCB=(RECFM=VB,LRECL=254,BLKSIZE=4096,DSORG=PS) -
                      SPACE=(254,(1000,100))                        -
                      SYSOPTS="DBCS=BG5XNHC"                        -
                    )                                               -
              FROM  ( PNODE                                         -
                      DSN=CD.DBCS.BG5XNHC                           -
                      DISP=SHR                                      -
                    )
*********************************************************************
* DBCS TABLE EBCXKSC
*********************************************************************
STEP03                                                              -
        COPY  TO    ( SNODE                                         -
                      DSN=CD.O.DBCS.EBCXKSC.SOSI                    -
                      DISP=(RPL,CATLG)                              -
                      UNIT=SYSDA                                    -
                      DCB=(RECFM=VB,LRECL=254,BLKSIZE=4096,DSORG=PS) -
                      SPACE=(254,(1000,100))                        -
                      SYSOPTS="DBCS=(EBCXKSC,0E,0F)"                -
                    )                                               -
              FROM  ( PNODE                                         -
                      DSN=CD.DBCS.EBCXKSC.FILE.SOSI                 -
                      DISP=SHR                                      -
                    )
*********************************************************************
* DBCS TABLE KSCXEBC
*********************************************************************
STEP04                                                              -
        COPY  TO    ( SNODE                                         -
                      DSN=CD.O.DBCS.KSCXEBC                         -
                      DISP=(RPL,CATLG)                              -
                      UNIT=SYSDA                                    -
                      DCB=(RECFM=VB,LRECL=254,BLKSIZE=4096,DSORG=PS) -
                      SPACE=(254,(1000,100))                        -
                      SYSOPTS="DBCS=KSCXEBC"                        -
                    )                                               -
              FROM  ( PNODE                                         -
                      DSN=CD.O.DBCS.EBCXKSC.SOSI                    -
                      DISP=SHR                                      -
                    )
*********************************************************************
* DBCS TABLE EBCXKPC
*********************************************************************
STEP05                                                              -
        COPY  TO    ( SNODE                                         -
                      DSN=CD.O.DBCS.EBCXKPC.NOSO                    -
                      DISP=(RPL,CATLG)                              -
                      UNIT=SYSDA                                    -
                      DCB=(RECFM=VB,LRECL=254,BLKSIZE=4096,DSORG=PS) -
                      SPACE=(254,(1000,100))                        -
                      SYSOPTS="DBCS=(EBCXKPC,00,00)"                -
                    )                                               -
              FROM  ( PNODE                                         -
                      DSN=CD.DBCS.EBCXKPC.NOSO                      -
                      DISP=SHR                                      -
                    )
*********************************************************************
* DBCS TABLE KPCXEBC
*********************************************************************
```

```
STEP06                                                          -
        COPY  TO    ( SNODE                                     -
                      DSN=CD.O.DBCS.KPCXEBC                     -
                      DISP=(RPL,CATLG)                          -
                      UNIT=SYSDA                                -
                      DCB=(RECFM=VB,LRECL=254,BLKSIZE=4096,DSORG=PS) -
                      SPACE=(254,(1000,100))                    -
                      SYSOPTS="DBCS=KPCXEBC"                    -
                    )                                           -
              FROM  ( PNODE                                     -
                      DSN=CD.O.DBCS.EBCXKPC.NOSO                -
                      DISP=SHR                                  -
                    )
```

MBCS Conversion During z/OS to UNIX Copy

The following is an example for copying a z/OS file to a UNIX system and converting the IBM®-943 code set on the z/OS system to the SJIS code set on the UNIX system.

```
COPY  FROM  (PNODE DISP=SHR                                     -
             DSN=TEST1.MBCS.IBM943                              -
             SYSOPTS="CODEPAGE=(IBM-943,UTF-8)" )               -
      TO    (SNODE DISP=RPL                                     -
             DSN=\'/home/unix3500/cab1\ ||                      -
                 \ /CD/TestData/Source/codepage/Sjis390.txt' \  -
             sysopts=":codepage=(UTF-8,SJIS):datatype=binary" )
```

First, the IBM-943 code set data is converted to UTF-8 on the z/OS node. Then, as specified in the TO clause, the UTF-8 encoded data is transferred to the UNIX node, which converts the data to the SJIS code set and writes a UNIX file.

MBCS Conversion During Microsoft Windows to z/OS Copy

The following is an example for copying a Microsoft Windows file to a z/OS system and converting the 932 code set on Microsoft Windows to the IBM-943 code set on the z/OS system.

```
FROMNT  COPY  TO    (PNODE DISP=(,CATLG)                        -
                     UNIT=SYSDA                                 -
                     VOL=SER=USER01                             -
                     SPACE=(TRK,(1,1))                          -
                     DCB=(DSORG=PS,RECFM=U,BLKSIZE=24000)       -
                     DSN=TEST2.MBCS.LMORL1.SJIS001              -
                     SYSOPTS="CODEPAGE=(UTF-8,IBM-943)" )       -
              FROM  (SNODE DISP=SHR                             -
                     file='I:\Lcm\CD\TestData\Source\SjisCodepage.txt' -
                     sysopts="codepage(932,65001) datatype=binary"  )
```

As specified in the FROM clause, the 932 code set data in the SjisCodepage.txt file is converted to the 65001 code set (the UTF-8 equivalent on Microsoft Windows) on the Microsoft Windows node. The 65001 encoded data is then transferred to the z/OS node, which converts the data to the IBM-943 code set and writes a z/OS sequential file.

Note that the z/OS output file is defined as having an undefined record format (RECFM=U). When an MBCS conversion is done for a file that is created on a z/OS receiving node, the RECFM for the file must be specified as either V (Variable), VB (Variable Block), or U. The results of an MBCS conversion can result

in a change to the physical length of the data, and the amount of the change can vary, depending on the data. RECFM=F (Fixed) or FB (Fixed Block) cannot be used, because the final data length is not fixed.

MBCS Conversion During z/OS to z/OS Copy

The following is an example for copying from one z/OS node to another and converting the IBM-930 code set to IBM-1047 via UTF-8.

```
STEP01  COPY  FROM  (PNODE DISP=SHR                          -
                     DSN=TEST3.MBCS0001.IBM930               -
                     SYSOPTS="CODEPAGE=(IBM-930,UTF-8)" )    -
              TO    (SNODE DISP=(,CATLG)                     -
                     UNIT=SYSDA SPACE=(CYL,(3,3))            -
                     VOL=SER=USER01                          -
                     DCB=(RECFM=VB,LRECL=90,BLKSIZE=24000)   -
                     DSN=CHICAGO.MBCS0001.IBM1047            -
                     SYSOPTS="CODEPAGE=(UTF-8,IBM-1047)" )
```

First, the IBM-930 code set data is converted to the UTF-8 encoding on the sending node. The UTF-8 encoded data is then transferred to the receiving node, which converts the UTF-8 data to the IBM-1047 code set and writes a z/OS sequential file.

The RECFM of the file is specified as Variable Format (RECFM=VB) to allow for a flexible output record length. Also, to allow for a possible increase in data length due to conversion, the LRECL of the receiving file must be larger than the LRECL of the sending file. In the example shown, the LRECL of the sending file is 80. For this particular MBCS conversion, the receiving file was successfully created by specifying LRECL as 90. Other conversions may require a larger value to avoid an SVSJ032I error during the Copy. If RECFM=VB, BLKSIZE for the output file must be at least as large as LRECL+4.

Copying a File from zOS to Microsoft Windows using Substitution in a Destination Path:

This example shows how to use symbolic substitution to specify Microsoft Windows path names in a COPY statement.

About this task

In this example, the z/OS data set TEST.DATASET is copied to the Microsoft Windows file STERLING\CD\CDWIN\TEMP\TEST.TXT

Procedure

1. Create a batch command that signs on to Sterling Connect:Direct, submits a Process that creates the variables, and signs off:

```
SIGNON USERID=(userid,)
   SUBMIT PROC=EXNTDIR      -
   &FROMDSN=TEST.DATASET    -
   &DIR1=CDWIN              -
   &DIR2=TEMP               -
   &FILENAME=TEST.TXT
SIGNOFF
```

This Process creates the following variables:

Variable	Value	Description
&FROMDSN	TEST.DATASET	The name of the source data set.
&DIR1	CDWIN	Third level of the destination path.
&DIR2	TEMP	Fourth level of the destination path.
&FILENAME	TEST.TXT	The destination file name.

2. Create the following Process to copy the file:

```
NTDIRPTH  PROCESS                                              -
          SNODE=STERLING.WINDOWS                               -
          &DIR1=,                                              -
          &DIR2=,                                              -
          &FILENAME=,
   SYMBOL &S1 = STERLING
   SYMBOL &S2 = CD
   SYMBOL &TODSN = \'\\\ || &S1 || \\ || &S2 || \\ ||          -
      &DIR1 || \\ || &DIR2 || \\ || &FILENAME || \'\
STEP01   COPY  FROM (PNODE                                     -
                     DSN=&FROMDSN                              -
                     DISP=SHR)                                 -
               TO   (DSN=&TODSN                                -
                     SYSOPTS="DATATYPE(TEXT)"                  -
                     DISP=(RPL))
```

This Process defines the following symbolic values:

Variable	Value	Description
&S1	STERLING	First level of the destination path.
&S2	CD	Second level of the destination path.
&TODSN	\'\\\ . . . &FILENAME || \'\	The full destination path.

The following table shows how the &TODSN variable resolves (two vertical bars [||] indicate concatenation).

Value	Resolves to . . .
\'\\\	'\
&S1	STERLING
\\	\
&S2	CD
\\	\
&DIR1	CDWIN
\\	\
&DIR2	TEMP
\\	\
&FILENAME	TEST.TXT
\'\	‘

As a result, the &TODSN variable resolves to

```
'\STERLING\CD\CDWIN\TEMP\TEST.TXT'
```

CODEPAGE Conversion During a File Copy (z/OS to Microsoft Windows)

The following example shows how to specify the CODEPAGE SYSOPTS parameter when copying a file from a U.S. or Canadian Sterling Connect:Direct for z/OS system (IBM-1047 codepage) to an ANSI Latin Sterling Connect:Direct for Microsoft Windows system (65001 codepage, which is the UTF-8 equivalent on Microsoft Windows).

```
CODEPGNT  PROCESS      SNODE=SHARE1.WINDOWS                -
                       HOLD=NO
CODEPGCP  COPY  FROM  (PNODE                               -
                       DSN=NQORR1.DATAFILE.MIXED           -
                       DISP=SHR                            -
                       SYSOPTS="CODEPAGE=(IBM-1047,UTF-8)") -
                       CKPT=5M                             -
                TO    (SNODE                               -
                       DSN='C:\SRN_TESTDATA\MIXED.TXT'     -
                       DISP=RPL                            -
                       SYSOPTS="CODEPAGE(65001,1252)")
```

Create and Copy a LARGE Data Set (z/OS)

This COPY statement transmits a z/OS BASIC data set to a new LARGE data set on another z/OS system. The DSNTYPE parameter specifies that the receiving data set is created as a LARGE format sequential data set.

```
COPY FROM (DSN=BASIC.SOURCE)             -
     TO   (DSN=LARGE.DEST DSNTYPE=LARGE)
```

Chapter 3. RUN JOB Statement Examples

HP NonStop

Run a Job on the z/OS Node from a Process Submitted on the HP NonStop Node

In this Process submitted from the HP NonStop node, STEP1 will execute FUP to purge $B.FILERESO.STEST. STEP2 will copy DATA1.SMALLER from the z/OS node to $B.FILERESO.STEST at the HP NonStop node. Conditional logic (STEP3) is then used to check the completion code of STEP1. If the completion code is greater than **4**, no further processing will occur. Otherwise STEP4 will execute DATA1.CNTL(IEFBR14A) at the z/OS node.

The Sterling Connect:Direct for HP NonStop system cannot execute the RUN JOB statement; however, the Sterling Connect:Direct for HP NonStop node as the PNODE can submit a Process to an z/OS or VSE node, and the SNODE can execute the RUN JOB.

```
RUN    PROCESS      PNODE=CD.HP NONSTOP          -
                    SNODE=SS.CD.OS390
STEP1  RUN TASK    (PGM=FUP                      -
                    PARM= ('/OUT $S.#STEST/',    -
                           'VOLUME $B.FILERESO', -
                           'PURGE STEST  '))
STEP2  COPY  FROM  (DSN=DATA1.SMALLER SNODE      -
                    DISP=SHR)                    -
             TO    (DSN=$B.FILERESO.STEST  PNODE -
                    DISP=NEW)
STEP   IF  (STEP1 GT 4)  THEN
            EXIT
       EIF
STEP4  RUN JOB  (DSN=DATA1.CNTL(IEFBR14A)) SNODE
```

Execute Commands on UNIX from a Process Submitted from HP NonStop

This Process shows how to initiate a Process from HP NonStop that executes commands at a UNIX node. The SYSOPTS string must be in the proper case for UNIX and enclosed in double quotation marks.

```
RUNJOB1  PROCESS  SNODE=CD.v1200                    -
                  snodeid=(user,pswd)
UNIXJOB  RUN JOB  SYSOPTS="ls -l > outjob.tan" SNODE
```

Microsoft Windows

Microsoft Windows Run Job Statement

This **run job** statement executes the program testwin.exe on a remote Microsoft Windows system.

```
jobstep   run job   snode (dsn=WINNT)
                    sysopts="pgm(c:\winnt35\system32 estwin.exe)"
```

OpenVMS

Submit a Process with a RUN JOB on OpenVMS

The following example shows a Process that submits the command procedure TEST_RJOB.COM from the directory JSMITH.TEST on disk DUC4 on the SNODE (OpenVMS). The SYSOPTS, or system-specific parameters, consist of four parameters, each enclosed in single quotes. P1, P2, and P3 are parameters passed to the TEST_RJOB.COM command procedure. LOG displays the TEST_RJOB.COM commands as they execute to view for error and completion messages. The entire SYSOPTS is enclosed in double quotation marks.

```
RUNJOBT  PROCESS   SNODE=CD.VMS.SMITH
STEP01   RUN JOB  (DSN='DUC4:[JSMITH.TEST]TEST_RJOB.COM' PNODE    -
                   SYSOPTS="P1='DEF' P2='123' P3='BR549' NOPRINT  -
                   LOG='DUC4:[JSMITH.TEST]TEST.LOG'"
```

Print and Delete the Log File on Sterling Connect:Direct for OpenVMS

In this example, a command procedure is submitted to the SNODE for execution in the batch queue. No parameters are specified. The log file is printed and deleted because the default parameters of PRINT and NOKEEP are assumed. Note that the command procedure NOTIFY.COM is executed from the default login directory of the submitter.

```
RUNJ1  PROCESS   SNODE=CD.VAX
       RUN JOB  (FILE='NOTIFY.COM' SNODE)
```

Keep the Log File on Sterling Connect:Direct for OpenVMS

In this example, a command procedure executes on the VAX in the batch queue. Only the NOPRINT subparameter is specified, resulting in the log file being kept and named after the first (or only) file in the job.

```
RUNJ2  PROCESS   SNODE=CD.VAX
       RUN JOB  (FILE='NOTIFY.COM' SNODE)  -
                 SYSOPTS="NOPRINT"
```

Print and Keep the Log File on Sterling Connect:Direct for OpenVMS

In this example, NOTIFY.COM executes on the SNODE. During Process execution, the parameter (P1) is passed to the command procedure. The log file is printed and kept. LOG is not specified because KEEP ensures that the output is saved.

```
RUNJ3  PROCESS   SNODE=CD.OS390
       RUN JOB  (FILE='NOTIFY.COM' SNODE)          -
                 SYSOPTS="P1='SUCCESS' PRINT  KEEP"
```

Run Job on OpenVMS

In this example, the RUN JOB statement specifies that RJOB.COM executes on the PNODE. During Process execution, two parameters (P1 and P2) are passed to the

command procedure. The LOG subparameter specifies that the file RJOB.LOG is created. It is not printed or deleted because NOPRINT is specified; KEEP is assumed.

```
RUNJ4  PROCESS   PNODE=CD.VAX
STEP1  RUN JOB  (DSN=$DISK1:[CD_20.PROCESS]RJOB.COM PNODE)
                 SYSOPTS="P1='CD_20' P2='1-JAN-1995' NOPRINT
                 LOG='$DISK1:[CD_20.LOG]RJOB.LOG'"
```

UNIX

Run a Job on UNIX from a Process Submitted from Another UNIX Node

This Process shows how to initiate a Process that executes commands at another UNIX node. The SYSOPTS string must be enclosed in double quotation marks.

```
proc2   process  snode=unix.node
step01  run job  snode
                 sysopts="ls > /abc/file/user/us1/lsout; ls -lax"
   pend
```

Run a Job on z/OS from a Process Submitted on UNIX

This Process shows how to initiate a Process from UNIX to run a job at an z/OS node. The DSN string must be in uppercase characters to satisfy z/OS syntax requirements.

```
proc2   process   snode=OS390.node
                  snodeid=(user01,pswd01)
step01  run job  (dsn=SRCDATA.SET(TEST))
                  snode
     pend
```

Run a Job on Microsoft Windows from a Process Submitted on UNIX

This Process shows how to initiate a Process from UNIX that executes commands at a Microsoft Windows node. In this example, the command **dir** is issued on the node with the output redirected to a file called list.out. When issuing a console command, such as **del**, **dir**, or **move**, specify the command in the **sysopts** parameter exactly the way it is issued at the command prompt directly on the Microsoft Windows system.

```
ntsub1  process  snode=cd.win
                 retain=yes
                 prty=yes
runj01  run job  snode
                 sysopts="cmd(dir d:\users\jdoe1\*.*
>>d:\users\jdoe1\list.out)"
  pend;
```

Example UNIX Run Job

In this example, the application **myjob** is submitted to the **pnode** system.

```
jobstep   run job     sysopts="myjob param1"
```

VM/ESA

RUN JOB Facility (VM to VM)

This Process sends the file named SIGNON CDOP in PUN format to the reader for CDA5A. This Process can be used to transmit jobs to be processed by VMBATCH.

```
RUNJOB2   PROCESS    SNODE=CD.VM.NODE NOTIFY=CDA8
STEP01    RUN JOB   (SNODE DSN='SIGNON CDOP'                -
                     LINK=(CDA6,RCDA6,RR,191) BATCHID=CDA5A)
```

Example VMESA Run Job

The example RUN JOB statement named STEP001 sends the job, BATJOB, to the reader for USER01.

```
VMBATCH  PROCESS   SNODE=CD.VM.OTHER NOTIFY=USER01
STEP001  RUN JOB  (PNODE                          -
                   DSN='EX BATJOB'                -
                   LINK=(USER01,RUSER01,RR,191)   -
                   BATCHID=VMBATCH                -
                  )
```

VSE

VSE Run Job Statement

The example VSE RUN JOB statement named STEP1 executes job TESTJOB.J from the source library on the SNODE. TESTJOB.J must exist as a member in a source library on the SNODE system. It must have been cataloged previously as a J member.

```
STEP1 RUN JOB  (DSN=J(TESTJOB))       -
      SNODE
```

z/OS

Submit a Job to the z/OS Internal Reader

You can submit a RUN JOB Process from either the SNODE or the PNODE. If you specify PNODE on the RUN JOB, the job is submitted to the primary node; if you specify SNODE, the job is submitted to the secondary node. In both cases, the job is submitted to the internal reader if both nodes are running Sterling Connect:Direct for z/OS.

The node that you submit the Process to is the PNODE or Process control node by definition; the other node involved in the Process is the SNODE.

In the following example, the Process, PROC1, is submitted from the PNODE, CDA. PROC1 then instructs the SNODE, CDB, to execute the job titled JOB123 in the library JOB.STREAM.LIB. The job is submitted to the z/OS internal reader on CDB.

The data set specified in the RUN JOB statement must exist on CDB.

```
PROC1   PROCESS   SNODE=CDB
STEP01  RUN JOB  (DSN=JOB.STREAM.LIB(JOB123) SNODE)
```

Alternatively, you can be signed on to CDA and submit the following RUN JOB Process from CDA. In this case, the data set specified in the RUN JOB must exist on CDA (PNODE).

In this example, the job contained in JOB.STREAM.LIB(JOB123) is submitted to the z/OS internal reader on the PNODE system.

```
PROC1   PROCESS   SNODE=CDB
STEP01  RUN JOB  (DSN=JOB.STREAM.LIB(JOB123) PNODE)
```

The JCL must exist on the node where you want it to execute.

Run a Job on the i5/OS Node from a Process Submitted on the z/OS Node

This example is submitted on z/OS to run a job on i5/OS. DSN is necessary when submitting from z/OS. The contents of the SYSOPTS parameter define the program to run on the i5/OS node.

```
RJPROC01  PROCESS   SNODE=CD2200                -
                    SNODEID=(USER1,PASSWD1)
STEP0001  RUN JOB  (DSN=AS400)  SNODE          -
                    SYSOPTS=\"\                 -
                            \CMD(\              -
                            \CALL\              -
                            \PGM(KRM/KJOBTST)\  -
                            \)\                 -
                            \"\
```

Execute Commands on UNIX from a Process Submitted from z/OS

This Process shows how to initiate a Process from z/OS that executes commands at a UNIX node. The SYSOPTS string must be in the proper case for UNIX and enclosed in double quotation marks.

```
PROC2   PROCESS   SNODE=UNIX.NODE
STEP01  RUN JOB  (DSN=UNIX)                                -
                  SNODE                                    -
                  SYSOPTS="/company/payroll/monthly/jan.exe"
```

Use Run Job to Submit a Job on an i5/OS Node

In this example z/OS or z/OS Process, the job JANPMTS is submitted on the i5/OS node.

```
JOBSTEP  PROCESS   SNODE=CDAS400
                   SNODEID(USER01,PSWD01)
STEP01   RUN JOB  (DSN=AS400) SNODE
                   SYSOPTS=\"\
                           \CMD(\
                           \CALL PGM(INV/JANPMTS)\
                           \)\
                           \"\
```

Submit a Run Job on i5/OS from z/OS

In this example z/OS Process, the job JANPMTS is submitted to the i5/OS node.

```
JOBSTEP  PROCESS   SNODE=CDAS400
                   SNODEID(USER01,PSWD01)
STEP01   RUN JOB  (DSN=AS400) SNODE
                   SYSOPTS=\"\
                           \CMD(\
                           \CALL PGM(INV/JANPMTS)\
                           \)\
                           \"\
```

Chapter 4. RUN TASK Statement Examples

HP NonStop

Run FUP (Sterling Connect:Direct for HP NonStop Run Task)

This Process starts FUP and copies a disk file to the spooler location $S.#SPL2. Output from the FUP command (success/failure) is sent to $S.#SPL1.

Note: This Process will only work in SNA environments.

```
PROC1  PROCESS    SNODE=SYSCLX
STEP1  RUN TASK  (PGM=FUP                                  -
                  PARM=('/NAME $FP,OUT $S.#SPL1/'          -
                       ,'COPY $A.PROCVOL.ACCT,$S.#SPL2')   -
                                                           -
                  PNODE
```

Submit a Process with a Sterling Connect:Direct for OpenVMS RUN TASK from an HP NonStop Node

This Process, submitted from the Sterling Connect:Direct for HP NonStop node, will issue a command to invoke a DCL command procedure. Output will be directed to the terminal. Upon successful execution of the command procedure, the terminal of the specified user will beep.

```
VAXRUN  PROCESS   PNODE=CD.HPNONSTOP                       -
                  SNODE=CD.VMS
STEP1   RUN TASK (PGM=VMS) SNODE                           -
                  SYSOPTS="OUTPUT = '_NDC31'               -
                  CMD = 'DIR DUC4:[USER1.DATA]*.DAT'       -
                  CMD = 'REPLY/USER=USER1/BELL SUCCESS'"
```

Submit a Process with a Sterling Connect:Direct for z/OS RUN TASK from an HP NonStop Node

This Process, submitted from the Sterling Connect:Direct for HP NonStop node, performs a RUN TASK on the Sterling Connect:Direct for z/OS node, then copies from HP NonStop to z/OS.

Note: This Process will only work in SNA environments.

```
TANCPY  PROCESS    SNODE=CD.HPNONSTOP                                -
                   SNODEID=(GRP.USR.PASWRD)
STEP1   RUN TASK  (
                   PGM=DMRTAMS,                                      -
                   PARM=(C"FREE=CLOSE,RETURN=(DD),SYSOUT=X           -
                         C" DELETE (CSDQA1.FILETEST.VSAM0016)        -",
                         C"  CLUSTER                                -",
                         C" DEFINE CLUSTER (                       -",
                         C"  NAME(CSDQA1.FILETEST.VSAM0016)         -",
                         C"  RECORDS(1500 100)                      -",
                         C"  VOLUMES(USER01)                        -",
                         C"  INDEXED                                -",
                         C"  KEYS(12 0)                             -",
                         C"  RECORDSIZE(80 80)                      -",
                         C"  SHAREOPTIONS(3)                        -",
                         C"  )                                      -",
                         C"  DATA (                                 -",
                         C"  NAME(CSDQA1.FILETEST.VSAM0016.DATA)    -",
                         C"  CONTROLINTERVALSIZE(4096)              -",
                         C"  )                                      -",
                         C"  INDEX (                                -",
                         C"  NAME(CSDQA1.FILETEST.VSAM0016.IDX)     -",
                         C"  CONTROLINTERVALSIZE(4096)              -",
                         C"  SHAREOPTIONS(3)                        -",
                         C"  )                                      -",
                         C"  DATA (                                 -",
                         C"  NAME(CSDQA1.FILETEST.VSAM0016.DATA)    -",
                         C"  CONTROLINTERVALSIZE(4096)              -",
                         C"  )                                      -",
                         C"  INDEX (                                -",
                         C"  NAME(CSDQA1.FILETEST.VSAM0016.IDX)     -",
                         C"  CONTROLINTERVALSIZE(4096)              -",
                         C"  )
                  )
                  ) SNODE
*
STEP2  COPY FROM  ( PNODE                                           -
                    DSN='\cycr.$user.qafiles.VSAM0001'              -
                    DCB=(LRECL=80)                                  -
                    SYSOPTS="SET XLATE ON"                          -
                  )                                                 -
           TO     ( SNODE                                          -
                    DSN=CSDQA1.FILETEST.VSAM0016                    -
                    DISP=OLD                                        -
                  )
```

Use Symbolics in a Sterling Connect:Direct HP NonStop Run Task to Place a Job In the Spooler On Hold

This Sterling Connect:Direct HP NonStop Process uses symbolics to place a job in the spooler on hold. Output is sent to $S.#SYMB. The job number will be resolved at Process submission. Double quotation marks enable the resolution of the &value in a quoted string.

```
PROC1   PROCESS       PNODE=CD.HPNONSTOP1
                      SNODE=CD.HPNONSTOP1
STEP1   RUN TASK      (PGM=SPOOLCOM)
                      SYSOPTS=("/OUT $S.#SYMB/JOB &JNUM,HOLD")
                      PNODE
```

The command to submit the Process is:

```
SUBMIT PROC PROC1 &JNUM=24
```

i5/OS

Create and Save an Online Save File Through Sterling Connect:Direct for i5/OS

This RUN TASK statement creates the online save file named SAVEFILE1 in the library TESTSV1 on the i5/OS. A copy of the library is saved to SAVEFILE1 in the library TESTDT1. All SYSOPTS keyword values must be enclosed in parentheses, and the entire SYSOPTS string must be enclosed in double quotation marks. Sterling Connect:Direct syntax requires using backslashes to continue the SYSOPTS over multiple lines. Bracketing backslashes allow for continuation of quotation marks when they begin and end on different lines.

```
STEP003  RUN TASK  (PGM=AS400) SNODE                              -
                    SYSOPTS=\"\                                   -
                            \CMD( \                               -
                            \CRTSAVF \                            -
                            \ FILE(TESTSV1/SAVEFILE1)  \          -
                            \ TEXT('CREATED BY PROCESS CDTEST') \  -
                            \    ) \                              -
                            \CMD( \                               -
                            \SAVLIB \                             -
                            \ LIB(TESTDT1) \                      -
                            \ DEV(*SAVF) \                        -
                            \ SAVF(TESTSV1/SAVEFILE1) \           -
                            \    ) \                              -
                            \"\
```

Notify the i5/OS User of Successful Process Completion

This RUN TASK statement sends the message **PROCESS NEWACT HAS COMPLETED** to the message queue of workstation DSP07.

```
STEP012  RUN TASK   (PGM=AS400) SNODE                             -
                     SYSOPTS=\"\                                  -
                             \CMD( \                              -
                             \ SNDBRKMSG \                        -
                             \ MSG('PROCESS NEWACT HAS COMPLETED')\  -
                             \ TOMSGQ(DSP07) \                    -
                             \ ) \                                -
                             \"\
```

Restore Libraries Through Sterling Connect:Direct for i5/OS

This Process restores to the i5/OS system the library named TESTDT1 in the save file named SAVEFILE2 in library TESTSV1. This library is restored to a library named TESTRL1.

```
STEP008B  RUN TASK  (PGM=AS400) SNODE                    -
                     SYSOPTS=\"\                         -
                             \CMD( \                     -
                             \ RSTLIB \                  -
                             \ SAVLIB(TESTDT1) \         -
                             \ DEV(*SAVF) \              -
                             \ SAVF(TESTSV1/SAVEFILE2) \ -
                             \ RSTLIB(TESTRL1) \         -
                             \ ) \                       -
                             \"\
```

Submit a Process with a Sterling Connect:Direct for OpenVMS RUN TASK from an i5/OS Node

The example command sends a file from Sterling Connect:Direct for i5/OS to Sterling Connect:Direct for OpenVMS and performs a RUNTASK.

```
CDSND  SNODE(DWY1.TCP)
SNODENVIRN(OPENVMS)
FDSN('CDABC220/INITPARMS(INITPARMS)')
TDSN('DISK$SUP:<DYOUN1>AS400.RCV') FMSYSOPTS('TYPE(MBR)')
SNODEID(USERID USERPASSWORD) TDISP(RPL)TDCB(*N *N *N PS)
CDRUNTASK  SNODE(DWY2.TCP)
SNODENVIRN(OPENVMS)
CMD('CMD='`DEL
DISK$SUP:<DYOUN1>AS400.RCV;*''')
SNODEID(USERID USERPASSWORD)
```

Microsoft Windows

Submit a Process with a Sterling Connect:Direct for HP NonStop RUN TASK from a Microsoft Windows Node

The following is an example of a Sterling Connect:Direct for HP NonStop RUN TASK submitted from a Sterling Connect:Direct for Miscrosoft Windows node.

```
NDM       PROCESS    SNODE=NDM.BAY1DR /*$HP NONSTOP$*/
                     SNODEID=(NCC.NDM,xxxxxxxx)
BEGINMSG  RUN TASK  (PGM = BTFNDMLG
          SYSOPTS = "/NAME, IN $DATA2.BTFDATA.DEFTABLE, VOL $DATA2.BTFLOAD          -
                     /ND033 DNLOAD FUNBCC FUNBBETA TRU05")
                     SNODE
COPYSTEP  COPY FROM (PNODE
                     FILE = g:\vms\mftd\export\fusi\backup\TradeExport_20000818.002
                     SYSOPTS = "DATATYPE(BINARY)STRIP.BLANKS(NO)XLATE(NO)" )
               TO   (SNODE
                     FILE = $test.bftdata.beta5
                     SYSOPTS = "'SET TYPE E' 'SET BLOCK 200' 'SET REC 200'"
                     DISP = RPL)
          IF (COPYSTEP EQ 0) THEN
                     GOTO OKMSG
          ELSE
ERRORMSG  RUN TASK  (PGM = BTFNDMLG
                     SYSOPTS = "/NAME, IN $DATA2.BTFDATA.DEFTABLE, VOL $DATA2.BTFLOAD -
                                /ND035 DNLOAD FUNBCC FUNBBETA TRU05")
                     SNODE
                     EXIT
OKMSG     RUN TASK  (PGM = BTFNDMLG
                     SYSOPTS = "/NAME, IN $DATA2.BTFDATA.DEFTABLE, VOL $DATA2.BTFLOAD -
                                / ND034 DNLOAD FUNBCC FUNBBETA TRU05")
                     SNODE
FT3IN     RUN TASK  (PGM = BTFNDMDN
                     SYSOPTS = "/NAME, IN $DATA2.BTFDATA.DEFTABLE, VOL $DATA2.BTFLOAD -
                                / DNLOAD FT3IN $test.btfdata.beta5 FUNBCC FUNBBETA TRU05")
```

Notify the i5/OS User of the Start of a Process

This Process performs a RUN TASK that sends the message PROCESS HAS STARTED to the message queue of workstation WSUSER01.

```
PROC01  PROCESS   SNODE=CD2200
                  SNODEID=(USER01,PSWD01)
STEP1   RUN TASK  SNODE (PGM=AS400)
                  SYSOPTS="CMD(SNDBRKMSG MSG('PROCESS HAS STARTED!')
                  TOMSGQ(WSUSER01))"
PEND
```

Submit a Process from Sterling Connect:Direct for Microsoft Windows to Run DMRTSUB on z/OS

The following Process is submitted from Sterling Connect:Direct for Microsoft Windows to run DMRTSUB on a Sterling Connect:Direct for z/OS node. This Run Task using DMRTSUB will submit a job to run on z/OS and pass the symbolic for &NTDISP to the JCL it submits.

See the *IBM Sterling Connect:Direct for z/OS User Guide* for more information on DMRTSUB.

```
DMRTASK  PROCESS
                   &NTDISP="RPL"
                   SNODE=CSD.MVS.LEVEL1 /*$MVS$*/
                   HOLD=NO
                   CLASS=1
                   PRTY=10
                   EXECPRTY=10
                   RETAIN=NO
                   SNODEID=(USERID,USERPASSWORD)
RTASK01  RUN TASK  SNODE (PGM=DMRTSUB)
                   SYSOPTS="C'DSN=JSMITH.PROCESS(SAMPLE),DISP=SHR'
                            C'NTDISP &NTDISP'"
     PEND
```

Microsoft Windows Run Task Statement

This **run task** statement starts testwin.exe and waits for it to finish:

```
jobstep   run task   snode (pgm=WINNT
                     sysopts="pgm(C:\winnt35\system32 estwin.exe)"
```

OpenVMS

Submit a Process with a Sterling Connect:Direct for HP NonStop RUN TASK from an OpenVMS Node

This Process, submitted from the Sterling Connect:Direct for OpenVMS node, will invoke FUP at the Sterling Connect:Direct for HP NonStop node and write detailed information about the files in the subvolume to the spooler.

```
TANRUN  PROCESS    PNODE=CD.VMS                                  -
                   SNODE=CD.HPNONSTOP SNODEID=(GRP.USR,PASWRD)
STEP1   RUN TASK  (PGM=FUP                                       -
                   PARM=('/OUT $S.#FUPOUT/'                      -
                         'INFO $DATA.TESTF.*,DETAIL'))           -
                   SNODE
```

Submit a Process with a RUN TASK on OpenVMS from an OpenVMS Node

The following example shows a Process submitted from OpenVMS and executed on OpenVMS. The RUN TASK statement is coded with DCL commands that execute synchronously.

```
RTVMS   PROCESS     SNODE=CD.VMS.NODE1
STEP01  RUN TASK   (PGM=VMS) PNODE                                -
                    SYSOPTS="OUTPUT='DUC4:[JSMITH.TEST]RTLOG.LIS'  -
                    CMD='SET DEFAULT DUC4:[JSMITH.TEST]'           -
                    CMD='DIR'                                      -
                    CMD='SHOW TIME'")
```

Run Task on OpenVMS

In this example, the RUN TASK statement executes the PURGE command at the OpenVMS node, which in this Process is also the PNODE. PURGE deletes all files with a type of .OUT in directory ACCT.TEST on device U1. The RUN TASK statement also executes the command procedure CREATE_DATA.COM in directory

SC1 on device U1. Output from the OpenVMS process created by the Sterling Connect:Direct for OpenVMS RUN TASK statement is routed to device NL.

```
RUNT4     PROCESS     PNODE=CD.VAX
STEP1     RUN TASK    (PGM=VMS)  PNODE SYSOPTS="
                      OUTPUT='NL:'
                      CMD='PURGE U1:[ACCT.TEST]*.OUT  '
                      CMD='@U1:[SC1]CREATE_DATA.COM'"
```

UNIX

Submit a Process from UNIX to Run a Program on z/OS

This example shows a Process initiating from UNIX that runs a program on z/OS. The **sysopts** string must be in uppercase characters to satisfy z/OS syntax requirements.

```
proc2   process   snode=OS390.node
step01  run task  snode (pgm=DMNOTIFY)
                  sysopts="CL44'DATA.BASE.P1',F'0010', XL8'FFA8'"
 pend
```

Submit a Process with a Run Task on UNIX from Another UNIX Node

This Process initiates a Process from a UNIX node that executes commands at another UNIX node. The **sysopts** string must be enclosed in double quotation marks.

```
proc2    process   snode=unix.node
step01   run task  snode sysopts = "ls; ls -lax > lsout.ncr"
       pend
```

Submit a Process from UNIX to Run a Program on i5/OS

This Process initiates a Process from a UNIX node that executes commands on an i5/OS node to delete two libraries. The **sysopts** string must be enclosed in double quotation marks.

```
proc1   process    snode=as400
                   snodeid=(userid,passwrd)
step02  run task  (pgm=AS400)
                   snode
                   sysopts="CMD(DLTLIB LIB(URGRSSDT1)) CMD(DLTLIB LIB(URGRSS))"
 pend
```

Submit a Process from UNIX to Run a Program on Microsoft Windows

This Process copies a binary file from a Microsoft Windows node to a UNIX node. If the copy is successful, a **run task** statement is performed on the Microsoft Windows node (**snode**) that will delete the source (from) file on the Microsoft Windows node. To delete the file, the keyword **cmd** is specified in the **sysopts** followed by the **del** command.

```
proc1  process      snode=CD.WIN
                    &file1="d:\data\out\reprts01.dat"
                    &file2="/data/in/reprts01.dat"
copy1  copy  from  (file=&file1
                    sysopts="datatype(binary)"
                    snode)
             to    (file=&file2
                    pnode)
       if (copy1 eq 0) then
run1   run task     snode
                    sysopts="cmd(del &file1)desktop(no)"
       eif
 pend
```

In the previous example, rather than coding the specific file names in the process, symbolic variables are used in both the **copy** and **run task** statements. Since no user input is required for the delete command, the **desktop** parameter is set to NO and no console window is created on the Microsoft Windows desktop.

Example UNIX Run Task

In this example, the user-supplied program is specified in a **run task** statement. The program requires two parameters. The program is located on the primary node. The **run task** statement is contained within a Process that is also located on the **pnode**.

```
taskstep run task sysopts="grep -e CTRC s19931215.001 > stat.txt"
```

VM/ESA

Execute DMRTDYN in a RUN TASK Environment (VM)

This Process calls DMRTDYN to determine if a file exists. If it does not exist, the user receives a nonzero return code, and a call is made to allocate the file.

```
DMRTDYN  PROCESS  SNODE=SC.VM.USER1
LOCATE1  RUN TASK (PGM=DMRTDYN                                    -
                   PARM=(C'ALLOC',                                -
                         C' DSN=''PROFILE EXEC''',                -
                         C' DISP=(SHR) LINK=(IVVB,WIVVB,RR,191)', -
                         C' DD=F1')) PNODE
         IF(LOCATE1 NE 0) THEN
LOCATE2  RUN TASK (PGM=DMRTDYN                                    -
                   PARM=(C'ALLOC',                                -
                         C' DSN=''PROFILE EXEC''',                -
                         C' DISP=(NEW) LINK=(IVVB,WIVVB,RR,191)', -
                         C' DD=F1')) PNODE
         EIF
LOCATE3  RUN TASK (PGM=DMRTDYN                                    -
                   PARM=(C'UNALLOC '                              -
                         C' DD=F1')) PNODE
```

Resolve Symbolics Within DMRTDYN in a RUN TASK Environment (VM)

This example illustrates the structure of a Sterling Connect:Direct Process that passes a parameter with single quotation marks in a DMRTDYN environment. Backslashes allow the resolution of the symbolic that must be entered between single quotation marks.

```
TESTSYM  PROCESS   SNODE=SC.VM.USER1 &XXX=XXX
                   SYMBOL        &VMFILE=\xd5 MTT\   &XXX   \ FILETYPE'\
LOCATE1  RUN TASK (PGM=DMRTDYN                                          -
                   PARM=(C'ALLOC ',                                     -
                         C' DSN='    \&VMFILE\    \xd5 ',\              -
                         C' DISP=(SHR) LINK=(IVVB,WIVVB,RR,191)',       -
                         C' DD=F1')) PNODE
LOCATE3  RUN TASK (PGM=DMRTDYN                                          -
                   PARM=(C'UNALLOC '                                    -
                         C' DD=F1')) PNODE
```

Submit a Process with a RUN TASK on i5/OS from a VM Node

This example is initiated on a Sterling Connect:Direct for VM/ESA node to execute a RUN TASK on Sterling Connect:Direct for i5/OS. The SYSOPTS parameter specifies the i5/OS CL command DLTLIB.

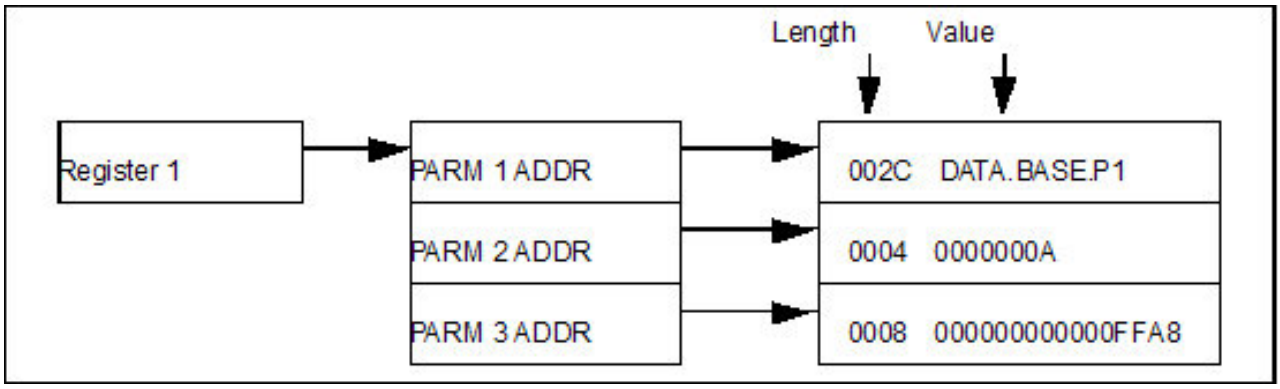
```
RTVM      PROCESS    SNODE=CD.OS400
*********************************************************************
*  RUN TASK INITIATED FROM VM TO RUN ON OS400
*********************************************************************
STEP3000  RUN TASK  (PGM=AS400) SNODE        -
                     SYSOPTS=\"\             -
                             \CMD( \         -
                             \ DLTLIB \      -
                             \ LIB(TEST1) \  -
                             \ ) \           -
                             \"\
```

Example VMESA Run Task

This example RUN TASK statement runs the program named MYTASK. It is attached to the Process on the secondary node (SNODE) and is passed a list of three parameter addresses.

```
STEP1  RUN TASK  (PGM=MYTASK                -
                  PARM=(CL44'DATA.BASE.P1',  -
                        F'0010', XL8'FFA8'))  -
                  SNODE
```

The following figure shows the parameter passing convention for the program MYTASK. Register 1 points to a parameter list of three parameters. It would contain zero (0) if no parameters were specified. Sterling Connect:Direct sets the high-order bit in PARM 3 ADDR to indicate the end of the PARM list.



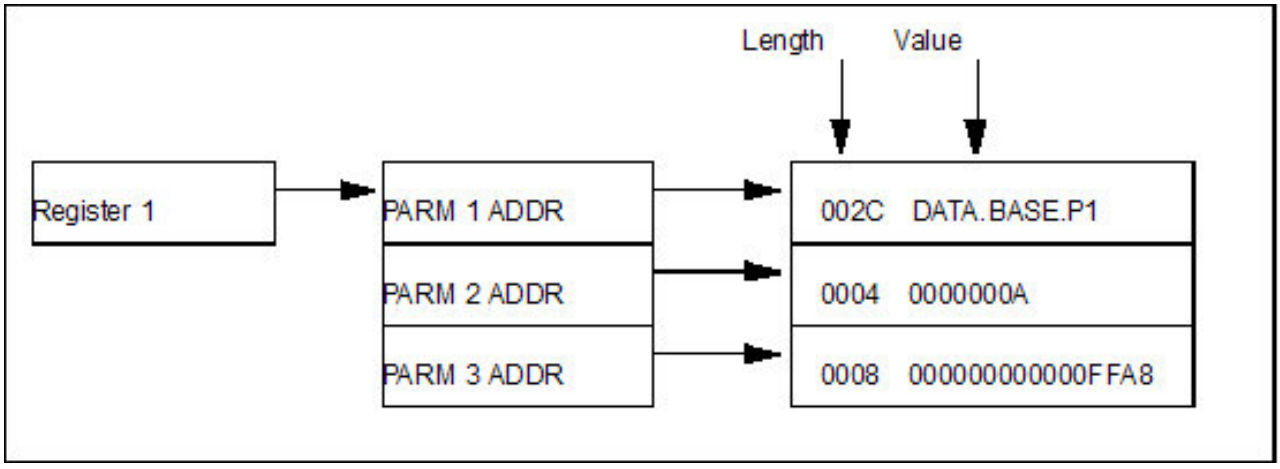
VSE

VSE Run Task Statement

This example RUN TASK statement runs the program named MYTASK. It is attached to the Process on the secondary node (SNODE) and is passed a list of three parameter addresses.

```
STEP1  RUN TASK  (PGM=MYTASK                  -
                  PARM=(CL44'DATA.BASE.P1',   -
                        F'0010', XL8'FFA8'))  -
                  SNODE
```

The following figure shows the parameter passing convention for the program MYTASK. In this case, Register 1 points to a parameter list of three parameters. It would contain 0 if no parameters were specified. Sterling Connect:Direct sets the high-order bit in PARM 3 ADDR to indicate the end of the PARM list.



z/OS

RUN TASK Examples for CA-7 (z/OS to z/OS)

The following is an example of a RUN TASK statement that interfaces with the CA-7 scheduling package. The U7SVC program is supplied by CA-7, the job

scheduling system by Computer Associates International, Inc. The U7SVC program may return a code of **0**, regardless of completing the request.

For Processes with RUN TASK statements running U7SVC programs, replace xx with the length of the entire PARM including the d=.

```
STEP01  RUN TASK  (PGM=U7SVC,            -
                   PARM=(CLxx"d=&dsn") )  -
                   SNODE
```

In the following example, the symbolic variable, &UCC7DSN, is resolved to Z456789.TARGET, which is the proper format for the U7SVC program.

Note: To properly execute with CA-7, Sterling Connect:Direct for z/OS must not be running under CA-7 control.

```
PROC02  PROCESS      SNODE=CD.NODEB                        -
                     &DSN1=Z123456.SRC                     -
                     &DSN2=Z456789.TARGET
HF201   COPY  FROM  (DSN=&DSN1 DISP=SHR)                   -
              TO    (DSN=&DSN2 DISP=(RPL,CATLG))
                     SYMBOL      &UCC7DSN=&DSN2
        IF  (HF201=0)  THEN
HF202         RUN TASK (PGM=U7SVC,PARM=(CLxx"D=&UCC7DSN"))  -
                       SNODE
        EIF
```

The following RUN TASK statement shows another way to interface with CA-7:

```
STEP01  RUN TASK  PGM=U7SVC,                              -
                  PARM=(\xd5 DEMAND,JOB=\ || &POSTJOB ||  -
                       \,SCHID=001'\))                    -
                  SNODE
```

RUN TASK Using Control-M

The following RUN TASK statement shows an example of a Control-M interface:

```
RUN_TASK  PROCESS    SNODE=CCC.RZ1
STEP01    RUN TASK  (PGM=CTM@IF10                                  -
                     PARM=(                                       -
                           C'FUNC=LOADNSKE',                      -
                           C'UNIQID=SKA#Y4OTH4T',                 -
                           C'DATREC=SKELLIB :JOBP.FT1A.SKEL',     -
                           C'DATREC=SKELNAME:CDSKEL',             -
                           C'DATREC=&&JOBNAME=Y4OTH4T',           -
                           C'DATREC=&&JCLL=JOBP.CI1A.JCLL',       -
                           C'DATREC=&&DSN=',                      -
                           C'DATREC=&&GROUP=CONNECT-DIRECT',      -
                           C'DATREC=&&DESC=CSLUX_TESTAUSLOESUNG'))  -
                     SNODE
```

RUN TASK Using DMRTDYN (z/OS)

This Process, SAMPLE1, is submitted from a Sterling Connect:Direct for z/OS node. Through the RUN TASK statement, DMRTDYN determines if the data set exists at the SNODE. If this data set exits, the LOCATE step receives a return code

of **0**, and the next step, DELETE, deletes the data set. STEP01 executes, regardless of the return code received from the LOCATE step. (STEP01 copies from the PNODE to the SNODE.)

```
SAMPLE1  PROCESS    PNODE=&PNODE                                          -
                    SNODE=&SNODE                                          -
                    HOLD=NO
LOCATE   RUN TASK  (PGM=DMRTDYN,                                          -
                    PARM=(C"LOCATE DSN=&HLQ2..SAMP.OUT"))                 -
                    SNODE
         IF  (LOCATE = 0)  THEN
DELETE   RUN TASK  (PGM=DMRTDYN,                                          -
                    PARM=(C"ALLOC DSN=&HLQ2..SAMP.OUT DISP=(OLD,DELETE)"  -
                          F'-1'                                           -
                          C"UNALLOC DSN=&HLQ2..SAMP.OUT"))
         EIF
STEP01   COPY  FROM  (DSN=&HLQ1..SAMPLE.IN)                               -
               TO    (DSN=&HLQ2..SAMP.OUT                                 -
                      DISP=RPL)
```

The command used to submit SAMPLE1 follows:

```
SUBMIT PROC=SAMPLE1                          -
         &PNODE=CD.LOCAL                     -
         &SNODE=CD.REMOTE                    -
         &HLQ1=$LOCAL                        -
         &HLQ2=$REMOTE
```

Copy a Member of an Object from i5/OS to a PDS Member and then Deleting the Library

This Process copies a member of an object from i5/OS to a member of a PDS on z/OS. The RUN TASK then deletes the library (with all its objects) from the i5/OS node.

```
PROC#001  PROCESS      SNODE=OS400.CD1                -
                       PNODE=SC.OS390.CDA             -
                       PRTY=8                         -
                       NOTIFY=%USER                   -
                       CLASS=4                        -
                       HOLD=NO                        -
                       SNODEID=(RSMITH,RSMITH)
STEP001   COPY  FROM   (                              -
                       SNODE                          -
                       DSN='LIBRY/RSMITH(LRECL80)'    -
                       SYSOPTS="TYPE(MBR)"            -
                       DISP=SHR                       -
                      )                               -
                       COMPRESS                       -
                TO     (                              -
                       PNODE                          -
                       DSN=RSMITH.OS400.CNTL(LRECL80) -
                       DISP=SHR                       -
                      )
STEP002   RUN TASK     (PGM=AS400) SNODE              -
                       SYSOPTS=\"\                    -
                               \ CMD( \               -
                               \ DLTLIB \             -
                               \ LIB(LIBRY) \         -
                               \ ) \                  -
                               \"\
```

Submit a Process with a RUN TASK on OpenVMS from an z/OS Node

The following example shows a Process with a RUN TASK statement submitted from z/OS to execute on an OpenVMS node:

```
RUNTASK   PROCESS    SNODE=CD.NODE
STEP01    RUN TASK  (PGM=VMS) SNODE                                    -
                     SYSOPTS=\"CMD='SUBM/LOG=CDL:CD/USER=USR/PARA  -
                            (\  || &FILENM  ||  \)CDC:RCV_CD'"\
```

Initiate a RUN TASK Statement at the HP NonStop Node (z/OS to HP NonStop)

Using conditional logic, this Process executes a program based on the completion code of STEP01. The PGM statement limits the user to eight characters. Concatenation characters are required within the RUN TASK statement because the parameters being passed to FUP are on multiple lines. TERM is coded on the RUN TASK statement so Sterling Connect:Direct Processes can continue uninterrupted in the event the program being executed abends. The Process is submitted on z/OS.

```
PROC1   PROCESS      SNODE=CD.TAN                                      -
                     SNODEID=(117.202,PSWRD)
STEP01  COPY  FROM  (PNODE DSN=JSMITH.GENPROC.LIB(COPY) DISP=SHR)      -
              TO    (SNODE DSN=$B.ROGER.ACCTJAN DISP=RPL)
STEP02  IF   (STEP01 GT 0) THEN
        EXIT
        EIF
STEP03  RUN TASK    (PGM=FUP                                          -
                     SYSOPTS="(/IN $USER.BGK.T1, TERM $ZTNP1.#PTH001H/)"  -
                     SNODE
```

Use Symbolics with a RUN TASK Statement (z/OS to HP NonStop)

This Process is submitted from the z/OS node to run a program at the HP NonStop node. The Process is structured so that the program name (PGM) and associated run-options for the HP NonStop RUN command are symbolics. (Symbolics allow you to predefine a Process that can be used for multiple applications.) The symbolics will be resolved when the Process is submitted. Because the NAME parameter is followed by a space, HP NonStop will assign a name to the HP NonStop process.

Note: This Process will only work in SNA environments.

```
PROCESS1  PROCESS    PNODE=CD.OS390                  -
                     SNODE=CD.HPNONSTOP
STEP01    RUN TASK  (PGM = &PGM                     -
                     PARM=(\ "/IN    \ || &INFILE  ||  -
                           \  ,OUT   \ || &OUTFILE ||  -
                           \  ,PRI   \ || &PRI     ||  -
                           \  ,CPU   \ || &CPU     ||  -
                           \  ,NAME  \                 -
                           \  /")\)                    -
                     SNODE
```

The following Sterling Connect:Direct syntax rules apply to this Process:

- The string of HP NonStop RUN command options must be enclosed in forward slashes (/). This is an HP NonStop syntax requirement.
- Bracketing backslashes (\) are positioned around variables in the string so that strings containing special characters can continue across multiple lines. Symbolics (&value) are not enclosed in bracketing backslashes.

Note: Run options for the HP NonStop RUN command must be separated by commas.

- Because the PNODE is the z/OS node, two vertical bars preceded and followed by blanks (||) are used to concatenate the value of a symbolic to the string. Resolution of the symbolic occurs before concatenation.

The command used to submit PROCESS1 is as follows:

```
SUB PROC=PROCESS1 &PGM=FUP &INFILE=FUPIN &OUTFILE=$S.#SPL1 &PRI=100 &CPU=0
```

Use Bracketing Backslashes and Quotation Marks (z/OS to HP NonStop)

PROCESS1 is coded to be submitted from the z/OS node to run a program at the HP NonStop node. The RUN TASK statement executes FUP to copy ACCTJAN with shared access to the spooler, $S.#SPL1, at the HP NonStop node. Any HP NonStop process-related error messages are directed to $TERM1. Sterling Connect:Direct-related messages are directed to $S.#SPL1.

Note: This Process will only work in SNA environments.

PROCESS1 shows a Process with a parameter (PARM) continuing over multiple lines.

```
PROCESS1  PROCESS    PNODE=CD.OS390                                    -
                     SNODE=CD.HPNONSTOP                                -
                     SNODEID=(127.201,RSMITH)
STEP01    RUN TASK  (PGM = FUP                                        -
                     PARM=(\ "/OUT $S.#SPL1    \ ||                    -
                           \  ,TERM $TERM1/    \ ||                    -
                           \ COPY $SYSTEM.BILLPROC.ACCTJAN,,SHARE") \)  -
                     SNODE
```

The following Sterling Connect:Direct and HP NonStop syntax rules apply to both of these Processes:

- Within a Sterling Connect:Direct Process submitted from an z/OS node, single quotation marks or double quotation marks must be used to allow special characters to be embedded within a file name.
- The string of HP NonStop RUN command options must be enclosed in forward slashes (/). This is an HP NonStop syntax requirement.
- Because the PNODE is an z/OS node (that is, the Process is submitted on the z/OS node), backslashes and vertical bars must be used to continue a string across multiple lines.

  Bracketing backslashes are not valid when the PNODE is an HP NonStop node.

Use Concatenation Characters in a Run Task (z/OS to HP NonStop)

PROC1, PROC2, and PROC3 demonstrate the use of concatenation characters within a Sterling Connect:Direct for HP NonStop RUN TASK statement.

Note: These Processes will only work in SNA environments.

```
PROC1   PROCESS    SNODE=SYSCLX
STEP01  RUN TASK  (PGM=FUP                                  -
                   PARM=("/OUT $S.#TEST,TERM $S.#TMP/",     -
                         "COPY $A.SMITH.TACLCSTM,,SHARE"))  -
                   SNODE
```

PROC2 and PROC3 require concatenation characters because the parameters being passed to FUP are on multiple lines.

```
PROC2   PROCESS    SNODE=SYSCLX
STEP01  RUN TASK  (PGM=FUP                                      -
                   PARM=(\"/OUT $S.#TEST,NAME $FUP   \||        -
                         \,TERM $S.#TMP,PRI 150/     \||        -
                         \ COPY $A.SMITH.TACLCSTM,,FOLD")   \)  -
                   SNODE
```

TERM is coded on the RUN TASK statement for both PROC2 and PROC3 so Sterling Connect:Direct Processes can continue uninterrupted in the event the program being executed abends. If an abend occurs, any abend message will be sent to the device specified by the TERM command. If TERM is not coded, all abend messages will be directed to the terminal from which the Sterling Connect:Direct for HP NonStop system was started (HOMETERM). If HOMETERM is not paused, the abend message will not be displayed and the RUN TASK will hang until HOMETERM is paused.

```
PROC3   PROCESS    SNODE=SYSCLX
STEP01  RUN TASK  (PGM=FUP                                  -
                   PARM=(\"/OUT $S.#TEST,PRI 150     \||   -
                         \,TERM $S.#TMP,NAME $FUP/   \||   -
                         \ COPY $A.SMITH.TACLCSTM,,  \||   -
                         \ SHARE")  \)                      -
                   SNODE
```

Select Statistics for the Current Day (Sterling Connect:Direct for HP NonStop Run Task)

This Process performs a RUN TASK to select statistics for the current day. The resulting spooler file is then sent to a disk file on z/OS, where it can be printed or viewed.

Note: This Process will only work in SNA environments.

```
RUNT0005  PROCESS      SNODE=HPNONSTOP.CD
STEP1     RUN TASK    (PGM=NDMCOM                          -
                       PARM=('/OUT $S.#JOECOM/',           -
                             ' SEL STAT STARTT=(TODAY,)',  -
                             ' EXIT ' )                    -
                      ) SNODE
STEP2     COPY  FROM  (SNODE                               -
                       DSN=$S.#JOECOM                      -
                       DISP=OLD)                           -
                TO    (PNODE                               -
                       DSN=JOE.FILETEST.COMDOC             -
                       DISP=(NEW,CATLG)                    -
                       SPACE=(TRK,(2,1),RLSE)              -
                       DCB=(DSORG=PS,                      -
                            RECFM=FBA,                     -
                            LRECL=133,                     -
                            BLKSIZE=3990                   -
                            UNIT=SYSDA                     -
                            VOL=SER=M80006)                -
                      )
```

Define a VSE VSAM File and Copying a Sequential File from z/OS

This multistep Process, initiated from an z/OS node, consists of RUN TASK statements and a COPY statement. STEP1 runs the DMRTAMS utility to delete and then define a target VSAM cluster on a Sterling Connect:Direct for VSE/ESA node. STEP2 runs the DMRTDYN utility to unallocate the SYSOUT output data set generated by STEP1. STEP3 copies a sequential file from an z/OS node to a VSE node.

```
VSEVSAM  PROCES     PNODE=SC.OS390.NODE1 SNODE=SC.VSE.NODE1
STEP1    RUN TASK  (PGM=DMRTAMS,                                          -
                PARM=(C" MSG=YES DSN=SYSOUT.SYS011 DD=123 DISP=SHR",      -
                      C" DELETE VSE.VSAM.TEST CLUSTER              ",     -
                      C" DEFINE CLUSTER                          - ",     -
                      C" (NAME(VSE.VSAM.TEST)                    - ",     -
                      C"  RECORDS(25000 5000)                     - ",     -
                      C"  VOLUMES(VSAM01)                        - ",     -
                      C"  INDEXED                                - ",     -
                      C"  REUSE                                  - ",     -
                      C"  KEYS(8 6)                              - ",     -
                      C"  RECORDSIZE(262 880)                    - ",     -
                      C"  NOREPLICATE                            - ",     -
                      C"  SPANNED                                - ",     -
                      C"  SHR(2))                                - ",     -
                      C"  DATA(                                  - ",     -
                      C"  NAME(VSE.VSAM.TEST.DATA)               - ",     -
                      C"  CISZ(4096))                            - ",     -
                      C"  INDEX(                                 - ",     -
                      C"  NAME(VSE.VSAM.TEST.INDEX)              - ",     -
                      C"  CISZ(512))                               "      -
                     ))
STEP2    RUN TASK  (PGM=DMRTDYN                                           -
                    PARM=(C'UNALLOC DSN=SYSOUT.SYS011 DD=123')) SNODE
STEP3    COPY FROM (DSN=OS390.SEQ.DATASET                                 -
                    DISP=(SHR,KEEP)                                       -
                    PNODE)                                                -
              TO   (DSN=VSE.VSAM.TEST                                     -
                    DISP=NEW                                              -
                    DCB=(DSORG=VSAM)                                      -
                    SNODE)
```

Submit a Process from z/OS to Execute UNIX Commands

This Process initiates a Process from z/OS that executes commands on UNIX. The SYSOPTS string must be in the proper case for UNIX and enclosed in double quotation marks.

```
PROC2  PROCESS  SNODE=UNIX.NODE
STEP01 RUN TASK SNODE (PGM=UNIX) SYSOPTS="ls -lax > lsout.ncr"
```

Submit a Process from z/OS to Execute Microsoft Windows Programs

This Process initiates a Process from z/OS that runs a program on Microsoft Windows. The **desktop** parameter is set to YES to allow the program to interact with the desktop. The **sysopts** string is enclosed in backlashes.

```
PROC2   PROCESS    SNODE=NT.NODE                                          -
                   SNODEID=(puser01,ppswd01)
STEP01  RUN TASK  (PGM="NT") SNODE                                        -
                   SYSOPTS=\xd5 CMD(D:\CDCLNT\PROGPACK.EXE                -
                   D:\CDCLNT\PROGRAM.PAC\ D:\CDCLNT\*.*) DESKTOP (YES)'\
```

Use Run Task to Create a Save File on i5/OS

In this RUN TASK statement submitted from z/OS or z/OS, the first command CRTSAVF is used to create a save file named SAVFILE. The file has the text description CREATED BY PROCESS TEST. The second command SAVLIB saves the library TESTDT1 in the save file created in the first command.

```
STEP1  RUN TASK  (PGM=AS400)
                  SNODE
                  SYSOPTS=\"\
                          \CMD(\
                          \CRTSAVF\
                          \FILE(TEST/SAVEFILE)\
                          \TEXT('CREATED BY PROCESS TEST')
                          \)\
                          \CMD(\
                          \SAVLIB\
                          \LIB(TESTDT1)\
                          \DEV(*SAVF)\
                          \SAVF(TEST/SAVEFILE)\
                          \)\
                          \"\
```

Submit a Run Task from z/OS to i5/OS

In this RUN TASK statement submitted from z/OS, the CRTSAVF command creates a save file named SAVFILE. The file has the text description CREATED BY PROCESS TEST. The SAVLIB command saves the library TESTDT1 in the save file created in the first command.

```
STEP1  RUN TASK  (PGM=AS400)
                 SNODE
                 SYSOPTS=\"\
                         \CMD(\
                         \CRTSAVF\
                         \FILE(TEST/SAVEFILE)\
                         \TEXT('CREATED BY PROCESS TEST')
                         \)\
                         \CMD(\
                         \SAVLIB\
                         \LIB(TESTDT1)\
                         \DEV(*SAVF)\
                         \SAVF(TEST/SAVEFILE)\
                         \)\
                         \"\
```

z/OS RUN TASK to Execute a COBOL Program

The following example shows how to execute a COBOL program from a Sterling Connect:Direct Process by using the RUN TASK statement.

```
ALLOC1  RUN TASK (PGM=DMRTDYN                                              -
                  PARM=(C'ALLOC DSN=XYZ.TEMP DISP=SHR DD=OLD1')) PNODE
ALLOC2  RUN TASK (PGM=DMRTDYN                                              -
                  PARM=(C'ALLOC DSN=XYZ.TEMP2 DISP=SHR DD=OLD2')) PNODE
CUSTRT  RUN TASK (PGM=COBOL program name)
UNALLO1 RUN TASK (PGM=DMRTDYN                                              -
                  PARM=(C'UNALLOC DD=OLD1')) PNODE
UNALLO2 RUN TASK (PGM=DMRTDYN PARM=(C'UNALLOC DD=OLD2')) PNODE
```

Pass Mixed Case Parameters through an z/OS RUN TASK to DMRTSUB

z/OS RUN TASK parameters are always passed as upper case data, with the exception of parameters passed by RUN TASK to the DMRTSUB program. Mixed case parameters can be passed to DMRTSUB by using the SYSOPTS parameter.

The following example shows how to pass mixed case parameters to DMRTSUB through SYSOPTS in an z/OS RUN TASK statement.

```
STEP1  RUN TASK  (PGM=DMRTSUB )                          /* RUNTASK PGM   */ -
                  SYSOPTS=\"\                            /* START SYSOPTS */ -
                 'DSN=PROD.JCL.LIB(TESTPGM),DISP=SHR'    /* DATASET NAME  */ -
                 'FIRSTNM Bill'                          /* PARAMETER #1  */ -
                 'LASTNM  Smith'                         /* PARAMETER #2  */ -
                  \"\                                    /* END SYSOPTS   */ -
                  SNODE
```

The next example shows passing mixed case parameters to DMRTSUB using variable substitution. The use of single and double quotes are reversed from the previous example.

```
STEP1    RUN TASK (PGM=DMRTSUB )                  /* RUNTASK PGM   */ -
         SYSOPTS=\'\                              /* START SYSOPTS */ -
         "DSN=PROD.JCL.LIB(TESTPGM),DISP=SHR"     /* DATASET NAME  */ -
         "FIRSTNM &FIRST"                         /* PARAMETER #1  */ -
         "LASTNM  &LAST"                          /* PARAMETER #2  */ -
         \'\                                      /* END SYSOPTS   */ -
         SNODE
```

You must use SYSOPTS rather than PARM to pass mixed-case parameters.

Chapter 5. SUBMIT Statement Examples

Pass JES Job Name, Number, and User as a Process Name

In this example, the COPYPROC Process takes an existing file on the primary node (TEST.INPUT.FILE) and copies it to the secondary node naming the new file CD.jobID.jobname.submitjobuserID.FILE. The resultant file name contains the job number, job name, and the ID of the user who submitted the job for easy identification.

```
SAMPLE PROCESS COPYPROC
 COPYPROC PROCESS                              -
          SNODE=CD.SNODE                       -
          &VAR1=%JOBID                         -
          &JOBID=%JOBID                        -
          &JOBNM=%JOBNM                        -
          &JUSER=%JUSER
 *
 VAR1     COPY  FROM  (PNODE DSN=TEST.INPUT.FILE -
                       DISP=SHR  )             -
                 TO  (SNODE                    -
                       DSN=CD.&VAR1..FILE      -
                       DISP=(,CATLG)           -
                       SPACE=(CYL,(5,5),RLSE)  -
                       UNIT=SYSDA)
 JOBID    COPY  FROM  (PNODE DSN=TEST.INPUT.FILE -
                       DISP=SHR  )             -
                 TO  (SNODE                    -
                       DSN=CD.&JOBID..FILE     -
                       DISP=(,CATLG)           -
                       SPACE=(CYL,(5,5),RLSE)  -
                       UNIT=SYSDA)
 JOBNM    COPY  FROM  (PNODE DSN=TEST.INPUT.FILE -
                       DISP=SHR  )             -
                 TO  (SNODE                    -
                       DSN=CD.&JOBNM..FILE     -
                       DISP=(,CATLG)           -
                       SPACE=(CYL,(5,5),RLSE)  -
                       UNIT=SYSDA)
 JUSER    COPY  FROM  (PNODE DSN=TEST.INPUT.FILE -
                       DISP=SHR  )             -
                 TO  (SNODE                    -
                       DSN=CD.&JUSER..FILE     -
                       DISP=(,CATLG)           -
                       SPACE=(CYL,(5,5),RLSE)  -
                       UNIT=SYSDA)
```

To initiate the file transfer, you can use any of the following SUBMIT statements:

```
SUBMIT PROC=COPYPROC,NEWNAME=%JOBNM
```

```
SUBMIT PROC=COPYPROC,NEWNAME=%JOBID,&VAR1=%JOBNM
```

```
SUBMIT PROC=COPYPROC,&JOBNM=%JOBNM,&JUSER=%JUSER
```

Use SUBMIT with Symbolic Substitution (z/OS to z/OS)

In the following examples, a source file at the CD.LA node is to be copied to the CD.NEWYORK node, using CD.DALLAS as a store-and-forward node. To do this, two Processes need to be defined.

PROCESS1 is submitted from CD.LA. PROCESS1 performs the following functions:

- STEP01 copies the source file from CD.LA to CD.DALLAS.
- STEP02 within the same Process submits PROCESS2.

PROCESS2 executes at CD.DALLAS and contains one step which copies the file received from CD.LA and forwards it to CD.NEWYORK.

The user submits a Process and does not pass symbolics to the Process. Values for symbolics to be resolved on the SNODE are contained within the Process. The Process on the PNODE passes symbolics to a Process submitted on the SNODE.

Symbolics for PROCESS2 are supplied from values coded in PROCESS1.

The operator at CD.LA issues the following Sterling Connect:Direct command to initiate the file transfer.

```
SUB PROC = PROCESS1
```

PROCESS1 executes:

```
PROCESS1  PROCESS      PNODE=CD.LA                -
                       SNODE=CD.DALLAS            -
                       &DSN1=USER1.TEXT           -
                       &DSN2=USER1.NEW.TEXT       -
                       &PRTY=14                   -
                       NOTIFY=USER1
STEP01    COPY  FROM  (DSN=&DSN1 DISP=SHR PNODE)  -
                TO    (DSN=&DSN2 DISP=(NEW,CATLG) -
                       SNODE)
STEP02    SUBMIT  DSN=USER1.PROCESS.LIB(PROCESS2) -
                  PRTY=&PRTY                      -
                  SUBNODE=SNODE                   -
                  &DSN=&DSN2
```

PROCESS1 submits PROCESS2:

```
PROCESS2  PROCESS      PNODE=CD.DALLAS             -
                       SNODE=CD.NEWYORK
STEP01    COPY  FROM  (DSN=&DSN PNODE)             -
                TO    (DSN=USER1.NEW.TEXT1 DISP=SHR)
```

- PROCESS1 copies the file USER1.TEXT in LA to a file called USER1.NEW.TEXT at CD.DALLAS. Then it submits PROCESS2, which executes on the CD.DALLAS node because the SUBNODE parameter designates the SNODE, or CD.DALLAS, as the submitting node for PROCESS2.
- PROCESS2 copies the file ABC.NEW.TEXT at CD.DALLAS to file ABC.NEW.TEXT1 in CD.NEWYORK. The default DSN symbolic name for the PNODE is taken from the previous PROCESS1.

Use SUBMIT with the DSN Parameter (z/OS to z/OS)

In this example, symbolics for PROCESS2 are supplied by the operator submitting PROCESS1.

The operator at CD.LA issues the following Sterling Connect:Direct command to initiate the file transfer:

```
SUB PROC=PROCESS1 &DSN1=A345.DATA &DSN2=A345.NEW.DATA
```

PROCESS1 executes:

```
PROCESS1  PROCESS      PNODE=CD.LA  SNODE=CD.DALLAS  -
                       &DSN1=&DSN1                   -
                       &DSN2=&DSN2                   -
                       &PRTY=14                      -
                       NOTIFY=A345
STEP01    COPY  FROM  (DSN=&DSN1 DISP=SHR PNODE)     -
                TO    (DSN=&DSN2 DISP=SHR SNODE)
STEP02    SUBMIT  DSN=A345.PROCESS.LIB(PROCESS2)     -
                  PRTY=&PRTY                          -
                  SUBNODE=SNODE                       -
                  &DSN=&DSN2
```

PROCESS1 submits PROCESS2:

```
PROCESS2  PROCESS      PNODE=CD.DALLAS             -
                       SNODE=CD.NEWYORK
STEP01    COPY  FROM  (DSN=&DSN PNODE)             -
                TO    (DSN=A345.NEW.DATA1 DISP=SHR)
```

- PROCESS1 copies the file A345.DATA at CD.LA to a file called A345.NEW.DATA at the CD.DALLAS node, and then it submits PROCESS2, which executes on the CD.DALLAS node.
- PROCESS2 copies the file A345.NEW.DATA at the CD.DALLAS to the file A345.NEW.DATA1 at CD.NEWYORK.

Submit a Microsoft Windows Process from a UNIX Node

This Process copies a binary file from a Microsoft Windows node to a UNIX node. If the copy is successful, a **submit** of another process is performed on the CD.WIN node by setting the **subnode** parameter to SNODE.

```
proc1  process      snode=CD.WIN
                    &file1="d:\estdat\frunix.cfg"
                    &file2="/tdat/frnt.cfg"
copy1  copy  from  (file=&file1
                    sysopts="datatype(binary)"
                    snode)
             to    (file=&file2
                    pnode)
       if  (copy1 eq 0)  then
subpr1 submit  file="d:\estdat\nt2nwcp.cdp"
               Subnode=snode
       eif
   pend
```

Use SUBMIT at the Sterling Connect:Direct for HP NonStop Node

In this Process, STEP1 will execute FUP to purge $B.FILERESO.FILEA. STEP2 and STEP3 will submit Sterling Connect:Direct Processes at the HP NonStop node (PNODE).

```
SUBMIT  PROCESS     PNODE=CD.HP NONSTOP           -
                    SNODE=CD.OS390.NODE
STEP1   RUN  TASK  (PGM=FUP                      -
                    SYSOPTS=('/OUT $S.#FUP/',     -
                             'VOLUME $B.FILERESO', -
                             'PURGE FILEA ! '))
STEP2   SUBMIT  FILE=$B.DATA1.FILEB               -
                SUBNODE=PNODE
STEP3   SUBMIT  FILE=$B.DATA1.FILEC               -
                SUBNODE=PNODE
```

Use submit with the hold Parameter (UNIX to UNIX)

This Process shows how to submit another Sterling Connect:Direct Process, named copy.cd, which resides on the **snode**. The Process will be held in the Hold queue until it is explicitly released by a **change process** command.

```
submit1  process   snode=unix.node
step01   submit    file=copy.cd
                   subnode=snode
                   hold=yes
      pend
```

Use submit with the startt Parameter (UNIX to UNIX)

This Process shows how to submit another Sterling Connect:Direct Process, named copy.cd, which resides on the **pnode**. The Process will be held in the Timer queue until it is automatically released at 11:30 p.m. on December 14, 2005.

```
submit1  process   snode=unix.node
step01   submit    file=copy.cd
         startt=(12/14/1999,23:30)
 pend
```

Submit a Process to Run Upon Successful Completion of a Copy (HP NonStop to HP NonStop)

This Process copies USER1T.TEXT to $B.HP NONSTOP.FILE at the SNODE (CD.TAN.NODE). If $B.HP NONSTOP.FILE already exists, the file will be replaced. Upon completion of STEP01, $B.HP NONSTOP.FILE will run at the SUBNODE (CD.TAN.NODE).

```
PROCESS1  PROCESS      PNODE=CD.OS390.NODE SNODE=CD.TAN.NODE
                       PRTY=1
STEP01    COPY  FROM  (DSN=USER1T.TEXT DISP=SHR PNODE)
                TO    (DSN=$B.HP NONSTOP.FILE DISP=RPL SNODE)
STEP02    SUBMIT       DSN=$B.HP NONSTOP.FILE
                       PRTY=5
                       SUBNODE=SNODE
```

Submit a Process on OpenVMS

In this Process, STEP1 copies ACCT.VMSPROC to ACCTPROC.Sterling Connect:Direct in directory WORKDIR on device $DISK1 at the SNODE. STEP2 submits ACCTPROC.CD, which executes on the SNODE. The SNODE parameter specified in the SUBMIT statement overrides the value specified in the PROCESS statement.

```
PROC1   PROCESS     PNODE=CD.CA
                    SNODE=CD.ATLANTA
STEP1   COPY  FROM  (DSN=ACCT.VMSPROC
                    DISP=SHR
                    PNODE)
              TO    (DSN=$DISK1:[WORKDIR]ACCTPROC.CD
                    DISP=RPL
                    SNODE)
STEP2   SUBMIT      DSN=$DISK1:[WORKDIR]ACCTPROC.CD
                    SUBNODE=SNODE
                    SNODE=CD.CA
```

Submit a Process That Copies a File from One UNIX Node to Another, Then to a Third UNIX Node

This example shows how to copy a file from your local node to a node in Dallas, which then copies it to a node in Tampa.

A user issues a **submit** command to initiate a Process.

```
submit file=process1
```

The **submit** statement contained within **process1** specifies the Process name of **process2**. As a result, **process2** will also be submitted.

```
process1    process     snode=dallas
copystep    copy  from  (file=myfile)
                  to    (dsn=yourfile
                        disp=(new,keep))
substep     submit      file=process2
                        subnode=snode
                        &dsn=yourfile
             pend
```

The copy statement in **process2** copies **yourfile** to the file **newfile** in Tampa.

```
process2     process     snode=tampa
copystep     copy  from  (dsn=&dsn)
                   to    (dsn=newfile
                         disp=(new,keep))
             pend
```

The symbolic parameter **&dsn** is converted in process2 to the value of yourfile, which was specified in the **submit** statement in process1.

Submit a Process with Symbolics on VSE

This example shows using SUBMIT with the DSN parameter. Symbolics for PROCESS2 are supplied by the operator submitting PROCESS1.

The operator at the PNODE issues the following SUBMIT PROCESS command to initiate the file transfer:

```
SUB   PROC=PROCESS1                    -
      &DSN1=A345.DATA                  -
      &DSN2=A345.NEW.DATA
```

PROCESS1 executes:

```
PROCESS1  PROCESS      PNODE=CD.LA  SNODE=CD.DALLAS  -
                       &DSN1=&DSN1                   -
                       &DSN2=&DSN2                   -
                       &PRTY=14
COPYSTEP  COPY  FROM  (DSN=&DSN1 DISP=SHR PNODE)     -
                TO    (DSN=&DSN2 DISP=SHR SNODE)
SUBSTEP   SUBMIT  DSN=N(PROCESS2)                    -
                  PRTY=&PRTY                         -
                  SUBNODE=SNODE                      -
                  &DSN=&DSN2
```

PROCESS1 submits PROCESS2:

```
PROCESS2  PROCESS      PNODE=CD.DALLAS              -
                       SNODE=CD.NEWYORK
COPYSTEP  COPY  FROM  (DSN=&DSN2 PNODE)             -
                TO    (DSN=A345.NEW.DATA1 DISP=SHR)
```

- PROCESS1 copies the file A345.DATA in LA to a file called A345.NEW.DATA in Dallas. It then submits PROCESS2, which executes on the Dallas node. PROCESS2 is submitted with a PRTY of 14.
- PROCESS2 copies the file A345.NEW.DATA in Dallas to the file A345.NEW.DATA1 in New York.

Submit a Process Using Symbolics on VMESA

This example shows using SUBMIT with the DSN parameter. Symbolics for PROCESS2 are supplied by the operator submitting PROCESS1.

The operator at the PNODE issues the following Sterling Connect:Direct SUBMIT PROCESS command to initiate the file transfer:

```
SUB   PROC=PROCESS1          -
      &DSN1='TEST FILE'      -
      &DSN2='TEST FILE2'
```

In the following Process, CD.LA and CD.DALLAS are Sterling Connect:Direct for VM/ESA nodes.

```
PROCESS1  PROCESS      PNODE=CD.LA  SNODE=CD.DALLAS  -
                       &DSN1=&DSN1                   -
                       &DSN2=&DSN2                   -
                       &PRTY=14                      -
                       NOTIFY=A345
COPYSTEP  COPY  FROM  (DSN=&DSN1 DISP=SHR PNODE      -
                       LINK=(USER1,READ,RR,191))     -
                TO    (DSN=&DSN2 DISP=SHR SNODE      -
                       LINK=(USER2,WRITE,W,191))
SUBSTEP   SUBMIT  DSN=A345.PROCESS.LIB(PROCESS2)     -
                  PRTY=&PRTY                         -
                  SUBNODE=SNODE                      -
                  &DSN=&DSN2
```

PROCESS1 submits PROCESS2, where CD.DALLAS is a Sterling Connect:Direct for VM/ESA node and CD.NEWYORK is a Sterling Connect:Direct for z/OS node.

```
PROCESS2  PROCESS      PNODE=CD.DALLAS             -
                       SNODE=CD.NEWYORK
COPYSTEP  COPY  FROM  (DSN=&DSN2 PNODE             -
                       LINK=(USER2,READ,RR,191)    -
                TO    (DSN=A345.NEW.DATA1 DISP=SHR)
```

- PROCESS1 copies the file TEST FILE in LA to a file called TEST FILE2 in Dallas. It then submits PROCESS2, which executes on the Dallas node. PROCESS2 is submitted with a PRTY of 14.
- PROCESS2 copies the file TEST FILE2 in Dallas to the file A345.NEW.DATA1 in New York.

Submit a Process from UNIX to z/OS to Sterling Connect:Enterprise Using BATCHID

The following examples submit a Process from Sterling Connect:Direct for UNIX to Sterling Connect:Direct for z/OS, which then executes an MB#ADD01 function on Sterling Connect:Enterprise®.

The following is the Sterling Connect:Direct for UNIX Process:

```
set -v
ndmcli -x << EOJ
submit
proc1 process snode=ip addr;portnum  snodeid=(ROGER,ROGERpassword)
step1  submit file="$hlq.cd.PROCESS(MB#ADD01)"
         &FROMDSN=TEST.DATAFILE
         &RMTID=rmtid
         &BATCHID=TESTBATCH
         subnode=snode
pend;
```

The following is the z/OS Process called by the UNIX Process. This Process performs an MB#ADD01 on Sterling Connect:Enterprise:

```
MB#ADD01  PROCESS  SNODE=CD.OS390                              -
   &NOTIFY=' '                  /* Notify nobody            */ -
   &FROMDSN='''FROM.DSN.NAME'''  /* DATA SOURCE FILE NAME    */ -
   &FILETYP=TEXT                 /* DATA SOURCE FILE TYPE    */ -
   &PROFDSN='$hlq.CD.PROFILE'    /* PROFILE DSN              */ -
   &PROFMEM=SUBADD01             /* EXTRACT PROFILE MEMBER   */ -
   &USER=%USER                   /* ORIGINATOR ID            */ -
   &SEQ=%NUM1                    /* UNIQUE NUMBER            */ -
```

```
   &RMTID=RMTID                  /* RDX BATCH RMTID          */ -
   &MBNAME=E100                  /* RDX NAME                 */ -
   &LOCAPPL=MBICO                /* RDX LOCAL APPLID         */ -
   &MBAPPL=SBLDUB74              /* RDX APPLID               */ -
   &LOGMODE=TESTLU62             /* RDX LOG MODE TABLE       */ -
   &USERID=USERID                /* RDX USERID               */ -
   &PASSWRD=PASS                 /* RDX PASSWORD             */ -
   &BATCHID=BATCH.ID             /* RDX BATCH ID             */ -
   &BATCHNO=0                    /* RDX BATCH NUMBER         */ -
   &BUFSIZE=0                    /* VTAM BUFFER SIZE         */
*
   SYMBOL  &HLQ1=\CD.\ || &RMTID
   SYMBOL  &SEQ1=\T\ || &SEQ
   SYMBOL  &ADSN=\&HLQ1\ || \.STOUT0.INFILE.\ || &SEQ1
   SYMBOL  &IDSN=\&HLQ1\ || \.STOUT0.SYSIN.\ || &SEQ1
   SYMBOL  &PDSN=\&HLQ1\ || \.STOUT0.SYSPRINT.\ || &SEQ1
   SYMBOL  &LDSN=\&HLQ1\ || \.STOUT0.LOGFILE.\ || &SEQ1
   SYMBOL  &SDSN=\&HLQ1\ || \.STOUT0.SNAPOUT.\ || &SEQ1
*
   SYMBOL  &BATCHID2=\'''\ || &BATCHID || \'''\
*
* COPY FILE FROM SOURCE NODE
*
MBA$COPY COPY   FROM(SNODE DSN=&FROMDSN   DISP=SHR)              -
                TO  (      DSN=&ADSN      DISP=(RPL,CATLG)       -
                                          TYPE=&FILETYP)
*
 IF (MBA$COPY > 0) THEN
   RUN TASK (PGM=DMIONOTE,                                       -
             PARM=(ADD, FAIL, "&FROMDSN", &RMTID, "&BATCHID2",   -
                   MBA$COPY, &NOTIFY)) PNODE
   EXIT
 EIF
*
* CREATE THE TEMPORARY FILE FOR USE BY ADD
*
MBA$CREI RUN TASK (PGM=DMRTDYN                                   -
                   PARM=(C'ALLOC',                               -
                         C" DSN=&IDSN"                           -
                         C' DISP=(NEW,CATLG)',                   -
                         C' SPACE=(TRK,(1,1))',                  -
          C' DCB=(RECFM=FB,DSORG=PS,LRECL=80,BLKSIZE=800)',      -
                         C'UNIT=SYSDA',                          -
                         F'-1',                                  -
                         C'UNALLOC',                             -
                         C" DSN=&IDSN")) PNODE
*
 IF (MBA$CREI > 0) THEN
   RUN TASK (PGM=DMIONOTE,                                       -
             PARM=(ADD, FAIL, "&FROMDSN", &RMTID, "&BATCHID2",   -
                   MBA$CREI,  &NOTIFY)) PNODE
   EXIT
 EIF
*
MBA$CREP RUN TASK (PGM=DMRTDYN                                   -
                   PARM=(C'ALLOC',                               -
                         C" DSN=&PDSN"                           -
                         C' DISP=(NEW,CATLG)',                   -
                         C' SPACE=(TRK,(1,1))',                  -
       C' DCB=(RECFM=FBA,DSORG=PS,LRECL=133,BLKSIZE=2660)',      -
                         C'UNIT=SYSDA',                          -
                         F'-1',                                  -
                         C'UNALLOC',                             -
                         C" DSN=&PDSN")) PNODE
*
 IF (MBA$CREP > 0) THEN
   RUN TASK (PGM=DMIONOTE,                                       -
```

```
            PARM=(ADD, FAIL, "&FROMDSN", &RMTID,  "&BATCHID2",   -
                  MBA$CREP, &NOTIFY)) PNODE
/* GOTO MBA$DELI   */
 EXIT
 EIF
*
MBA$CREL RUN TASK (PGM=DMRTDYN                                    -
                  PARM=(C'ALLOC',                                 -
                        C" DSN=&LDSN"                             -
                        C' DISP=(NEW,CATLG)',                     -
                        C' SPACE=(TRK,(1,1))',                    -
        C' DCB=(RECFM=FB,DSORG=PS,LRECL=196,BLKSIZE=1960)',       -
                        C'UNIT=SYSDA',                            -
                        F'-1',                                    -
                        C'UNALLOC',                               -
                        C" DSN=&LDSN")) PNODE
*
 IF (MBA$CREL > 0) THEN
   RUN TASK (PGM=DMIONOTE,                                        -
            PARM=(ADD, FAIL, "&FROMDSN", &RMTID, "&BATCHID2",    -
                  MBA$CREL, &NOTIFY)) PNODE
/* GOTO MBA$DELP  */
   EXIT
 EIF
*
MBA$CRES RUN TASK (PGM=DMRTDYN                                    -
                  PARM=(C'ALLOC',                                 -
                        C" DSN=&SDSN"                             -
                        C' DISP=(NEW,CATLG)',                     -
                        C' SPACE=(TRK,(1,1))',                    -
        C' DCB=(RECFM=FB,DSORG=PS,LRECL=196,BLKSIZE=1960)',       -
                        C'UNIT=SYSDA',                            -
                        F'-1',                                    -
                        C'UNALLOC',                               -
                        C" DSN=&SDSN")) PNODE
*
 IF (MBA$CRES > 0) THEN
   RUN TASK (PGM=DMIONOTE,                                        -
            PARM=(ADD, FAIL, "&FROMDSN",  &RMTID, "&BATCHID2",   -
                  MBA$CRES, &NOTIFY)) PNODE
/* GOTO MBA$DELP   */
   EXIT
 EIF
*
* EXECUTE THE ADD TO CONNECT:Enterprise RDX
*
MBA$STO0 RUN TASK (PGM=DMSTOUT0,                                  -
                  PARM=(C'ADD'                                    -
                        C"&PROFDSN"                               -
                        C'&PROFMEM'                               -
                        C"&IDSN"                                  -
                        C"&PDSN"                                  -
                        C"&ADSN"                                  -
                        C"&LDSN"                                  -
                        C'&SDSN'                                  -
                        C'&RMTID'                                 -
                        C'&BATCHID2'                              -
                        C'&BATCHNO'                               -
                        C'&MBNAME'                                -
                        C'&LOCAPPL'                               -
                        C'&MBAPPL'                                -
                        C'&LOGMODE'                               -
                        C'&USERID'                                -
                        C'&PASSWRD'                               -
                        C'&BUFSIZE')) PNODE
*
  IF (MBA$STO0 > 4) THEN
```

```
RUN TASK (PGM=DMIONOTE,                                          -
          PARM=(ADD, FAIL, "&FROMDSN", &RMTID, "&BATCHID2",    -
                MBA$STO0, &NOTIFY)) PNODE
  ELSE
*
*
* NOTIFY ORIGINATOR OF SUCCESS
 RUN TASK (PGM=DMIONOTE,                                          -
           PARM=(ADD, GOOD, "&FROMDSN", &RMTID, "&BATCHID2",    -
                 MBA$STO0, &NOTIFY)) PNODE
  EIF
*
* DELETE TEMPORARY FILES
* ...
```

Chapter 6. Conditional Statements Examples

Use of Conditional Statements (OpenVMS to z/OS)

This Process uses conditional logic to check the completion code of STEP01. If the completion code is equal to 0, STEP03 copies a file from OpenVMS to z/OS. If the completion code is greater than 0, STEP04 initiates the RUN TASK statement to notify the OpenVMS user that the copy was unsuccessful.

```
PROC1   PROCESS      SNODE=XYZNODE                                  -
                     SNODEID=(JSMITH,JERRY)
STEP01  COPY  FROM  (DSN=JSMITH.DATA                                -
                     SNODE)                                         -
              TO    (DSN=U1:[RJONES.CDTEST]RUSS.DAT                -
                     DISP=RPL)
STEP02  IF   (STEP01 = 0)  THEN
STEP03  COPY  FROM  (DSN=U1:[RJONES.DATA_FILES]TEXT.DAT)           -
              TO    (DSN=JSMITH.YES SNODE                          -
                     DCB=(DSORG=PS,RECFM=V) DISP=RPL)
        ELSE
STEP04  RUN TASK    (PGM=VMS)  PNODE SYSOPTS="                     -
                     CMD = 'REPLY/USER=RJONES COPY_UNSUCCESSFUL'"
        EIF
```

Use of Conditional Logic (z/OS to VSE)

The following Process issues a VSE/Power command based on the return code of the previous COPY step. If the transfer is successful, the VSE/Power command releases GSVSE01. If the transfer fails, GSVSE02 is released. This Process is designed to be submitted from the z/OS node.

```
TESTVSE  PROCESS      SNODE=CD.VSE.NODE             -
                      PNODE=CD.OS390.NODE
STEP01   COPY  FROM  (DSN=JSMITH.CNTL.JCL(VSAMDEL)  -
                      DISP=SHR                      -
                      UNIT=SYSDA)                   -
                      CKPT=0K                       -
               TO    (DSN=JSMITH.CNTL.JCL(XXXX)    -
                      DISP=RPL                      -
                      LIBR=(SUBLIB=JCL,TYPE=JCL)    -
                      DCD=(DSORG=VSAM)
     IF (STEP01 NE 0) THEN
         RUN TASK    (PGM=DMRTPOWR                 -
                      PARM=(F'5'                    -
                            F'3'                    -
                           ('R RDR,GSVSE01'))       -
                      SNODE
                      EXIT
     ELSE
         RUN TASK    (PGM=DMRTPOWR                 -
                      PARM=(F'5'                    -
                            F'3'                    -
                           ('R RDR,GSVSE02'))       -
                      SNODE
                      EXIT
     EIF
```

Use of Conditional Logic (i5/OS to z/OS)

This Process copies a file from an Sterling Connect:Direct for i5/OS node to z/OS. STEP02 checks the completion code from STEP01 and executes STEP04 if the completion code is greater than 0. If STEP01 receives a completion code of 0, STEP03 will execute and send a message to terminal DSP03 indicating the file transfer was successful. If STEP01 receives a completion code greater than 0, STEP04 will execute and send a message to terminal DSP03 indicating the file transfer failed.

```
COPY02  PROCESS  SNODE=CD.SANFRAN.AS400                              -
                 PRTY=8                                             -
                 NOTIFY=USERID                                      -
                 CLASS=4                                            -
                 SNODEID=(USERID,PSWD)
STEP01  COPY     FROM    (                                          -
                 SNODE                                              -
                 DSN='OS400/FILE001'                                -
                 SYSOPTS="TYPE(MBR)"                                -
                 DISP=SHR                                           -
                )                                                   -
        TO       (                                                  -
                 PNODE                                              -
                 DSN=OS390.FILE002                                  -
                 DCB=(DSORG=PS,BLKSIZE=6160,LRECL=080,RECFM=FB)     -
                 DISP=RPL                                           -
                )
STEP02  IF      (STEP01 > 0) THEN
        GOTO     STEP04
        ELSE
STEP03  RUN TASK  (PGM=OS400) SNODE                                 -
                   SYSOPTS=\"\                                      -
                           \CMD(\                                   -
                           \SNDBRKMSG\                              -
                           \MSG('FILE TRANSFER SUCCESSFUL')\        -
                           \TOMSGQ(DSP03)\                          -
                           \)\                                      -
                           \"\
        EIF
        EXIT
STEP04  RUN TASK  (PGM=OS400) SNODE                                 -
                   SYSOPTS=\"\                                      -
                           \CMD(\                                   -
                           \SNDBRKMSG\                              -
                           \MSG('FILE TRANSFER FAILED!!  ')\        -
                           \TOMSGQ(DSP03)\                          -
                           \)\                                      -
                           \"\
```

Use of Conditional Logic (HP NonStop to z/OS)

In this multi-step Process, STEP01 will execute FUP to purge files FILE1, FILE2, FILE3, and FILE4 from $B.FILERESO on the PNODE. A message will be sent to the spooler ($S.#FUPTEST) that indicates whether FUP executed successfully.

STEP02 will copy DATA1.FILEA from the z/OS node (SNODE) to $B.FILERESO.FILE1 at the PNODE. The file will default to the same type of file being copied.

Conditional logic in STEP03 is then used to check the completion code of STEP02. If the completion code is greater than 4, then STEP04 will execute. If the

completion code from STEP02 is 4 or less, then DATA1.FILEB will be copied from the z/OS node to $B.FILERESO.FILE2 at the HP NonStop node.

STEP04 will then execute and copy DATA1.FILEC from the z/OS node to $B.FILERESO.FILE3 at the HP NonStop node. If the completion code is greater than 8, then no further processing will occur. If the completion code of STEP03 is greater than 4, DATA1.FILED will be copied from the z/OS node to $B.FILERESO.FILE4 at the HP NonStop node.

```
COND1   PROCESS    PNODE=CD.HP NONSTOP                              -
                   SNODE=CD.OS390.NODE
STEP01  RUN TASK  (PGM=FUP                                          -
                   PARM=('/OUT $S.#FUPTEST/',                       -
                         'VOLUME $B.FILERESO',                      -
                         'PURGE FILE1!  ',                          -
                         'PURGE FILE2!  ',                          -
                         'PURGE FILE3!  ',                          -
                         'PURGE FILE4!  '))
STEP02  COPY  FROM  (DSN=DATA1.FILEA  DISP=SHR  SNODE)              -
              TO    (DSN=$B.FILERESO.FILE1 DISP=NEW  PNODE)
STEP03  IF   (STEP02 GT 4) THEN
              GOTO STEP04
        ELSE
              COPY  FROM  (DSN=DATA1.FILEB  SNODE)                  -
                    TO    (DSN=$B.FILERESO.FILE2  DISP=NEW PNODE)
        EIF
STEP04  COPY  FROM  (DSN=DATA.FILEC  SNODE)                         -
              TO    (DSN=$B.FILERESO.FILE3  DISP=NEW  PNODE)
        IF   (STEP03 GT 8) THEN
        EXIT
        EIF
        IF   (STEP03 LT 4) THEN
              COPY  FROM  (DSN=DATA1.FILED SNODE)                   -
                    TO    (DSN=$B.FILERESO.FILE4 DISP=NEW PNODE)
        EIF
```

Use of Conditional Statements to Test for Process Completion on OpenVMS

In this example, the Process runs a job on OpenVMS. It uses conditional statements to check the completion code from STEP01. If the completion code equals 0, a message is issued to the operator indicating that the command procedure was successful. If the completion code is any value other than 0, a message is issued to the operator indicating that the command procedure failed.

```
PROC01      PROCESS     SNODE=CD.VMS.NODE
STEP01      RUN JOB     (DSN=DISK1:[USER1.PROC_CD1]DIR.COM
                        SYSOPTS="NOPRINT LOG WAIT" SNODE
STEP02      IF          (STEP01 EQ 0) THEN
            RUN TASK    (PGM=VMS PNODE  SYSOPTS="
                        CMD='REPLY/USER=USER1/BELL BATCH_JOB_SUCCEED'"
            ELSE
            RUN TASK    (PGM=VMS PNODE  SYSOPTS="
                        CMD='REPLY/USER=USER1/BELL BATCH_JOB_FAILED'"
            EIF
```

Use of Conditional Statement on OpenVMS

In this example, the Process runs a job on OpenVMS. It uses conditional statements to check the completion code from STEP01. If the completion code equals 0, a message is issued to the operator indicating that the command procedure was successful. If the completion code is any value other than 0, a message is issued to the operator indicating that the command procedure failed.

```
PROC01      PROCESS     SNODE=CD.VMS.NODE
STEP01      RUN JOB     (DSN=DISK1:[USER1.PROC_CD1]DIR.COM
                        SYSOPTS="NOPRINT LOG WAIT" SNODE
STEP02      IF          (STEP01 EQ 0) THEN
            RUN TASK    (PGM=VMS PNODE  SYSOPTS="
                        CMD='REPLY/USER=USER1/BELL BATCH_JOB_SUCCEED'"
            ELSE
            RUN TASK    (PGM=VMS PNODE  SYSOPTS="
                        CMD='REPLY/USER=USER1/BELL BATCH_JOB_FAILED'"
            EIF
```

Use of Conditional Statements in a UNIX Process

This Sterling Connect:Direct for UNIX Process contains all of the conditional statements.

```
copy1     process     snode=chicago
step01    copy  from  (file=abc pnode)
                to    (file=def snode)
step02    if          (step01 gt 4) then
          goto step07
          else
step03    runjob      sysopts="myjob"
          eif
step04    if          (step03 >= 8) then
          exit
step05    if          (step03 lt 4) then
step06    copy  from  (file=xyz pnode)
                to    (file=uvw snode)
          eif
          exit
step07    run task    sysopts="verify"
          exit
          pend
```

copy01 is the **process** statement defining the secondary node as chicago.

step01 copies file abc on the **pnode** to file def on the **snode**.

step02 checks the completion code of step01. If step01 fails (return code greater than **4**), step07 executes. If step01 completes with a return code of **4** or less, step03 executes.

step03 submits a job, myjob, on the **pnode**.

step04 checks the completion code of step03. If step03 fails with a code of 8 or greater, the Process terminates. Otherwise, step05 executes.

step05 checks the completion code from step03. If less than **4**, indicating the step completed without error, step06 executes.

step06 copies file xyz on the **pnode** to file uvw on the **snode**. The Process will then exit.

step07 only executes if step01 fails. The program verify runs, sending an Operation Failed Message to the console operator.

pend marks the end of a Process.

Use of Conditional Statements in VMESA

The following example contains all conditional VMESA statements. A description of each step follows the example Process.

```
COPY01  PROCESS      SNODE=CD.CHICAGO
STEP01  COPY  FROM  (DSN=ABC.FILEA PNODE)      -
              TO    (DSN=JKL.FILEA)
STEP02  IF   (STEP01 GT 4) THEN
              GOTO STEP07
        ELSE
STEP03  RUN JOB     (DSN=USERJOB) SNODE
        EIF
STEP04  IF   (STEP03 >= 8) THEN
              EXIT
        EIF
STEP05  IF   (STEP03 LT 4) THEN
STEP06  COPY  FROM  (DSN=ABC SNODE)            -
              TO    (DSN=MNO PNODE)
        EIF
        EXIT
STEP07  RUN TASK    (PGM=DMNOTIFY,             -
                     PARM=('FAIL',ABC.FILEA))  -
                     PNODE
```

COPY01 is the **PROCESS** statement defining the secondary node as CD.CHICAGO.

STEP01 copies file ABC.FILEA on the PNODE to file JKL.FILEA on the SNODE.

STEP02 checks the completion code of STEP01. If STEP01 fails, STEP07 executes. If STEP01 ended with a completion code of **4** or less, STEP03 executes.

STEP03 submits the job, USERJOB, on the SNODE.

STEP04 checks the completion code of STEP03. If STEP03 fails with a completion code of **8** or greater, the Process terminates. Otherwise, STEP05 executes.

STEP05 checks the completion code from STEP03. If less than **4**, indicating the step completed without errors, the COPY statement in STEP06 executes and the Process terminates.

STEP06 copies file ABC on the SNODE to file MNO on the PNODE.

STEP07 only executes if STEP01 fails. The program DMNOTIFY runs, sending an OPERATION FAILED message to the console operator.

Use of Conditional Statements in a VSE Process

The following VSE Process contains all of the conditional statements. A description of each step follows.

```
COPY01  PROCESS      SNODE=CD.CHICAGO
STEP01  COPY  FROM  (DSN=ABC.FILEA                                    -
                     DCB=(DSORG=PS, BLKSIZE=800, LRECL=80, RECFM=FB  -
                          VOL=SER=PACK01 DISP=(SHR) PNODE)           -
              TO    (DSN=JKL.FILEA                                    -
                     VOL=SER=PACK02                                   -
                     SPACE=(1993, (15)) DISP=(NEW) SNODE
STEP02  IF  (STEP01 GT 4) THEN
             GOTO STEP07
        ELSE
STEP03  RUN JOB     (DSN=USERJOB) SNODE
        EIF
STEP04  IF  (STEP03 >= 8) THEN
             EXIT
        EIF
STEP05  IF  (STEP03 LT 4) THEN
STEP06  COPY  FROM  (DSN=ABC                                          -
                     DCB=(DSORG=PS, BLKSIZE=800, LRECL=80, RECFM=FB  -
                     VOL=SER=PACK01 DISP=(SHR) PNODE)                 -
              TO    (DSN=MNO                                          -
                     VOL=SER=PACK02                                   -
                     SPACE=(2008, (15)) DISP=(NEW) SNODE
        EIF
        EXIT
STEP07  RUN TASK    (PGM=DMNOTIFY,                                    -
                     PARM=('FAIL',ABC.FILEA))                         -
                     PNODE
```

COPY01 is the **PROCESS** statement defining the secondary node as CD.CHICAGO.

STEP01 copies file ABC.FILEA on the PNODE to file JKL.FILEA on the SNODE.

STEP02 checks the completion code of STEP01. If STEP01 fails, STEP07 executes. If STEP01 ended with a completion code of **4** or less, STEP03 executes.

STEP03 submits the job, USERJOB, on the SNODE.

STEP04 checks the completion code of STEP03. If STEP03 fails with a completion code of **8** or greater, the Process terminates. Otherwise, STEP05 executes.

STEP05 checks the completion code from STEP03. If less than **4**, indicating the step completed without errors, the COPY statement in STEP06 executes and the Process terminates.

STEP06 copies file ABC on the SNODE to file MNO on the PNODE.

STEP07 only executes if STEP01 fails. The program DMNOTIFY runs, sending a message indicating the operation failed to the console operator.

Use of Conditional Statements in a Microsoft Windows Process

This Microsoft Windows Process contains all of the conditional statements.

```
copy01     process     snode=cdchicago
step01     copy  from  (file=myfile1 pnode)
                 to    (file=yourfile1 snode)
step02     if          (step01 gt 4) then
                       goto step07
           else
step03     runjob      dsn=WINNT
                       sysopts="pgm(testwin.exe)"
           eif
step04     if          (step03 >= 8) then
           exit
           eif
step05     if          (step03 lt 4) then
step06     copy  from  (file=myfile2 pnode)
                 to    (file=yourfile2 snode)
           eif
           exit
step07     run task     pgm=Windows
                        sysopts="pgm(failjob.exe)"
           exit
```

copy01 is the **process** statement defining the Process name as copy01 and the snode as cdchicago.

step01 copies file myfile1 on the pnode to file yourfile1 on the snode.

step02 checks the completion code of step01. If step01 fails (return code greater than 4), step07 executes. If step01 completes with a return code of **4** or less, step03 executes.

step03 starts the program testwin.exe.

step04 checks the completion code of step03. If step03, not the program testwin.exe, fails with a code of **8** or greater, the Process terminates. Otherwise, step05 executes.

step05 checks the completion code from step03. If less than **4**, indicating the step completed without error, not the program testwin.exe, step06 executes.

step06 copies file myfile2 on the pnode to file yourfile2 on the snode. The Process will then exit.

step07 only executes if step01 fails. The **run task** step executes the program failjob.exe.

Chapter 7. pend Statement Examples

Example UNIX pend Statement

The following Process copies a file from a local node to a node named Atlanta. The **pend** statement indicates the end of the Process.

```
copyseq  process      snode=atlanta
step01   copy  from  (file=myfile)
               to    (dsn=yourfile)
         pend
```

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2012. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2012.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise, Gentran®, Gentran®:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce™, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Printed in USA