PN 00448

Connect:Direct®

Process Concepts and Examples Guide

sterling commerce

PN 00448

Connect:Direct®

Process Concepts and Examples Guide

sterling commerce

Connect:Direct Process Concepts and Examples Guide
First Edition
February 2004

CDPRCON402

Contents

Preface

The *Connect:Direct Concepts and Examples Guide* provides the information you need to write Connect:Direct Processes. This guide introduces you to the Process language and includes examples of syntax, statements, and Connect:Direct Processes that illustrate the full functionality of Connect:Direct.

Connect:Direct Process Documentation

The Connect:Direct Process documentation consists of the following publications:

- *Connect:Direct Process Concepts and Examples Guide* provides an overview of Connect:Direct, describes the general structure and syntax rules for the Process language, and includes numerous examples.
- *Connect:Direct Process Statements Guide* describes the Process statements for the following platforms:
  - Connect:Direct OS/390
  - Connect:Direct VM/ESA
  - Connect:Direct VSE
  - Connect:Direct Tandem NonStop Kernel
  - Connect:Direct OS/400
  - Connect:Direct OpenVME
  - Connect:Direct OpenVMS
  - Connect:Direct UNIX
  - Connect:Direct Windows

The Connect:Direct Process documentation assumes knowledge of the operating systems for Connect:Direct products. If you are not familiar with a specific operating system, including its applications, network, and environment, refer to the appropriate library of manuals. For additional information about Connect:Direct, refer to the library of Connect:Direct documentation.

Notational Conventions

The Connect:Direct Process documentation set uses certain notational conventions. This following table describes the conventions used the documentation set.

Notation	Description
Uppercase Letters	Uppercase letters in the statement format indicate that you type in information as shown. For Connect:Direct UNIX, and OpenVME: Values must be in the proper case for the node on which they will be processed. For Connect:Direct Windows: This convention does not apply.
Uppercase and Lowercase Letters	A statement in uppercase letters followed by lowercase letters indicates an alternative to typing the entire statement. For example, PROCess means that you need only type `PROC` for the statement to be valid. For Connect:Direct UNIX, OpenVME and Windows: This convention does not apply.
Lowercase Letters	Lowercase letters or words in statements or syntax boxes require substitution by the user. For example, PNODE=primary-node-name indicates that you must provide the name of the primary node. For Connect:Direct UNIX, OpenVME, and Windows: This convention does not apply. Refer to Italic Letters for the notational convention used for variable substitution for these products.
Bold Letters	Bold print in syntax boxes indicates required labels, statements, and parameters. For example, **DSN=filename** indicates that the parameter DSN is required. For Connect:Direct UNIX, OpenVME, and Windows: In text, parameters are shown in boldface characters to differentiate them from surrounding text.
Italic Letters	For Connect:Direct UNIX, OpenVME and Windows, italic letters act as placeholders for information you must provide. For example, *dsn=filename* indicates that you would type the actual name for a file instead of the word shown in italic type.
Underlined Letters	Underlining indicates default values for parameters and subparameters. For example, RETAIN=Yes|<u>No</u>|Initial specifies that the default for RETAIN is No.
Vertical Bars	Vertical bars indicate that you can supply one of a series of values separated by the vertical bars. For example HOLD=Yes|No|Call specifies that Yes or No or Call is valid.
Brackets	Brackets indicate optional information within an optional parameter. For example, STARTT=([date|day][,hh:mm:ssXM]) indicates that you can specify either a date or a day, a date or a day plus a time, or just a time. For Connect:Direct OpenVMS: OpenVMS uses brackets in its directory naming convention. Because brackets are not available on most keyboards on IBM systems, use less than and greater than signs (< >) in a Connect:Direct Process that is submitted from the IBM node.

Notation	Description
Horizontal Ellipsis	Horizontal ellipsis indicates that the preceding item may be repeated. For example, &symbolic_name_2=variable-string-2. . . indicates that you can specify multiple symbolic parameters.
Vertical Ellipsis	Vertical ellipsis indicates that not all of the data is displayed. Typically, the vertical series of ellipses is used in examples.
Additional Notations	Code all commas and parentheses as they appear. Process with a capital P refers to a Connect:Direct Process. Monospaced characters (characters of equal width) represent information for screens, commands, Processes, and reports.

Getting Support for Sterling Commerce Products

Sterling Commerce provides intuitive technical products and superior Help and documentation to enable you to work independently. However, if you have a technical question about a Sterling Commerce product, use the Sterling Commerce Customer Support Web site.

The Sterling Commerce Customer Support Web site at www.sterlingcommerce.com is the doorway to Web support, information, and tools. This Web site contains several informative links, including a solutions database, an issue tracking system, fix information, documentation, workshop information, contact information, sunset and retirement schedules, and ordering information. Refer to the Customer Support Reference Guide at www.sterlingcommerce.com/customer/tech_support.html for specific information on getting support for Sterling Commerce products.

Chapter 1

About Connect:Direct Processes

A Connect:Direct Process is a collective unit of work. Steps in each work unit are defined using a scripting language unique to the Connect:Direct product. The Connect:Direct Process language consists of statements and parameters that provide instructions for initiating such activities as:

- ✦ Copying files between systems
- ✦ Running jobs, programs, and commands
- ✦ Handling error situations through conditional logic
- ✦ Starting another Process

Structure and syntax of Connect:Direct Process statements are discussed in Chapter 2, *Process Statement Structure and Syntax*.

An integral part of Connect:Direct are commands that allow users to submit, monitor, and control the execution of Connect:Direct Processes. For command usage and syntax, refer to the Connect:Direct user's guide for your operating environment.

Description of Process Statements

A Connect:Direct Process consists of a Process definition statement (PROCESS statement) and one or more additional Connect:Direct statements that may be coded in any sequence. Parameters are specified to further qualify Process instructions. The following table defines Connect:Direct Process statements:

Statement	**Description**
PROCESS statement	defines general Process characteristics, including identification of the remote node, and is always the first statement in a Process.
COPY statement	initiates a data transfer. COPY parameters include source and destination file names and attributes.
RUN JOB statement	submits work to the host operating system that executes asynchronous to the remaining steps in a Process. The work can execute on either the local or remote node.

Statement	Description
RUN TASK statement	executes a program or command within the Connect:Direct environment synchronous to the remaining steps in a Process. The program or command can execute on either the local or remote node. User programs can be run in different environments.
SUBMIT statement	allows one Process to be submitted from within another executing Process. The Process can execute on either the local or remote node.
SYMBOL statement	allows symbolic substitution of Process parameter values.
Conditional statements	controls Process step execution by allowing conditional tests of Process step return codes using IF THEN, ELSE, EIF, GOTO, and EXIT conditional logic statements.
pend statement (valid only for Connect:Direct UNIX, OpenVME, and Windows)	marks the end of a Process. There are no parameters associated with the pend statement.

Writing Connect:Direct Processes

When writing a Process, use the PROCESS, SYMBOL, and Conditional statements for the operating environment from which you are initiating the Process. For example, if you are writing a Connect:Direct Process to copy a file from UNIX to OS/390 and the Process is submitted from the UNIX node, refer to the structure of the PROCESS statement for Connect:Direct UNIX.

For Connect:Direct OS/390, VM/ESA, VSE/ESA: The maximum storage area allowed for a Process statement is 64K. To accommodate a larger Process, split the Process in half and include a SUBMIT statement in the first Process to run the second Process.

Use the COPY FROM and COPY TO formats for the appropriate FROM and TO platforms. For example, if you are writing a Process to copy a file from UNIX to OS/390, refer to the COPY FROM information for Connect:Direct UNIX and the COPY TO information for Connect:Direct OS/390.

Use the RUN JOB and RUN TASK formats for the platform on which the job will execute. For example, if you are submitting a Process on OpenVMS that will submit a batch job on OS/390, refer to the RUN JOB format for Connect:Direct OS/390.

Use the SUBMIT statement format for the platform on which the Process will execute. If you are writing a Process on OS/390 that submits another Process on UNIX, refer to the structure of the SUBMIT statement for Connect:Direct UNIX.

Refer to the *Example Processes* part in this volume for examples of activities that can be performed in various operating environments.

Summary of Connect:Direct Process Submission

A Process can be constructed and submitted for execution in several ways, depending upon the user interface(s) available on your operating environment. These Connect:Direct user interfaces include command line, interactive (online), operator background (batch) utilities, and user-written programs through the Connect:Direct Application Program Interface (API).

When a Process is submitted, the following tasks are performed:

Step 1-User Submits a Process

The user creates and submits a new Process or submits a predefined Process from a Connect:Direct Process library.

Step 2-Parsing for Correct Syntax

The parser checks the syntax of the Process.

Step 3-Process Queued for Execution and Submit Message Issued

If the Process passes syntax checking, it is placed in an appropriate work queue according to Process parameters, such as priority, class, and start time. The Connect:Direct work queues are jointly referred to as the Transmission Control Queue (TCQ) or the process queue. A Process is found in one of the following states in the TCQ:

- EXECUTION indicates that the Process is executing.
- WAIT indicates that the Process is waiting until a connection with the target node (SNODE) is established or available based on Connect:Direct server configuration. Processes may also be waiting for their turn to execute on an existing session.
- HOLD indicates that the Process was submitted with a Process HOLD or RETAIN parameter. Process execution is held on the queue until released by an operator or the SNODE connects with a request for held work. The HOLD queue state also applies to Processes that stop executing when an error (HOLDERROR) occurs. If a Process is submitted with a RETAIN parameter, a copy of the Process will remain in the hold queue after successful execution.
- TIMER indicates that the Process was submitted with a Process STARTT parameter that designates the time, date, or both that the Process should begin execution. Processes that initially failed due to failure to connect with the SNODE or a file allocation failure may also be found in this queue state waiting for their retry interval to expire. Processes will retry automatically.

A queued Process can be queried and manipulated through Connect:Direct commands such as SELECT, CHANGE, DELETE, FLUSH, and SUSPEND PROCESS. For complete information on the Connect:Direct commands and the various queues, refer to the Connect:Direct user’s guide for your platform.

A message indicating that the Process completed successfully is returned when the Process is placed in the TCQ. The Process statements have been checked for syntax, but the Process may not have been selected for execution.

Step 4-Connect:Direct Finds an Available Connection and Process Execution Begins

The Process is selected for execution based on Process parameters and the availability of the remote node.

Chapter 2

Process Statement Structure and Syntax

This chapter describes the components of a Connect:Direct Process and the syntax used in the Process language. An example Process is included.

Components of a Process

A Connect:Direct statement consists of the following components:

✦ Label

✦ Statement Identifier

✦ Parameters or Subparameters

A discussion of each element follows.

Label

Connect:Direct statements are identified by user-defined labels. A label is any character or character string beginning in column one. The label consists of a 1-8 character alphanumeric string, with the first character alphabetic only. The PROCESS statement is the only statement that requires a label.

Statement Identifier

The statement identifier specifies the function requested. Statements must begin *after* column one. A statement identifier beginning in column one is considered a label by Connect:Direct. Statement identifiers are separated from parameters by one or more blanks or commas.

Parameters or Subparameters

Parameters or subparameters specify further instructions for the statement and must be set apart by one or more blanks or commas. Parameters can be either keyword or positional. Multiple symbolic substitutions must be separated by one or more spaces.

Keyword Parameters

Keyword parameters are usually followed by an equal sign and may have a set of subparameters.

Positional Parameters

Positional parameters must be entered in a specific order, with commas replacing any parameter omitted. These parameters are always on the right of the equal sign. Positional parameters must be enclosed in parentheses, with the parentheses optionally preceded and followed by blanks or commas.

Subparameters

A positional parameter or the variable information in a keyword parameter is sometimes a list of subparameters. The list may contain both positional and keyword parameters. Positional subparameters must be enclosed in parentheses, with the parentheses optionally preceded and followed by blanks or commas.

Connect:Direct Syntax

This section describes the syntax used in constructing Connect:Direct Process statements.

Special Characters

Those symbols defined to be special characters in the Connect:Direct product are hyphens (-), two vertical bars (||), ampersands (&), and the following Connect:Direct delimiters or operators:

	(blank)	¬	(not sign)	’	(single quotation mark)
<	(less-than sign)	/	(slash)	"	(double quotation mark)
>	(greater-than sign)	\	(backslash)	[ ]	(brackets)
( )	(parentheses)	,	(comma)	{ }	(braces)
=	(equal sign)	.	(period)	*	(asterisk)

Note: The EBCDIC Hex value for the slash (/) is X’EO. The EBCDIC Hex value for the vertical bar (|) is X’4F.

Commas

Commas have two functions:

- Separate items within a list.
- Control the order of values specified as positional parameters.

A comma must be used to indicate omission of a positional parameter.

```
SIGNON USERID=(id,,newpswd)
```

Do not use commas to separate multiple symbolics in a Process. Separate multiple symbolics with one or more spaces.

Continuation Marks

Use the hyphen as a continuation mark to indicate the statement continues on multiple lines. The hyphen must be separated from the preceding characters by at least one blank.

For Connect:Direct Tandem: Both hyphen (-) and ampersand (&) are supported as continuation characters.

For Connect:Direct UNIX, and OpenVME: Continuation characters are not required.

Parentheses

Parentheses enclose lists and associate a group of values. For example, the FROM clause of the COPY statement is enclosed in one set of parentheses. Lists in the FROM clause are nested in subsequent pairs of parentheses.

Asterisks

Asterisks are used to indicate generic specifications of parameters. With generics, you request information by specifying a single asterisk (*) or a character string plus an asterisk.

For example, the following FROM clause of a COPY statement selects generically all member names beginning with ACCT (the first four characters of the data set names) from the PDS named PDS.SOURCE.

```
COPY FROM (DSN=PDS.SOURCE SELECT=(ACCT*))
```

For Processes for Connect:Direct OS/400: Asterisks precede some subparameters and must be typed when they are in the statement format shown in this manual. This is an OS/400 convention.

Comments

Comments allow you to include additional information within a Connect:Direct Process. Comments are allowed in the following formats:

- ✦ An asterisk (*) in column one, followed by the comment.
- ✦ Preceded by a slash-asterisk (/*) and followed by an asterisk-slash (*/).

Note: This format can be used after a continuation mark as well as at the beginning of a line.

- ✦ Preceded by a slash-asterisk (/*), continuing over multiple lines, and terminated by an asterisk-slash (*/).

  Note: The terminating */ cannot begin in column one.

  Note: The only permitted form of commenting for statements that will be processed by DMBATCH is an asterisk (*) in column one, followed by the comment.

The following example illustrates all of the ways you can use comments.

```
/* This type of comment can be written on one line*/
/*
It can also continue across multiple lines.  Remember that
the terminating asterisk-slash cannot begin in column one.
 */
COPY      FROM (                                 -             /* INPUT */
               DSN=&DSN1                         - /* SYMBOLIC DATA SET */
  UNIT=SYSDA)
* After submitting this Process,
* enable the Connect:Direct UNIX node.
```

Concatenation for the Connect:Direct OS/390, VM/ESA, VSE/ESA, and OS/400

Concatenation joins character strings to form a single string. The operator blank-vertical bar-vertical bar-blank (||) indicates concatenation. For example:

```
DSN=CD || NODE
```

Resolves to:

```
DSN=CDNODE
```

Concatenation for Connect:Direct Tandem

The Connect:Direct Tandem supports two concatenation operators - vertical bars (||) or the ampersand (&) for the PACCT and SACCT parameters. Typically, concatenation is not necessary for any other parameters. PACCT is used only for documentation for Connect:Direct Tandem.

The following example PROCESS statements illustrate the differences between the two types of concatenation for the SACCT parameter. In both examples, ampersands are used to indicate continuation of the PROCESS statement.

Example of Ampersand Concatenation in Connect:Direct Tandem

The following example shows an ampersand (&) used as a concatenation character. When used for concatenation, the ampersand (&) must be in column 80 and the remainder of the string must begin in column one of the next line to ensure that blanks are not added to the string. The entire string must be enclosed in single quotation marks.

```
PROC1  PROCESS     SNODE=CD.OS390.NODE                                      &
                   SACCT=‘123456789012345678901234567890123456789012345678 9 &
012345678901234567890’                                                      &
                     SNODEID=(USERID,PASWRD)
```

Example of Vertical Bar Concatenation in Connect:Direct Tandem

The following example shows two vertical bars (||) used as a concatenation character. Do not use blanks before or after the two vertical bars (||) so blanks are not added to the string. Each line of the SACCT string must begin and end with single quotation marks.

```
PROC1   PROCESS   SNODE=CD.OS390.NODE                     &
                  SACCT=‘12345678901234567890’||&
                         ‘1234567890123456789’            &
                  SNODEID=(USERID,PASWRD)
```

Concatenation for Connect:Direct OpenVMS

Either double quotation marks (“ ”) or single quotation marks (‘ ’) can be used in conjunction with the continuation character to allow a string spanning multiple records to be concatenated without extraneous blanks. The second and subsequent records must begin in column one. For example:

```
CONCAT   PROCESS      SNODE=CD.VMS.NODE
                      SYMBOL &TO="$DISK1:"               -
"<DIRECTORY>TEST.DAT"
STEP01   COPY     FROM(DSN=IBMFILE SNODE)                -
                  TO  (DSN=&TO PNODE DISP=NEW)
```

Concatenation for Connect:Direct UNIX and Connect:Direct OpenVME

Use stream input instead of concatenation or continuation symbols as required by other Connect:Direct operating environments. Also, grammar is based on the sequence of parameters and arguments instead of position within the inline buffer. The exception is that comment identifiers (asterisks [*] and pound [#] signs) in column one are positional.

Note: Concatenation is also used in conjunction with Special Purpose Bracketing and in Symbolic Substitution to join values that are represented as symbolic parameters. See the following sections for more information.

```
&USERID=BOB
DSN=CD || &USERID
```

Resolves to:

```
DSN=CDBOB
```

Note: Resolution of the symbolic occurs before concatenation.

Special Purpose Bracketing

Special characters must often be maintained as part of a string. For this to occur, the string must be enclosed within bracketing characters. Those symbols defined as bracketing characters are backslashes (\\), single quotation marks (‘ ’), and double quotation marks (“ ”).

Bracketing Backslashes

Bracketing backslashes are indicators of special processing of a character string and are not maintained as part of the string at its final resolution.

For Connect:Direct OS/390, VM/ESA, and VSE/ESA: Use backslashes for bracketing.

For Processes submitted from Connect:Direct Tandem, UNIX, OpenVME, and OpenVMS: Bracketing backslashes are not valid.

For Connect:Direct Tandem and OpenVMS: Refer to the following section, *Single and Double Quotation Marks*, for details.

Bracketing backslashes should be used to:

- ✦ Continue a string containing special characters across multiple lines.
- ✦ Ensure that quotation marks in the string are maintained.

Both backslashes must be in the same line. If a string containing special characters continues across multiple lines, each line containing a special character must be enclosed in backslashes and must be concatenated.

For example, the following SYSOPTS parameter for Connect:Direct OS/400 is a quoted string and must be enclosed in backslashes when it continues across multiple lines:

```
SYSOPTS=     \"CMD(\        ||                               -
             \SNDBRKMSG\    ||                               -
             \)"\
```

Resolves to:

```
 SYSOPTS=“CMD(SNDBRKMSG)”
```

If the bracketing character being used to enclose the string is also part of the character string, you must place an additional bracketing character where the single bracketing character is needed. For example:

```
PACCT=   \‘DEPT\\MIS\       ||                                          -
         \602’\
```

Resolves to:

```
PACCT=‘DEPT\MIS602’
```

Single and Double Quotation Marks

Single and double quotation marks are used to allow special characters or blanks to be embedded within a parameter or subparameter value. For example:

```
COPY TO (DSN=‘VMFILE FILETYPE’)
```

```
COPY TO (DSN=\"C:\\PCDIR\\BAT.EXE"\)
```

For Connect:Direct OS/400 and OpenVMS: Syntax conventions require the entire SYSOPTS string to be enclosed in double quotation marks (“ ”).

For Connect:Direct OS/390: The DMRTSUB utility requires parameters to be enclosed in double quotation marks (“ ”).

The rules for using single and double quotation marks are as follows:

- Single-quoted strings allow the parsing of parameters as entered.
- Double-quoted strings allow the resolution of &values in a quoted string.

For Connect:Direct UNIX, and OpenVME: The software supports the use of double quotation marks. Any other exceptions are noted in the text.

SUBMIT Statement Parsing

Parsing of special bracketing and single and double quotes is performed differently in a SUBMIT command than in a SUBMIT statement within a Process.

For example, a SUBMIT command executed from DMBATCH, resolves:

```
SYMBOL  &BATCHID2=\'''\ || BATCHID || \'''\
```

to:

```
'''BATCHID'''
```

A SUBMIT statement within a Process resolves the same string to:

```
'BATCHID'
```

This is an important to remember when a SUBMIT is performed between different platforms and products, such as from Connect:Direct UNIX to Connect:Direct OS/390 to Connect:Enterprise.

Symbolic Substitution

Symbolic substitution is used to substitute information in a Process. When Connect:Direct encounters an ampersand (&) followed by 1-8 alphanumeric characters, it will substitute a string represented by the ampersand and alphanumeric characters. For example:

```
&USERID=BOB
DSN=CD || &USERID
```

Resolves to:

```
DSN=CDBOB
```

Separate multiple symbolics with spaces, as shown in the following:

```
SUBMIT PROC=TSTSEND &DSN1=TSTSEND.VAR0001.S200010 &RUNDATE=200012 &TSTDATE=200010
```

Symbolic resolution occurs before concatenation.

The following is an example of enclosing a string in double quotation marks to allow resolution of the symbolic &FILTYP.

```
PROC2  PROCESS      SNODE=CD.VM &FILTYP=FT
       COPY  FROM   (DSN=OS390.DATA                                -
                    DISP=SHR)                                      -
             TO     (DSN="FN || &FILTYP"                           -
                    LINK=(IVVB,WIVVB,W,191)                        -
                    DISP=(RPL))
```

Note: Double quotation marks are not valid for symbolic substitution in Windows. The SYMBOL statement and concatenation must be used.

Termination

A statement is terminated by the end of data without a continuation mark.

Example

The following example Process uses Connect:Direct OS/390 Process statements to illustrate product functionality.

```
/*
   COPY01  PROCESS  --  copies PDS members beginning with INV*
                           to the Chicago node
                        if successful, runs USERJOB on the
                           Chicago node to modify data
                        copies modified FILEA back from Chicago
                        if not successful, sends NOTIFY to inform about
                           STEP01 failure

   Modification History:

   04.Jul.1998  CSG  Initial Implementation
 */

COPY01  PROCESS      SNODE=CD.CHICAGO
STEP01  COPY    FROM (DSN=MASTER.PDS          -  /* send PDS member INVxxxx */
                      SELECT=INV*)            -  /* to Chicago and place in */
               TO   (DSN=MUNGE.ME)               /* file “MUNGE.ME” (only 1 */
                                                /* member matches criteria) */
        IF           (STEP01 < 8) THEN             /* if copy successful */

STEP02  RUN JOB      (DSN="xxNT") SNODE        -  /* run “data modify” job */
                    SYSOPTS=\"PGM(fix.bat)\ ||                            -
                             \ARGS (PO_num)"\

STEP03  COPY    FROM (DSN=MUNGE.ME SNODE)      -  /* and bring file back */
                TO   (DSN=MODIFIED || %SUBDATE)

        ELSE
                                        /* or if step 1 failed */
STEP04  RUN TASK     (PGM=DMNOTIFY, PARM=(’FAIL’, FILEA || %SUBDATE)) PNODE
        EIF
```

A description of the Process follows:

COPY01 is a PROCESS statement that identifies the secondary node (SNODE) as CD.CHICAGO. The SNODE is the Connect:Direct node that interacts with the primary node (PNODE) during Process execution. The SNODE can also be referred to as the participating, target, or remote node. A remote node must be specified in every Process. The PNODE is the Connect:Direct node on which the Process is submitted. The PNODE may also be referred to as the local node. PNODE is used for documentation only.

STEP01 is a COPY statement that copies a file on the PNODE to a file on the SNODE.

STEP02 and STEP03 use Conditional statements. If STEP01 completes successfully (that is, the return code [RC] is less than **8**), the THEN path is taken and STEP02 and STEP03 execute. If STEP01 fails, the ELSE path is taken and STEP04 executes.

STEP04 is a RUN TASK statement. The DMNOTIFY program executes within the Connect:Direct environment.

See the following chapters for examples on various operating systems.

Chapter 3

PROCESS Statement Examples

This chapter contains examples of Process Statements. Each Process statement example contains a different set of parameters.

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 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 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 Connect:Direct UNIX, VSE/ESA, OpenVMS, and Tandem nodes: NOTIFY is ignored.

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

Using %PNODE

This Process shows using a variable to substitute to %PNODE. %PNODE is set to the node name from which the Process is submitted.

Note: %PNODE is only valid for Connect:Direct OS/390 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
```

Using a TCP/IP Dotted-Form Address for the SNODE Keyword

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

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

Using 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
```

Specifying Accounting Data (Connect:Direct UNIX and Connect:Direct OpenVME)

This Process shows how to specify PNODE and SNODE accounting data in the PROCESS statement. The string must be enclosed in double quotation marks.

```
proc2      process      snode=unix.node
                        pacct="abc"
                        sacct="123"
step01     copy  from   (
                        file=file1
                        snode
                        )
                 to     (
                        file=file2
                        pnode
                        )
           pend
```

Specifying a Process Starting Day and Time (Connect:Direct UNIX and Connect:Direct OpenVME)

This Process shows how to specify the day and time for execution of a Process. In this example, the Process will execute today at 3:31 p.m. The day and time must be separated by a comma, and the string enclosed in parentheses.

```
oc2        process      snode=unix.node
                        startt=(today, 03:31:00 pm)
step01     copy  from   (
                        file=file1
                        snode
                        )
                 to     (
                        file=file1
                        pnode
                        )
           pend
```

Chapter 4

COPY Statement Examples

This chapter contains examples of Connect:Direct COPY statements for various Connect:Direct platforms. Cross platform COPY statements are emphasized.

Examples in this chapter are divided into the following sections:

- File allocation (OS/390 to OS/390)
- Copying PDS to PDS (OS/390 to OS/390)
- Copying to tape (OS/390 to OS/390)
- Copying files using exits (OS/390 to OS/390)
- Copying files using SMS parameters
- Copying to OS/390 nodes with unique member name allocation (AXUNIQ Exit)
- Copying between OS/390 and other Connect:Direct nodes
- Copying between Tandem and other Connect:Direct nodes
- Copying between UNIX and other Connect:Direct nodes
- Copying between OpenVME and other Connect:Direct nodes
- Copying between VM and other Connect:Direct nodes
- Copying between OpenVMS and other Connect:Direct nodes
- Copying between Windows and other Connect:Direct nodes

File Allocation (OS/390 to OS/390)

Using Defaults to Allocate a File

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=$DAL.PSDATA)                           -
               CKPT=128K                                              -
               COMPRESS-
               TO    (SNODE DSN=$CHI.PSDATA                           -
                    DISP=(NEW,CATLG)                                  -
                    UNIT=3380                                         -
                    VOL=(SER=SYS009)                                  -
                    SPACE=(4096,(200,40),,,ROUND))
```

Using %SUBDATE and %SUBTIME for a File Name

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 Connect:Direct OS/390.

```
CD1      PROCESS    SNODE=CD.VM.NODE                                  -
                     &DATE=%SUBDATE                                   -
                     &TIME=%SUBTIME

STEP01   COPY  FROM (                                                 -
                    DSN=ACCTNG.DAILY.DATA                             -
                    SELECT=(PL*)                                      -
                    )                                                 -
               TO   (                                                 -
                    DSN=ACCTNG.UPDATES.D || %DATE || .T || %TIME      -
                    DISP=(NEW,CATLG)                                  -
                    DCB=(DSORG=PO,LRECL=80,RECFM=FB,BLKSIZE=8000)     -
                    )
```

File Allocation Using a TYPE File

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=128K                                              -
               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)
```

Note: 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. Refer to the *Connect:Direct User's Guide* for the appropriate platform for details on setting up entries in the TYPE file.

Copying a SAM File

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=128K                                                   -
                COMPRESS                                                    -
                TO    (DSN=CHI.PSDATA)
```

Copying a DFDSS Volume Dump

This Process transmits an OS/390 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)
```

Copying PDS to PDS (OS/390 to OS/390)

Copying an Entire PDS

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                                                   -
         COMPRESS                                                  -
         TO    (DSN=PDS.DEST)
```

Specifying a Range and the NR Subparameter to Copy Selected PDS Members

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)
```

Copying One Member of a PDS

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)
```

Copying a PDS and Excluding an Individual Member

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)
```

Copying a PDS and Excluding Members Generically

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)
```

Copying a PDS and Using a Range to Exclude PDS Members

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.

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

Copying a PDS and Generically Selecting Members

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)
```

Copying PDS Members Using the EXCLUDE and SELECT Parameters

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 from the COPY 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.

Refer to the *Connect:Direct Process Guide for Mainframe Products* volume for the precedence for the SELECT and EXCLUDE parameters.

Copying a PDS Using the ALIAS Parameter with SELECT and EXCLUDE

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.

Copying to Tape (OS/390 to OS/390)

Copying a PDS Member to Tape

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 Connect:Direct OS/390 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))
```

Copying Files Using Exits (OS/390 to OS/390)

Using the IOEXIT Parameter

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 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 Connect:Direct to OUEXT03, an IOEXIT on the destination Connect:Direct node.

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

Copying Files Using SMS Parameters

Copying to a New SMS-Controlled Data Set

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)
```

Copying to a New SMS Data Set Using LIKE

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)
```

Creating and Copying a PDSE Data Set

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)
```

Creating and Copying a VSAM KSDS Data Set

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)
```

Creating and Copying a VSAM ESDS Data Set

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)
```

Creating and Copying a VSAM RRDS Data Set

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)
```

Creating and Copying a VSAM Linear Data Set

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)
```

Copying a Data Set with a Security Profile

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)
```

Copying a File from Connect:Direct Tandem to Connect:Direct OS/390

This Process copies a file from a Connect:Direct Tandem node to a new SMS-controlled file at a Connect:Direct OS/390 node.

```
SMS1     PROCESS     PNODE=CD.TANDEM                                       -
                     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")
```

Copying to OS/390 Nodes with Unique Member Name Allocation (AXUNIQ Exit)

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

Resolution Method

The member name is made to be 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, Connect:Direct will modify the specified member name to create a unique name.
- ✦ If the *seed* name is less than eight characters long, 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, 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 OS/390 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")
```

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 OS/390 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")
```

Connect:Direct resolves the member name to DATAFIL1.

Copying Between OS/390 and OS/400 Nodes

Copying a Sequential File from OS/390 to a Member of a Physical Data Base File

This Process copies a sequential file from OS/390 to a member of a physical data base file on the OS/400 node. The RUN TASK then sends a message notifying an OS/400 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. Connect:Direct syntax requires backslashes to continue the SYSOPTS over multiple lines when the Process is submitted from an OS/390 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)\                           -
                            \   )\                                    -
                            \"\
```

Copying a Member of a Physical Data Base File from OS/400 to a Sequential File on OS/390

This Process copies a member of a physical data base file from the OS/400 node to a sequential file on OS/390. DCB information is specified for file allocation on OS/390. 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)
```

Copying a Data Set from OS/390 to a Spooled File on OS/400

This example copies a data set from OS/390 to a spooled file on an OS/400 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 OS/390 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)
```

Copying a Member of a PDS from OS/390 to a Spooled File on OS/400

This Process copies a member of a PDS from OS/390 to a spooled file on an OS/400 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)\                              -
                             \"\)
```

Copying Between OS/390 and Tandem Nodes

Submitting a Process from a Connect:Direct Tandem Node that Copies a File from OS/390 to Tandem

This Process is submitted from a Connect:Direct Tandem node, the PNODE, and copies the sequential file JSMITH.EFILE from a Connect:Direct OS/390 node, the SNODE, to $B.BILL.EFILE. EFILE is an entry-sequenced file. EBCDIC-to-ASCII translation is enabled with the SYSOPTS parameter SET XLATE ON. This parameter must be positioned in the Process wherever the Tandem file (DSN) is specified.

```
EPROC     PROCESS    SNODE=CD.OS390.JSMITH
STEP01    COPY  FROM (SNODE DSN=JSMITH.EFILE                              -
                     DCB=(DSORG=PS,RECFM=FB)                              -
                     DISP=SHR)                                            -
                TO   (PNODE DSN=$B.BILL.EFILE                             -
                     DCB=(DSORG=E)                                        -
                     SYSOPTS=\"’SET XLATE ON’"  \                         -
                     DISP=RPL)
```

Copying to an Entry-Sequenced File (OS/390 to Tandem)

In this multi-step Process, STEP1 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. STEP2 will copy DATA1.SEQ.A110257 from the OS/390 node (SNODE) to the entry-sequenced file, $B.FILERESO.A110257, at the PNODE.

SYSOPTS subparameters are used to define the attributes of the new receiving file on the Tandem system.

```
A110257   PROCESS    PNODE=CD.TANDEM                                        -
                     SNODE=SC.CD.OS390
STEP1     RUN TASK   (PGM=FUP                                               -
                     PARM=('/OUT $S.#FUPTEST/'                              -
                           'PURGE A110257  '))                              -
                     PNODE
STEP2     COPY  FROM (DSN=DATA1.SEQ.A110257                                 -
                     SNODE                                                  -
                     DISP=SHR)                                              -
                TO   (DSN=$B.FILERESO.A110257                               -
                     PNODE                                                  -
                     DISP=NEW                                               -
                     SYSOPTS=\"'SET TYPE E'       \||                       -
                             \'SET BLOCK 4096'    \||                       -
                             \'SET MAXEXTENTS 16' \||                       -
                              \'SET XLATE ON'"     \)
```

Creating an Unstructured Code Type 101 File (OS/390 to Tandem)

This Process is submitted at the OS/390 node and will copy a file from OS/390 and create a Tandem unstructured code type 101 file ($B.FILERESO.UNST1).

```
CHECK1    PROCESS    PNODE=SC.OS390.NODE                                    -
                     SNODE=CD.TANDEM
STEP1     COPY  FROM (DSN=DATA1.VSAME001.DATA                               -
                     PNODE)                                                 -
                TO   (DSN=$B.FILERESO.UNST1 DISP=(NEW)                      -
                     SYSOPTS=\"'SET TYPE U'       \||                       -
                             \'SET CODE 101'      \||                       -
                             \'SET BLOCK 2048'    \||                       -
                             \'SET EXT (2,2)'"    \)
```

Creating an Unstructured Code Type 0 File (OS/390 to Tandem)

This Process is submitted at the OS/390 node and will copy a file from OS/390 and create a Tandem unstructured, code type **0** file ($B.FILERESO.UNST1).

```
CHECK1    PROCESS    PNODE=SC.OS390.NODE                                    -
                     SNODE=CD.TANDEM
STEP1     COPY  FROM (DSN=DATA1.VSAME001.DATA                               -
                     PNODE)                                                 -
                TO   (DSN=$B.FILERESO.UNST1 DISP=(NEW)                      -
                     SYSOPTS=\"'SET TYPE U'    \||                          -
                             \'SET CODE 0'     \||                          -
                             \'SET BLOCK 4096' \||                          -
                             \'SET EXT (2,2)'" \)
```

Copying an Unstructured Code Type 0 File from Tandem to OS/390

This Process will copy an unstructured file code type **0**, reading in records of 100 bytes, from the Tandem node to the OS/390 node. In this example, file attributes are specified using the DCB parameter. Because the file is unstructured Code **0**, include LRECL on the FROM clause of the COPY statement as instruction to the Connect:Direct Tandem system on reading logical records.

```
CHECK1   PROCESS    PNODE=CD.TANDEM                                        -
                    SNODE=SC.OS390.NODE
STEP1    COPY  FROM (DSN=$B.FILERESO.UNST1 PNODE                           -
                    DCB=(LRECL=100))                                       -
               TO   (DSN=DATA1.VSAME001.DATA SNODE                         -
                    DCB=(RECFM=FB                                          -
                        ,LRECL=100                                         -
                        ,DSORG=PS                                          -
                        ,BLKSIZE=4000)                                     -
                    DISP=NEW)
```

Copying a Sequential File from an OS/390 Node to a Tandem Node

The following Process transfers a sequential file from OS/390 to Tandem. The Process is submitted on the OS/390 node. Because the parameter SET EXT (50 50) is specified, the Tandem 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.

```
TANDEM   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)
```

Copying a File Submitted from OS/390 to Tandem

The following Process, submitted on a Connect:Direct OS/390 node, copies a data set from OS/390 to a remote node on a Tandem EXPAND network. The SNODEID field passes the userid 147.200 to the Tandem, along with the password MONEY.

The userid and password are validated before the Connect:Direct OS/390 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 OS/390 to Tandem.

```
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'    \||                      -
                             \'SET XLATE ON'"  \)
```

Note: Because the source file includes the special backslash character, single quotation marks must enclose the data set name.

Copying a File from Tandem on an EXPAND Network to OS/390

This Process is submitted on the OS/390 node to copy a file from a Tandem node on an EXPAND network to an OS/390 node. The Connect:Direct Tandem data transformation facility (DTF) (or server) is on system \SYSTEM, but the file is being copied from system \SYSEXT.

Note: Because the Process was submitted from a Connect:Direct OS/390 node, single quotation marks must enclose the data set name.

```
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)
```

Copying a File from Tandem to OS/390 After Running DMRTDYN on OS/390

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 Tandem node to DATA1.SEQ.VSAMKSDS at the Tandem node.

```
VSAMKS   PROCESS    PNODE=CD.TANDEM                                   -
                    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))
```

Using FUP in a Process Submitted on OS/390 to Delete a File on Tandem

In this Process, the FUP utility is used to delete a Tandem file. The Process is submitted on OS/390.

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

Using Connect:Direct to Allocate a Partitioned File on a Single System (OS/390 to Tandem)

In this Process, the Connect:Direct system is used to allocate a Tandem partitioned file. All partitions reside on the same system. The Process is submitted on OS/390.

```
PART0001  PROCESS    SNODE=TANDEM.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 (OS/390 to Tandem)

This Process will copy a file from the OS/390 node to the Tandem node. Because the Process is submitted from the OS/390 node, syntax conventions must follow those established for OS/390. 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.TANDEM                                     -
                      SNODEID=(157.214 ABLE1)
STEP01    COPY  FROM  (DSN=DATA1.SMALLER                                    -
                      UNIT=SYSDA)                                           -
                TO    (DSN=$C.ACCTPROC.TANDEM                               -
                      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.TANDEM                                     -
                      SNODEID=(157.214 ABLE1)
STEP01    COPY  FROM  (DSN=DATA1.SMALLER                                    -
                      UNIT=SYSDA)                                           -
                TO    (DSN=$C.ACCTPROC.TANDEM                               -
                      SYSOPTS=\"’SET EXT(10,10)\ ||                         -
                              \  SET XLATE ON\ ||                           -
                              \  SET TYPE E\ ||                             -
                              \  SET REC 100’"\                             -
                      DISP=RPL)
```

Copying a File from a Tandem Spooler to an OS/390 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 Tandem 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.

```
EXTOS390F PROCESS    PNODE=CD.TANDEM                                         -
                     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)
```

Copying Between an OS/390 Node and a Remote Tandem Spooler on an EXPAND Network

This multi-step Process copies between an OS/390 node and a remote Tandem spooler on an EXPAND network. The Process is submitted from the Tandem node. STEP01 copies a file from a DSN on the OS/390 node to the Tandem spooler. STEP02 copies a file from the Tandem spooler to an OS/390 node.

Note that \SYSEXT is an EXPAND node other than the one that the Connect:Direct Tandem server resides on. Because the Tandem 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)
```

Copying Files Between the Tandem Spooler System and an OS/390 Node Using Job Number

This Process transfers a file from the Tandem spooler $S to an OS/390 sequential file. The Tandem 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)
```

Copying a Disk File from Tandem to a Tape Device at OS/390

This Process is submitted on the Tandem node to copy a disk file on Tandem to a tape device at the OS/390 node. Because NL (no labels) is specified, DCB attributes must be specified to identify the file on the volume**.**

```
PROCESS1  PROCESS    PNODE=CD.TANDEM                                       -
                     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)
```

Copying a Tape File from OS/390 to a Disk File on Tandem

This Process is submitted at the Tandem node to copy a tape file from the OS/390 node to a disk file on the Tandem 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 Tandem.

```
PROCESS1  PROCESS    PNODE=CD.TANDEM                                  -
                     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)"))
```

Copying a File from OS/390 to Tandem Using the FASTLOAD Option

This Process is submitted at the Tandem node to copy an entry-sequenced file from OS/390 to a key-sequenced file at the Tandem 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.TANDEM                                  -
                     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)
```

Allocating a VSAM Data Set and Copying a File from Tandem to OS/390

This Process invokes 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 a Tandem 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)
```

Copying Between OS/390 and UNIX Nodes

Copying Files Between UNIX and OS/390

This Process copies a file from UNIX to OS/390. 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 OS/390 node, copies a text file from UNIX to OS/390. 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)
```

Copying Files Between OS/390 and UNIX

This Process copies a sequential file from an OS/390 node to a UNIX node (STEP1) and back to the OS/390 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 an OS/390 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=50K                                                            -
       COMPRESS EXTENDED                                                   -
       TO  (SNODE DSN=’benchm1’ DISP=MOD                                   -
            SYSOPTS=”:datatype=text:strip.blanks=no:”)
```

Copying Between OS/390 and OpenVME Nodes

Copying Files Between OpenVME and OS/390

This Process copies a file from OpenVMS to OS/390. The Process was initiated from the OpenVME 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=newfile
                      pnode)
                ckpt=no
                to    (file=newfile2
                      dcb=(dsorg=PS,
                            recfm=FB,
                            lrecl=80,
                            blksize=4800)
                      space=(TRK,(1,,))
                      snode
                      disp=rpl)
          pend
```

Copying a File from OpenVME to OS/390

This Process, submitted from the OS/390 node, copies a text file from OpenVME to OS/390. The DSN and SYSOPTS strings for the OpenVME system must be in the proper case for OpenVME and enclosed in double quotation marks.

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

Copying Between OS/390 and VM Nodes

Copying a File from OS/390 to VM

This Process copies an OS/390 data set to a file on the 191 disk of VM user IVVB. If the file exists, the Connect:Direct system replaces it. If the file does not exist, the Connect:Direct system creates it as indicated by the DISP=RPL parameter.

The following Process illustrates copying a file from OS/390 to VM.

```
OS390TOVM1  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))
```

Copying an OS/390 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 OS/390. 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))
```

Copying an OS/390 File to Tape on VM

This Process copies a disk file from a Connect:Direct OS/390 node to tape at the Connect:Direct VM node. If the file exists, the Connect:Direct system replaces it. If the file does not exist, the 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)
```

Copying an OS/390 File to Spool on VM

This Process copies a disk file from a Connect:Direct OS/390 node to spool at a Connect:Direct 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))
```

Copying Between OS/390 and OpenVMS Nodes

Copying PDS Members from OS/390 to OpenVMS

This Process copies a PDS and all of its members from OS/390 to a text library on the VAX. 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)
```

Using the SYSOPTS Parameter (OS/390 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. Connect:Direct syntax requires using backslashes to continue the SYSOPTS over multiple lines when the Process is submitted from an OS/390 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' "\)
```

Copying a 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,PASSWORD)
STEP01   COPY  FROM  (DSN=DUXX:<DIRECTORY>TESTFILE.DAT)                  -
               TO    (DSN=MUXX:TESTFILE.DAT                               -
                     SYSOPTS="MOUNT='MUXX: TAPE /OVERRIDE=ID' ")
```

Copying a 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,PASSWORD)
STEP01   COPY  FROM (DSN=MUXX:TESTFILE.DAT                                -
                    SYSOPTS="MOUNT='MUXX: TAPE /OVERRIDE=ID' ")           -
               TO   (DSN=DUXX:<DIRECTORY>TESTFILE.DAT)
```

Copying from OS/390 to OpenVMS and Specifying a User-Defined Translation Table

This Process, submitted from the OpenVMS node, copies from a Connect:Direct OS/390 node to a Connect:Direct 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 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)
```

Copying a Single Entry from the OpenVMS Text Library to an OS/390 Member

This Process sends a single entry from an OpenVMS text library to a member of a PDS on OS/390.

```
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)
```

Copying All Entries from an OpenVMS Text Library to OS/390

This Process copies all the modules of a text library to respective members of a PDS on OS/390. 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)
```

Copying a Data Set from an OS/390 Node to an Executable File on an OpenVMS Node

This Process, submitted from the OpenVMS node, copies the OS/390 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)
```

Copying an Executable File from an OpenVMS Node to an OS/390 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 OS/390 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 OS/390 node (the PNODE is an OS/390 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)
```

Copying Between OS/390 and VSE Nodes

Copying a File from OS/390 to VSE (DYNAM/T Tape Files)

This multi-step Process copies various sequential files from an OS/390 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)
```

Copying an OS/390 PDS Member to a New VSE File in a DYNAM Pool

This Process copies a PDS member on OS/390 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)
```

Copying an OS/390 BSAM File to a VSE-Controlled Disk Data Set

Use this Process to transfer a OS/390/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.

```
OS3902DYD1 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
```

Copying an OS/390 Sequential Data Set or PDS to a VSE-Controlled Tape Data Set

This Process copies either an OS/390 sequential data set or an OS/390 partitioned data set into a CA-DYNAM/T or CA-EPIC noncontrolled tape data set. The input resides on OS/390 and the output file is written to tape (or cartridge) on VSE.

```
OS3902DYT1 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)               -
                   )
```

Copying an OS/390 PDS Member to a VSE BSAM Sublibrary Member

This Process copies an OS/390 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 OS/390 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 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.

```
OS3902LIB1 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
```

Copying an OS/390 PDS Member to a VSE VSAM Sublibrary Member

This Process copies an OS/390 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 OS/390 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.

```
OS3902LIB2 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)                                    -
                   )
```

Copying an OS/390 Sequential Data Set or OS/390 PDS to a VSE-Controlled Tape Data Set

This Process copies either an OS/390 sequential data set or an OS/390 partitioned data set into a CA-DYNAM/T or CA-EPIC controlled tape data set. The input is from OS/390 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.

```
OS3902TAP1 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
```

Copying Between OS/390 and Windows Nodes

Copying a File from OS/390 to Windows

This Process copies a file from an OS/390 node to a Windows node. The DSN (data set name) for the Windows file uses the Universal Naming Convention (UNC) to specify the computer name \\NT-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 Connect:Direct server is not running. 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=NT.1200.SNA                                     -
                    HOLD=NO                                               -
                    RETAIN=NO
STEP1    COPY  FROM (DSN=TEST.DEL.DATA99)                                 -
               TO   (DSN='\\NT-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=NT.1200.SNA                                     -
                    HOLD=NO                                               -
                    RETAIN=YES                                            -
                    STARTT=(MONDAY,03:00)                                 -
                    &DSN1='TEST.DEL.TEST99'                               -
                    &DSN2='\\NT-NODE\ROOT_D\USERS\TEST1\DATA1.TXT'
STEP1    COPY  FROM (DSN=&DSN1)                                           -
               TO   (DSN=&DSN2                                            -
                    SYSOPTS="DATATYPE(TEXT)"                              -
                    DISP=(RPL))
```

Copying Between Tandem Nodes

Copying to an Entry-Sequenced File (Tandem to Tandem)

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")
```

Copying Files Between Tandem Spooler Systems

This Process selects a spooler job by job number and copies it to another Tandem spooler system at a remote node. The Tandem 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)
```

Copying Between Tandem and OS/400 Nodes

Copying a Tandem File to an OS/400 Node

This Process copies a Tandem file to a member of a physical data base file on OS/400. The OS/400 file is created with maximum members specified as 100. The Tandem 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)"))
```

Copying Between Tandem and UNIX Nodes

Copying Text Files from Tandem to UNIX

This Process, submitted from the Connect:Direct Tandem node, will copy datafile in $B.smith to /payroll/monthly/jan on the Connect:Direct UNIX node. For Connect:Direct UNIX nodes, the security userids 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
```

Copying Binary Files from Tandem to UNIX

This Process, submitted from the Connect:Direct Tandem node, will copy a file in $user.unixdata.cdcom to a binary file on the Connect:Direct 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)
```

Copying Between Tandem and VM Nodes

Copying Files from Tandem to VM

This Process illustrates the transmission of a file from a Tandem node to a VM node. The Process is submitted on the Tandem 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 Tandem to VM.

```
TANTOVM   PROCESS    PNODE=DALL.TX                                 -
                     SNODE=CD.VM.BOSTON                            -
                     SNODEID=(IDXXXX, PASSWD)                      -
                     SACCT='TRANSFERRING FROM TANDEM 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)
```

Copying a VSAM File from VM to an Entry-Sequenced Tandem File

The following Process copies a VM VSAM file to an entry-sequenced Tandem 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 Connect:Direct system assumes the file is a VSAM file.

```
TANDEM2  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))
```

Copying a VSAM File from VM to a Key-Sequenced Tandem File

The following Process copies a VM VSAM file to a key-sequenced Tandem 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 Connect:Direct system assumes the file is a VSAM file.

```
TANDEM6  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))
```

Copying Between Tandem and OpenVMS Nodes

Copying a Tandem Key-Sequenced File to a Connect:Direct OpenVMS Node

This Process, submitted from the Connect:Direct Tandem node, copies a key-sequenced file to the Connect:Direct OpenVMS node. When Processes are submitted from Connect:Direct Tandem to Connect:Direct 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.TANDEM                                        -
                    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)
```

Copying an OpenVMS Key-Sequenced File to a Connect:Direct Tandem Node

This Process, submitted from the OpenVMS node, will copy a key-sequenced file to the Connect:Direct Tandem node. When copying files from Connect:Direct OpenVMS to Connect:Direct Tandem nodes, include SET XLATE ON in the SYSOPTS parameter of the TO clause of the COPY statement. Because the Connect:Direct 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.TANDEM 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")
```

Copying Between Tandem and VSE Nodes

Copying Files from Tandem to VSE

This Process illustrates the transmission of files from a Tandem node to a VSE node. The parameter XLATE must be set to ON and positioned in the Process where the Tandem 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)
```

Copying a Tandem File to a VSE VSAM File

This Process copies a file from a Tandem 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=TANDEM.NODE                  -
                      SNODEID=(TANDEM.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)
```

Copying a VSE VSAM File to a Tandem Node

This Process copies a VSE VSAM file to a Tandem 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=TANDEM.NODE                 -
                     SNODEID=(TANDEM.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)
```

Copying Between UNIX Nodes

Copying 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
```

Copying Files and Using 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 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
```

Copying Files and Using 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;
```

Archiving Files Using the Connect:Direct 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;
```

Restoring Files Using the Connect:Direct 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;
```

Archiving and Restoring Files in a Single Step Using the Connect:Direct 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;
```

Copying Between UNIX and OS/400 Nodes

Copying Files from UNIX to a Member on OS/400

This Process copies an ASCII file from UNIX to a member on the OS/400.

```
* 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;
```

Copying Files from UNIX to a Member on OS/400

This Process copies an ASCII file from UNIX to a spool file on the OS/400. See the *Connect:Direct OS/400 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;
```

Copying Save Files from OS/400 to UNIX

This Process copies a save file from OS/400 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;
```

Copying Save Files from UNIX to OS/400

This Process copies a save file from UNIX to OS/400. Set **datatype=binary** for save files. Specify DCB information to copy a save file to OS/400.

```
* 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;
```

Copying Executables from UNIX to OS/400

This Process copies an executable from UNIX to OS/400. Specify FILETYPE(*DATA) in the **sysopts** parameter.

```
* COPY UNIX EXECUTABLE TO OS/400 *
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)
```

Copying Between UNIX and Windows Nodes

Copying a File from UNIX to Windows

This Process copies a binary file from a UNIX node to a Windows node. A TCP/IP address is specified instead of the 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)
               to   (file=d:\users\data\in\invoi01.dat
                    sysopts="datatype(binary)"
                    snode)
               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)
               to   (file=&file2
                    sysopts="datatype(binary)"
                    snode)
               pend;
```

Copying Between OpenVME Nodes

Copying Files and Using the Checkpointing Feature (OpenVME to OpenVME)

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

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

Copying Files Between OpenVME and UNIX

Copying Files and Using SYSOPTS (OpenVME to UNIX)

This Process copies a file from an OpenVME node to a UNIX node. 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=arch.dat
                    sysopts=":datatype=text:"
                    snode)
               to   (file=arch_no
                    sysopts=":datatype=text:strip.blanks=yes:"
                    pnode)
         pend
```

Copying Files and Using the Checkpointing Feature (UNIX to OpenVME)

This Process copies a file from UNIX to OpenVME. The **ckpt** parameter specifies that checkpoints will be taken at 128K intervals. If the COPY operation is interrupted, the 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=vme.node
step01   copy  from (file=pmts1
                    snode)
               ckpt=128k
               to   (file=pmts2
                    disp=new
                    pnode)
         pend
```

Copying Files Between OpenVME and Windows

Copying a File from OpenVME to Windows

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

```
ux2nt    process    snode=111.11.11.111
                    hold=no
                    retain=yes
copy1    copy  from (file=/acct1/out/tue07.dat
                    pnode)
               to   (file=d:\acct1\in\tue07.dat
                    sysopts=":datatype(binary):"
                    snode)
               pend;
```

Copying Between VM Nodes

Copying 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))
```

Copying an Entire 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)
```

Copying from 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 Connect:Direct 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.

Copying 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, Connect:Direct replaces it. If the file does not exist, the 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=\'FILETEST\&PROCESS.1\'\                    -
                     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=\'FILETEST\&PROCESS.2\'\                    -
                     DCB=(LRECL=80,RECFM=F)                          -
                     DISP=RPL                                        -
               )
```

continued

```
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=\'FILETEST\&PROCESS.3\'\                -
                     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=\'FILETEST\&PROCESS.4\'\                 -
                    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=\'FILETEST\&PROCESS.5\'\                -
                     DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)        -
                     DISP=RPL                                    -
               )
```

Extracting 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')                       -
```

Copying Between VM and OpenVMS Nodes

Copying 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)
```

Copying Between VM and VSE Nodes

Copying 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)
```

Copying a 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)
```

Copying 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)
```

Using a Typekey to Copy a 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)
```

Copying Between VM and OS/400 Nodes

Copying Files from VM to OS/400

This Process copies a sequential file from a Connect:Direct VM node to an OS/400 node. The SYSOPTS parameter specifies that the data is to be copied to the Connect:Direct OS/400 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)
```

Copying VSAM Files from VM to OS/400

This Process copies a VSAM file from a Connect:Direct VM node to a sequential file on a Connect:Direct OS/400 node. The DSN parameter on the COPY TO side specifies the destination object name based on the OS/400 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 OS/400 to VM

This Process is initiated from the Connect:Direct VM/ESA to copy a file from an OS/400 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 OS/400 Spool

This Process is initiated from the Connect:Direct VM/ESA to copy a file from a VM node to a spooled output file on an OS/400 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 OS/400 to VM Spool

This Process is initiated from the Connect:Direct VM/ESA to copy a physical database file member from a Connect:Direct OS/400 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)
```

Copying a VM VSAM file to OS/390

Copying a VM VSAM file to OS/390

This Process copies a VSAM Disk from a VM/ESA node to a sequential disk file on an OS/390 node. If the file exists, Connect:Direct replaces it. If the file does not exist, Connect:Direct creates it as indicated by the DISP parameter.

```
VSMtoOS390  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)
```

Copying a CMS Disk File to an OS/390 Node

This Process copies a CMS Disk file from a VM/ESA node to a disk on a Connect:Direct OS/390 node. If the file exists, Connect:Direct replaces it. If the file does not exist, Connect:Direct creates it as indicated by the DISP parameter.

```
VMTOOS390  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)
```

Copying Between VM and UNIX Nodes

Copying 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:"                -
             )
```

Copying VM and Windows Nodes

Copying a CMS Sequential File from VM to Windows 95

This Process copies a CMS Sequential file from VM to Windows 95.

```
PROC01   PROCESS                                                    -
               &CKPT=0K                                             -
               &COMPRESS=,                                          -
               &EXT=,                                               -
               &CUU1=0192                                           -
               &USER=MYUSR1                                         -
               &USERPW=ALL                                          -
               &SNODE=WIN95                                         -
               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                                           -
               )
```

Copying a VM CMS File to Windows

This Process copies a VM CMS file to 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)
```

Copying Between OpenVMS Nodes

Copying a 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,PASSWORD)
STEP01    COPY  FROM  (FILE=DUXX:[DIRECTORY]TESTFILE.DAT)                 -
                TO    (FILE=MUXX:TESTFILE.DAT                             -
                      SYSOPTS="MOUNT='MUXX: TAPE /OVERRIDE=ID' ")
```

Copying a 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,PASSWORD)
STEP01    COPY  FROM  (FILE=MUXX:TESTFILE.DAT                             -
                      SYSOPTS="MOUNT='MUXX: TAPE /OVERRIDE=ID' ")         -
                TO    (FILE=DUXX:[DIRECTORY]TESTFILE.DAT)
```

Using Symbolics in a COPY Statement (Connect:Direct OpenVMS)

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")
```

Copying a 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))
```

Copying Between OpenVMS and VSE Nodes

Copying 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)
```

Copying Between VSE Nodes

Copying 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))
```

Copying 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)
```

Copying the Connect:Direct Message File to SL Tape

This Process copies the Connect:Direct OS/390 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)
```

Copying a Non-managed Disk Data Set into Another Non-managed CKD Disk Data Set

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 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
```

Copying a Noncontrolled Disk Data Set to a Managed CKD Disk Data Set

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 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
```

Copying a Nonmanaged Disk File into a Start Track 1 FBA Noncontrolled Data Set

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

This Process runs on the same Connect:Direct node using PNODE=SNODE processing and uses symbolic values. CA-DYNAM/D or CA-EPIC will perform the disk allocation for 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
```

Copying to Non-TMS Controlled Tapes

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 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
```

Copying a Nonmanaged Disk File to a CA-DYNAM/D or CA-EPIC Start Track 1 FBA Noncontrolled Data Set

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

This Process uses symbolic values. CA-DYNAM/D or CA-EPIC will perform the disk allocation for 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
```

Printing a Managed Disk Data Set

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 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.

Copying a Noncontrolled Sequential File to a MSAM File

This Process copies a noncontrolled 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 Connect:Direct node using PNODE=SNODE processing and uses symbolic values.

When you reference BSAM libraries in a 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
```

Copying a Noncontrolled Tape Data Set to a Controlled Disk File

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 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)             -
                    )
```

Copying Nonmanaged Disk Data Set to a Nonmanaged Tape Data Set

This Process copies a nonmanaged disk data set into a nonmanaged tape data set and runs on the same 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
```

Copying a Managed Disk Data Set to Another Managed Data Set

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
```

Copying a Managed Generation Disk Data Set to Another Managed Data Set

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 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
```

Copying a Controlled Disk Data Set to a Controlled Tape Output File

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 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
```

Copying a Controlled BSAM Data Set to a MSAM Output Data Set

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 Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

Note: You can ignore the 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 Connect:Direct node using PNODE=SNODE processing. This Process uses symbolic values.

Copying a Controlled Tape Data Set to a Controlled FBA Disk Output Data Set

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 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
```

Copying a Controlled CKD Disk Data Set to a Noncontrolled Tape Data Set

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 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
```

Copying 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 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 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
```

Copying 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 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 Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

Note: 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
```

Copying 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 Connect:Direct node using PNODE=SNODE processing and uses symbolic values.

Copying 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 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
```

Copying a BSAM VSE Sublibrary to a New OS/390 PDS

This Process sends an entire BSAM VSE sublibrary into a new OS/390 partitioned data set. The Process runs on the PNODE (VSE) and sends the data to the SNODE (OS/390). This Process uses symbolic values.

When you reference BSAM libraries in a 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. An example follows.

```
LIB2OS3904 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
```

Copying a MSAM Data Set to a Controlled BSAM Data Set

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 Connect:Direct node using PNODE=SNODE processing. This Process uses symbolic values.

When you reference BSAM libraries in a 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.

Copying Between VSE and OS/400 Nodes

Copying a VSE VSAM to an OS/400 PDS Member

This Process copies a VSAM file from a VSE node to an OS/400 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)
```

Copying a VSE VSAM File to an OS/400 Spooled File

This Process copies a VSAM file from a VSE node to an OS/400 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)
```

Copying Between VSE and OS/390 Nodes

Copying a VSE Librarian BSAM Member to a Preallocated OS/390 PDS Member

This Process copies a VSE Librarian BSAM member into a preallocated OS/390 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 Connect:Direct Process, you must specify: DSORG, DSN, UNIT, and VOL=SER= parameters.

```
LIB2OS3901 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
```

Copying a VSAM VSE Library Member to a Preallocated OS/390 PDS Member

Use the Process to copy a VSAM VSE Library member into a preallocated OS/390 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 Connect:Direct Process you must specify the DSORG and DISP parameters.

```
LIB2OS3902 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
```

Copying a VSE/POWER LST Queue Member to a Preallocated OS/390 PDS

Use this Process to copy a VSE/POWER LST queue member into a preallocated OS/390 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.

```
LST2OS3901 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
```

Copying Files Between VSE and UNIX

Copying a File from UNIX HP to a Controlled Disk Data Set 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
```

Copying a File from HP UNIX to a 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
```

Copying Between Windows and OS/390 Nodes

Copying a File from Windows to OS/390

This Process copies a text file from a remote Windows system (where Connect:Direct is not installed) to an OS/390 partitioned data set (PDS). When copying a file to or from a remote 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. In this example, the computer name is WIN95_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=\\WIN95_SYS1\ROOT_C\DATA\OUT\SALESJAN.DAT
                     PNODE/*$WINDOWS$*/
                     SYSOPTS="datatype(text)")
                TO   (SNODE/*$OS390$*/
                     FILE=SALES.DATA.JAN(MBR99)
                     DISP=(RPL,CATLG))
          PEND
```

Copying Between Windows and Tandem

Copying a File from Windows to Tandem

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

```
NT2TAN    PROCESS     SNODE=SS.TAN/*$TANDEM$*/
                      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/*$TANDEM$*/
                      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
```

Chapter 5

RUN JOB Statement Examples

This chapter contains examples of Connect:Direct RUN JOB statements for various Connect:Direct platforms.

Submitting a Job to the OS/390 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 Connect:Direct OS/390.

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 OS/390 internal reader on CDB.

Note: 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 OS/390 internal reader on the PNODE system.

```
PROC1    PROCESS    SNODE=CDB
STEP01   RUN JOB    (DSN=JOB.STREAM.LIB(JOB123) PNODE)
```

Note: The JCL must exist on the node where you want it to execute.

Submitting 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'"
```

Printing and Deleting the Log File (Connect:Direct 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)
```

Keeping the Log File (Connect:Direct 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"
```

Printing and Keeping the Log File (Connect:Direct 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 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)
```

Running a Job on the OS/390 Node from a Process Submitted on the Tandem Node

In this Process submitted from the Tandem node, STEP1 will execute FUP to purge $B.FILERESO.STEST. STEP2 will copy DATA1.SMALLER from the OS/390 node to $B.FILERESO.STEST at the Tandem 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 OS/390 node.

Note: The Connect:Direct Tandem system cannot execute the RUN JOB statement; however, the Connect:Direct Tandem node as the PNODE can submit a Process to an OS/390 or VSE node, and the SNODE can execute the RUN JOB.

```
RUN       PROCESS    PNODE=CD.TANDEM                                        -
                     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)
STEP3     IF         (STEP1 GT 4)  THEN
                EXIT
                EIF
STEP4     RUN JOB    (DSN=DATA1.CNTL(IEFBR14A)) SNODE
```

Running a Job on the OS/400 Node from a Process Submitted on the OS/390 Node

This example is submitted on OS/390 to run a job on OS/400. DSN is necessary when submitting from OS/390. The contents of the SYSOPTS parameter define the program to run on the OS/400 node.

```
RJPROC01  PROCESS    SNODE=CD2200                                       -
                     SNODEID=(USER1,PASSWD1)
STEP0001  RUN JOB    (DSN=AS400)  SNODE                                 -
                     SYSOPTS=\"\                                        -
                             \CMD(\                                     -
                             \CALL\                                     -
                             \PGM(KRM/KJOBTST)\                         -
                             \)\                                        -
                             \"\
```

Running 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
```

Executing Commands on UNIX from a Process Submitted from OS/390

This Process shows how to initiate a Process from OS/390 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"
```

Running a Job on OS/390 from a Process Submitted on UNIX

This Process shows how to initiate a Process from UNIX to run a job at an OS/390 node. The DSN string must be in uppercase characters to satisfy OS/390 syntax requirements.

```
proc2     process    snode=OS390.node
                     snodeid=(user01,pswd01)
step01    run job    (dsn=SRCDATA.SET(TEST))
                     snode
                pend
```

Executing Commands on UNIX from a Process Submitted from Tandem

This Process shows how to initiate a Process from Tandem 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
```

Running a Job on OS/390 from a Process Submitted on OpenVME

This Process shows how to initiate a Process from OpenVME to run a job at an OS/390 node. The DSN string must be in uppercase characters to satisfy OS/390 syntax requirements.

```
proc2    process    snode=OS390.node
                    snodeid=(usera,pswda)
step01   run job    (dsn=NEWDATA.SET(TEST))
                    snode
              pend
```

Running a Job on Windows from a Process Submitted on UNIX

This Process shows how to initiate a Process from UNIX that executes commands at a 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 console prompt directly on the Windows system.

```
ntsub1   process    snode=cd.nt
                    retain=yes
                    prty=yes
runj01   run job    snode
                    sysopts="cmd(dir d:\users\jdoe1\*.*
>>d:\users\jdoe1\list.out)"
         pend;
```

Use the previous example as a Windows 95 example by using a Windows 95 node for the snode parameter.

Chapter 6

RUN TASK Statement Examples

This chapter contains examples of Connect:Direct RUN TASK statements for various platforms.

RUN TASK Examples for CA-7 (OS/390 to OS/390)

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.

Note: 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, Connect:Direct OS/390 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=(\'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

This Process, SAMPLE1, is submitted from a Connect:Direct OS/390 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
```

Creating and Saving an Online Save File Through Connect:Direct OS/400

This RUN TASK statement creates the online save file named SAVEFILE1 in the library TESTSV1 on the OS/400. 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. 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) \                    -
                             \    ) \                                       -
                             \"\
```

Copying a Member of an Object from OS/400 to a PDS Member and then Deleting the Library

This Process copies a member of an object from OS/400 to a member of a PDS on OS/390. The RUN TASK then deletes the library (with all its objects) from the OS/400 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) \                            -
                             \ ) \                                     -
                             \"\
```

Notifying the OS/400 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) \                         -
                             \ ) \                                     -
                             \"\
```

Restoring Libraries Through Connect:Direct OS/400

This Process restores to the OS/400 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) \                        -
                            \ ) \                                      -
                            \"\
```

Submitting a Process with a RUN TASK on OpenVMS from an OS/390 Node

The following example shows a Process with a RUN TASK statement submitted from OS/390 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’"\
```

Initiating a RUN TASK Statement at the Tandem Node (OS/390 to Tandem)

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 Connect:Direct Processes can continue uninterrupted in the event the program being executed abends. The Process is submitted on OS/390.

```
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
```

Using Symbolics with a RUN TASK Statement (OS/390 to Tandem)

This Process is submitted from the OS/390 node to run a program at the Tandem node. The Process is structured so that the program name (PGM) and associated run-options for the Tandem 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, Tandem will assign a name to the Tandem process.

```
PROCESS1  PROCESS    PNODE=CD.OS390                                       -
                     SNODE=CD.TANDEM
STEP01    RUN TASK   (PGM = &PGM                                          -
                     PARM=(\ "/IN    \ || &INFILE  ||                     -
                           \  ,OUT   \ || &OUTFILE ||                     -
                           \  ,PRI   \ || &PRI     ||                     -
                           \  ,CPU   \ || &CPU     ||                     -
                           \  ,NAME  \                                    -
                           \  /")\)                                       -
                     SNODE
```

The following Connect:Direct syntax rules apply to this Process:

- The string of Tandem RUN command options must be enclosed in forward slashes (/). This is a Tandem 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 Tandem RUN command must be separated by commas.

- Because the PNODE is the OS/390 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
```

Using Bracketing Backslashes and Quotation Marks (OS/390 to Tandem)

PROCESS1 is coded to be submitted from the OS/390 node to run a program at the Tandem node. The RUN TASK statement executes FUP to copy ACCTJAN with shared access to the spooler, $S.#SPL1, at the Tandem node. Any Tandem process-related error messages are directed to $TERM1. Connect:Direct-related messages are directed to $S.#SPL1.

PROCESS1 shows a Process with a parameter *(*PARM*)* continuing over multiple lines.

```
PROCESS1  PROCESS    PNODE=CD.OS390                                       -
                     SNODE=CD.TANDEM                                      -
                     SNODEID=(127.201,RSMITH)
STEP01    RUN TASK   (PGM = FUP                                           -
                     PARM=(\ "/OUT $S.#SPL1    \ ||                       -
                           \  ,TERM $TERM1/    \ ||                       -
                           \ COPY $SYSTEM.BILLPROC.ACCTJAN,,SHARE") \)    -
                     SNODE
```

The following Connect:Direct and Tandem syntax rules apply to both of these Processes:

- Within a Connect:Direct Process submitted from an OS/390 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 Tandem RUN command options must be enclosed in forward slashes (/). This is a Tandem syntax requirement.
- Because the PNODE is an OS/390 node (that is, the Process is submitted on the OS/390 node), backslashes and vertical bars must be used to continue a string across multiple lines.

Note: Bracketing backslashes are not valid when the PNODE is a Tandem node.

Using Concatenation Characters (OS/390 to Tandem)

PROC1, PROC2, and PROC3 demonstrate the use of concatenation characters within a Connect:Direct Tandem RUN TASK statement.

```
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 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 Connect:Direct Tandem 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
```

Running FUP (Connect:Direct Tandem)

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.

```
PROC1    PROCESSSNODE=SYSCLX
STEP1    RUN TASK   (PGM=FUP                                              -
                    PARM=('/NAME $FP,OUT $S.#SPL1/'                       -
                          ,'COPY $A.PROCVOL.ACCT,$S.#SPL2')               -
                                                                          -
                    PNODE
```

Running RESTORE (Connect:Direct Tandem)

This Process starts RESTORE from a Connect:Direct Process. Output from the RESTORE command (success/failure) is sent to $S.#SPL1. An IN file can be used as well. If an abend occurs, any abend message will be sent to the device (in this case, $TAN.#T5) specified by the TERM command.

```
PROC1    PROCESS    SNODE=SYSCLX
STEP1    RUN TASK   (PGM=RESTORE                                          -
                    PARM=('/OUT $S.#SPL1,NAME $REST,TERM $TAN.#T5/'       -
                          ,'$REEL,,LISTALL,NOUNLOAD')                     -
                    )                                                     -
                    PNODE
```

Selecting Statistics for the Current Day (Connect:Direct Tandem)

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 OS/390, where it can be printed or viewed.

```
RUNT0005  PROCESS    SNODE=TANDEM.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)                               -
                     )
```

Submitting a Process with a Connect:Direct OpenVMS RUN TASK from a Tandem Node

This Process, submitted from the Connect:Direct Tandem 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.TANDEM                                    -
                     SNODE=CD.VMS
STEP1     RUN TASK   (PGM=VMS) SNODE                                    -
                     SYSOPTS="OUTPUT = '_NDC31'                         -
                              CMD = 'DIR DUC4:[USER1.DATA]*.DAT'        -
                              CMD = 'REPLY/USER=USER1/BELL SUCCESS'"
```

Submitting a Process with a Connect:Direct OpenVMS RUN TASK from an OS/400 Node

The example command sends a file from Connect:Direct OS/400 to Connect:Direct OpenVMS and performs a RUNTASK.

```
CDSND  SNODE(DWY1.TCP)
SNODENVIRN(OPENVMS)
FDSN('CDABC220/INITPARMS(INITPARMS)')
TDSN('DISK$SUP:<DYOUN1>AS400.RCV') FMSYSOPTS('TYPE(MBR)')
SNODEID(USERID PASSWORD) TDISP(RPL)TDCB(*N *N *N PS)
CDRUNTASK  SNODE(DWY2.TCP)
SNODENVIRN(OPENVMS)
CMD('CMD=''`DEL
DISK$SUP:<DYOUN1>AS400.RCV;*''')
SNODEID(USERID PASSWORD)
```

Submitting a Process with a Connect:Direct OS/390 RUN TASK from a Tandem Node

This Process, submitted from the Connect:Direct Tandem node, performs a RUN TASK on the Connect:Direct OS/390 node, then copies from Tandem to OS/390.

```
TANCPY   PROCESS    SNODE=CD.TANDEM                                    -
                    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                                         -
                    )
```

Submitting a Process with a Connect:Direct Tandem RUN TASK from an OpenVMS Node

This Process, submitted from the Connect:Direct OpenVMS node, will invoke FUP at the Connect:Direct Tandem node and write detailed information about the files in the subvolume to the spooler.

```
TANRUN   PROCESS    PNODE=CD.VMS                                          -
                    SNODE=CD.TANDEM SNODEID=(GRP.USR,PASWRD)
STEP1    RUN TASK   (PGM=FUP                                              -
                    PARM=(’/OUT $S.#FUPOUT/’                              -
                          ’INFO $DATA.TESTF.*,DETAIL’))                   -
                    SNODE
```

Submitting a Process with a Connect:Direct Tandem RUN TASK from a Windows Node

The following is an example of a Connect:Direct Tandem RUN TASK submitted from a Connect:Direct Windows node.

```
NDM PROCESS  SNODE=NDM.BAY1DR /*$TANDEM$*/
             SNODEID=(NCC.NDM,xxxxxxxx)

BEGINMSG  RUN TASK (PGM = BTFNDMLG
          SYSOPTS = "/NAME, IN $DATA2.BTFDATA.DEFTABLE, VOL $DATA2.BTFLOAD -
                     /ND033 DNLOAD FUNBCC FUNBBETA TRUO5")
          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 TRUO5")
           SNODE
           EXIT

OKMSG  RUN TASK (PGM = BTFNDMLG
       SYSOPTS = "/NAME, IN $DATA2.BTFDATA.DEFTABLE, VOL $DATA2.BTFLOAD -
                  / ND034 DNLOAD FUNBCC FUNBBETA TRUO5")
       SNODE

FT3IN  RUN TASK (PGM = BTFNDMDN
       SYSOPTS = "/NAME, IN $DATA2.BTFDATA.DEFTABLE, VOL $DATA2.BTFLOAD -
                  / DNLOAD FT3IN $test.btfdata.beta5 FUNBCC FUNBBETA TRUO5")
```

Executing 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   PROCESSSNODE=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
```

Resolving Symbolics Within DMRTDYN in a RUN TASK Environment (VM)

This example illustrates the structure of a 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=\'MTT\    &XXX    \ FILETYPE'\
LOCATE1   RUN TASK   (PGM=DMRTDYN                                           -
                     PARM=(C'ALLOC ',                                       -
                           C' DSN='    \&VMFILE\    \'',\                   -
                           C' DISP=(SHR) LINK=(IVVB,WIVVB,RR,191)',         -
                           C' DD=F1')) PNODE
LOCATE3   RUN TASK   (PGM=DMRTDYN                                           -
                     PARM=(C'UNALLOC '                                      -
                           C' DD=F1')) PNODE
```

Submitting a Process with a RUN TASK on OS/400 from a VM Node

This example is initiated on a Connect:Direct VM/ESA node to execute a RUN TASK on Connect:Direct OS/400. The SYSOPTS parameter specifies the OS/400 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) \                                -
                             \ ) \                                         -
                             \"\
```

Submitting 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'")
```

Defining a VSE VSAM File and Copying a Sequential File from OS/390

This multistep Process, initiated from an OS/390 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 Connect:Direct 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 OS/390 node to a VSE node.

```
VSEVSAM  PROCESS    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)
```

Submitting a Process from OS/390 to Execute UNIX Commands

This Process initiates a Process from OS/390 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"
```

Submitting a Process from OS/390 to Execute Windows Programs

This Process initiates a Process from OS/390 that runs a program on 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=\’CMD(D:\\CDCLNT\\PROGPACK.EXE           -
                   D:\\CDCLNT\\PROGRAM.PAC\\ D:\\CDCLNT\\*.*) DESKTOP (YES)’\
```

Submitting a Process from UNIX to Run a Program on OS/390

This example shows a Process initiating from UNIX that runs a program on OS/390. The **sysopts** string must be in uppercase characters to satisfy OS/390 syntax requirements.

```
proc2    process    snode=OS390.node
step01   run task   snode (pgm=DMNOTIFY)
                    sysopts="CL44’DATA.BASE.P1’,F’0010’, XL8’FFA8’"
         pend
```

Submitting 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
```

Submitting a Process from UNIX to Run a Program on OS/400

This Process initiates a Process from a UNIX node that executes commands on an OS/400 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;
```

Submitting a Process from UNIX to Run a Program on Windows

This Process copies a binary file from a Windows node to a UNIX node. If the copy is successful, a **run task** statement is performed on the Windows node (**snode**) that will delete the source (from) file on the Windows node. To delete the file, the keyword **cmd** is specified in the **sysopts** followed by the **del** command.

```
proc1    process    snode=CD.NT
                    &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 Windows desktop.

Submitting a Process with a Run Task on an OpenVME Node to Run a Program on Another OpenVME Node

This Process initiates a Process from an OpenVME node that executes commands at a remote OpenVME node. The **sysopts** string must be enclosed in double quotation marks.

```
proc2    process    snode=vme.node
step01   run task   snode sysopts = "ls; ls -lax > lsout.ncr"
         pend
```

Submitting a Process from OpenVME to Run a Program on Windows

This Process copies a binary file from a Windows node to an OpenVME node. If the copy is successful, a **run task** statement is performed on the Windows node (**snode**) that deletes the source (from) file on the Windows node. To delete the file, the keyword **cmd** is specified in the **sysopts** followed by the **del** command. 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 Windows desktop.

This example illustrates a Process from OpenVME to run a program on Windows NT. To use this illustration as a Windows 95 example, you must specify the snode as Windows 95.

```
proc1    process    snode=CD.NT
copy1    copy  from (file="d:\data\out\monthend.dat"
                    sysopts="datatype(binary)"
                    snode)
               to   (file="/data/in/monthend.dat"
                    pnode)
         if (copy1 eq 0) then
run1     run task   snode
                    sysopts="cmd(del "d:\data\out\monthend.dat")desktop(no)"
         eif
         pend;
```

Notifying the OS/400 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
```

Submitting a Process from Connect:Direct Requester for Windows to Run DMRTSUB on OS/390

The following Process is submitted from Connect:Direct Requester for Windows to run DMRTSUB on a Connect:Direct OS/390 node. This Run Task using DMRTSUB will submit a job to run on OS/390 and pass the symbolic for &NTDISP to the JCL it submits.

Note: Refer to the *Connect:Direct OS/390 User's 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,PASSWORD)

RTASK01 RUN TASK SNODE (PGM=DMRTSUB)
       SYSOPTS=”C’DSN=JSMITH.PROCESS(SAMPLE),DISP=SHR’
C’NTDISP &NTDISP’”

     PEND
```

Chapter 7

SUBMIT Statement Examples

This chapter contains examples of Connect:Direct SUBMIT statements. The examples include SUBMIT statements for cross platform Processes.

Using SUBMIT with Symbolic Substitution (OS/390 to OS/390)

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 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.

Using SUBMIT with the DSN Parameter (OS/390 to OS/390)

In this example, symbolics for PROCESS2 are supplied by the operator submitting PROCESS1.

The operator at CD.LA issues the following 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.

Submitting a Windows Process from a UNIX Node

This Process copies a binary file from a Windows node to a UNIX node. If the copy is successful, a **submit** of another process is performed on the CD.NT node by setting the **subnode** parameter to SNODE.

```
proc1     process    snode=CD.NT
                     &file1="d:\testdat\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:\testdat\nt2nwcp.cdp"
                     subnode=snode
          eif
          pend;
```

The previous example illustrates a submitting a Windows NT Process from a UNIX node. To use this illustration as a Windows 95 example, you must specify a valid Windows 95 SNODE.

Using SUBMIT at the Connect:Direct Tandem Node

In this Process, STEP1 will execute FUP to purge $B.FILERESO.FILEA. STEP2 and STEP3 will submit Connect:Direct Processes at the Tandem node (PNODE).

```
SUBMIT    PROCESS    PNODE=CD.TANDEM                               -
                     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
```

Using submit with the hold Parameter (UNIX to UNIX)

This Process shows how to submit another 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
```

Using submit with the startt Parameter (UNIX to UNIX)

This Process shows how to submit another 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, 1999.

```
submit1   process    snode=unix.node
step01    submit     file=copy.cd
                     startt=(12/14/1999,23:30)
          pend
```

Chapter 8

SYMBOL Statement Examples

This chapter contains examples of Connect:Direct SYMBOL and symbolic statements for various Connect:Direct platforms.

Using the SYMBOL Statement to Construct a Symbolic Value for Data Set Names (OS/390)

This Process illustrates a Connect:Direct Process that uses the SYMBOL statement to construct a symbolic value for DSN1 and DSN2. These SYMBOL statements must be resolved when submitting the Process. In addition, the SNODE is a symbolic that also must be resolved when the Process is submitted.

Note: Two vertical bars preceded and followed by blanks (||) are used to indicate concatenation. Bracketing backslashes ensure that special characters within the string are maintained.

As specified by the REQUEUE parameter in the PROCESS statement, the Process is requeued in the event of an x37-type error. The Process is then retained in the queue and rerun every Monday at 8:30 a.m. as specified by the RETAIN and STARTT parameters.

```
SYMPROC   PROCESS     SNODE=&SNODE                                          -
                      NOTIFY=%USER                                          -
                      CLASS=T                                               -
                      REQUEUE=YES                                           -
                      PRTY=2                                                -
                      STARTT=(MO,08:30:00AM)                                -
                      RETAIN=YES
          SYMBOL      &DSN1=&HLQ || \.\ || &D1
          SYMBOL      &DSN2=&HLQ || \.\ || &D2
STEP01    COPY  FROM  (DSN=&DSN1)                                           -
                TO    (DSN=&DSN2                                            -
                      DISP=RPL)
```

To submit the Process on a Connect:Direct OS/390 node, issue the following Submit Process command:

```
SUB PROC SYMPROC &HLQ=JSMIT &D1=FILEA &D2=FILEB &SNODE=CD.SAB.F5
```

Using Symbolics Within DMRTSUB to Substitute a Windows Pathname into OS/390 JCL

This example shows a Connect:Direct Process that passes a Windows pathname to DMRTSUB. The symbolic substitution accommodates the backslashes in the Windows pathname.

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

```
SUB1     PROCESS    SNODE=SC.OS390.USER1                                  -
                    PNODE=SC.OS390.USER1                                  -
                    &TST1=’\C:\\TEST\\DATA.TXT\’                          -
                    &CASEPARM=CASE7                                       -
STEP1    RUN TASK   ( PGM=DMRTSUB                                         -
                    PARM=(DSN=USER1.CD211.PROCESS(SUB1BR14),DISP=SHR",    -
                    “NTDS &CASEPARM”,                                     -
                    “TST1 &TST1" ) )    SNODE
```

The following JCL is present in member SUB1BR14 of data set USER1.CD211.PROCESS for the above example.

```
//USERC1N JOB (11111),’USER 1’,CLASS=C,REGION=4096K,
//     MSGLEVEL=(1,1),MSGCLASS=X,NOTIFY=USER1
//*
//ALLOC   EXEC PGM=IEFBR14
//* NTDS=&NTDS
//* TST1=&TST1
```

Using SYMBOL Statements for Input and Output Data Sets (OS/390 and OS/400)

In this Process, SYMBOL statements are used for input/output data sets. SYMBOL statements have also been defined for DISP fields. The following describes the steps in the Process.

- STEP01 copies a sequential file from OS/390 to a member of a physical data base file on the Connect:Direct OS/400 node. The OS/390 input file is BUDGET.PROD.DAILY. The output data set on the Connect:Direct OS/400 node is TESTLIB/BUDGET.
- STEP02 copies a member of a physical data base file on the Connect:Direct OS/400 node to a sequential file on OS/390. The input data set from the Connect:Direct OS/400 node is PRODLIB/BUDGET. The OS/390 output file is BUDGET.TEST.DAILY.

```
PROC#01  PROCESS    SNODE=AS400.NY1                                 -
                    PRTY=8                                          -
                    NOTIFY=RSMITH                                   -
                    CLASS=4                                         -
                    &PROD=PROD                                      -
                    &TEST=TEST                                      -
                    SNODEID=(RSMITH,ROGER)
         SYMBOL     &DSN1I=\BUDGET.\ || &PROD || \.DAILY\
         SYMBOL     &DSN1O='TESTLIB/BUDGET'
         SYMBOL     &DSN2I='PRODLIB/BUDGET'
         SYMBOL     &DSN2O=\BUDGET.\ || &TEST || \.DAILY\
         SYMBOL     &DISP01=SHR
         SYMBOL     &DISP02=RPL
STEP01   COPY  FROM (PNODE                                          -
                    DSN=&DSN1I                                      -
                    DISP=&DISP01)                                   -
               COMPRESS                                             -
               TO   (SNODE                                          -
                    DSN=&DSN1O                                      -
                    SYSOPTS=\"\                                     -
                            \TYPE(MBR) \                            -
                            \TEXT('ADDED BY PROCESS PROC#01 \       -
                            \IN STEP01') \                          -
                            \"\                                     -
                    DISP=RPL)
STEP02   COPY  FROM (SNODE                                          -
                    DSN=&DSN2I                                      -
                    SYSOPTS="TYPE(MBR)"                             -
                    DISP=SHR)                                       -
               COMPRESS                                             -
               TO   (PNODE                                          -
                    DSN=&DSN2O                                      -
                    DCB=(DSORG=PS,RECFM=FB,LRECL=080,BLKSIZE=23440) -
                    SPACE=(CYL,(15,1),RLSE,CONTIG,ROUND)            -
                    DISP=&DISP02)
```

Note: Two vertical bars preceded and followed by blanks (||) are used to indicate concatenation. Bracketing backslashes ensure that special characters within the string are maintained.

Using Symbolics in a COPY statement (OS/390 to UNIX)

In the following example, the Process, PROCVAR, is copying a file from OS/390 to UNIX. Two variables are stated in the SYSOPTS parameter. The Process will not work if the variables are stated as shown. To substitute two or more variables in the SYSOPTS statement, you must use the SYMBOL statement to identify the variables.

```
PROCVAR   PROCESS    SNODE=&SNODE                                        -
                     &JCLLIB=USER01.DALLAS.CNTL                          -
                     &FROMDISP=SHR                                       -
                     &TODISP=RPL                                         -
                     &TOPERMISS=770                                      -
                     &STRIP=NO                                           -
                     &COMPRESS=,                                         -
STEP01    COPY  FROM (DSN=&FROMDSN                                       -
                     DISP=&FROMDISP                                      -
                     )
                TO   (DSN="&TODSN"                                       -
                     DISP=(&TODISP)                                      -
                     SYSOPTS=":permiss=\ &TOPERMIS :strip.blanks= &STRIP:" -
                     )                                                   -
                     &COMPRESS
```

The following Process correctly uses the SYMBOL statement to declare the substitution of more than one variable in the SYSOPTS statement. The COPY from OS/390 to UNIX with certain permission and no blank stripping is successful.

```
PROCVAR   PROCESS    SNODE=&SNODE                                        -
                     &JCLLIB=USER01.DALLAS.CNTL                          -
                     &FROMDISP=SHR                                       -
                     &TODISP=RPL                                         -
                     &TOPERMISS=770                                      -
                     &STRIP=NO                                           -
                     &COMPRESS=,
          SYMBOL     &BLANKS=&STRIP
          SYMBOL     &PERMISS=&TOPERMIS
          SYMBOL     &SYSOPTS=\"\||:permiss=||                           -
                     &PERMISS || :strip.blanks= || &BLANKS||\:\||\"
STEP01    COPY  FROM (DSN=&FROMDSN                                       -
                     DISP=&FROMDISP                                      -
                     )
                TO   (DSN="&TODSN"                                       -
                     DISP=(&TODISP                                       -
                     SYSOPTS=&SYSOPTS                                    -
                     )                                                   -
                     &COMPRESS
```

Using Symbolics in a COPY Statement (Connect:Direct Tandem)

This example shows the basic use of symbolics in a Connect:Direct Process submitted from the Tandem node. Both the FROM and TO DSNs are resolved at Process submission.

```
SYMBOLS   PROCESS     PNODE=CD.TANDEM                                        -
                      SNODE=CD.OS390.NODE                                    -
                      &DSN1=DATA1.FILXYZ                                     -
                      &DSN2=$B.FILERESO.SYMBTEST
STEP1     RUN TASK    (PGM=FUP                                               -
                      PARM=('/OUT $S.#SYMBOL/',                              -
                            'VOLUME $B.FILERESO',                            -
                            'PURGE SYMBTEST  '))
STEP2     COPY  FROM  (DSN=&DSN1 DISP=SHR SNODE)                             -
                TO    (DSN=&DSN2 DISP=NEW PNODE)
```

The Process, SYMBOLS, can be submitted with the following example command:

```
SUBMIT FILE SYMBOLS
```

Using Symbolics in a COPY Statement (Tandem to UNIX)

This example shows the basic use of symbolics in a Connect:Direct Process submitted from the Tandem node. Both the FROM and TO DSNs are resolved at Process submission. STEP01 will COPY FROM a Tandem node TO a UNIX node. STEP02 will COPY FROM the UNIX node TO another Tandem node. SYSOPTS specifies how the Tandem file will be created.

```
TOFROMK   PROCESS     snode=cd.v1200                                         -
                      snodeid=(user,pswd)
                      symbol &file1=\clx.$qa1.filetest.ksds0007
                      symbol &file2='tdata/ksdstxt'
                      symbol &file3=$user.FROMUNIX.ksdstxt
STEP01    COPY  FROM  (DSN=&file1                                            -
                      PNODE                                                  -
                      SYSOPTS=('SET XLATE OFF')                              -
                      DISP=SHR)                                              -
                TO    (dsn=&file2                                            -
                      SNODE                                                  -
                      DISP=RPL)
STEP02    COPY  FROM  (DSN=&file2                                            -
                      SNODE                                                  -
                      DISP=SHR)                                              -
                TO    (DSN= &file3                                           -
                      DISP=RPL                                               -
                      SYSOPTS=("set type K"                                  -
                               "set code 0"                                  -
                               "set rec 100"                                 -
                               "set keylen 10"                               -
                               "set keyoff 0"                                -
                               "set block 4096")                             -
                      PNODE)
```

Passing Parameters to a DCL Command Procedure from a RUN JOB Statement (Connect:Direct OpenVMS)

This example shows an example Connect:Direct Process that allows you to pass parameters to a DCL command procedure from a RUN JOB statement.

```
SYMBOL3   PROCESS    SNODE=CD.VMS  &P1=,
          SYMBOL     &SO="P1=&P1"
STEP01    RUN JOB    (DSN=TEST.COM PNODE)                              -
                     SYSOPTS=&SO
```

The Process, SYMBOL3, can be submitted with the following example command issued in DCL command format:

```
$NDMUI SUBMIT SYMBOL3 /SYMBOLICS=("P1=‘PARM1’")
```

Chapter 9

Conditional Statements Examples

This chapter contains examples of conditional statements within Processes. An example of conditional logic statements in a cross platform Process is included.

Using Conditional Statements

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 OS/390. 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
```

Using Conditional Logic (OS/390 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 OS/390 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
```

Using Conditional Logic (Connect:Direct OS/400 to Connect:Direct OS/390)

This Process copies a file from an Connect:Direct OS/400 node to OS/390. 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)\                               -
                            \)\                                           -
                            \"\
```

Using Conditional Logic Where the PNODE is a Connect:Direct Tandem Node

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 OS/390 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 OS/390 node to $B.FILERESO.FILE2 at the Tandem node.

STEP04 will then execute and copy DATA1.FILEC from the OS/390 node to $B.FILERESO.FILE3 at the Tandem 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 OS/390 node to $B.FILERESO.FILE4 at the Tandem node.

```
COND1    PROCESS    PNODE=CD.TANDEM                                -
                    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
```

Index