netCONVERT

Reference Manual

525 Capital Drive
Lake Zurich, IL 60047
Telephone: (800) 228-0255
Fax: (847) 540-5629
Email: tech@wrkgrp.com
http://www.wrkgrp.com

This manual describes the software product netCONVERT, a data conversion program for large companies converting from IBM mainframe to UNIX environments. This manual refers to release 2.20 and subsequent releases of the product.

netCONVERT is also distributed by CoSORT / Innovative Routines International, Inc. and selected agents outside the United States.

Table of Contents

Chapter 1: Product Overview

Over the past decade, the information technology industry has begun to address a number of issues that alter the traditional ways of thinking about applications and data. These include

- an enterprise network includes a variety of computing platforms, from the desktop PC through the Unix workstation to mid-range and mainframe systems; both hardware and software must be fully interoperable in this heterogeneous environment
- data formats and encodings vary among the platforms in a network requiring conversion or translation for use in a distributed application; complex record structures complicate such conversion
- storage of record definitions may vary from one computing platform to another and from one application to another
- the data itself may be stored on various media, further complicating access to it from a given application or platform

The Workstation Group's Network Data Management product line is designed to automate the process of delivering data to applications in a network environment, facilitating interoperability of the distributed components. The netCONVERT product provides the data conversion and translation services required.

netCONVERT Features

netCONVERT accepts data in one format and converts it to another. The conversion may be any of the following

- one mainframe format to a different mainframe format
- mainframe format to workstation or desktop format
- workstation or desktop format to mainframe format
- one workstation or desktop format to a different workstation or desktop format

Appendix C: System Limitations & Platform Dependencies includes a chart of all the conversions supported.

Input may be from disk files, tape files, or other sources piped to stdin. Output may be directed to disk, tape, or piped to other sources through stdout.

Tape I/O includes support for both IBM and ANSI standard labelled tapes as well as for unlabelled tapes. You may specify a volume id, device name and file position for both input and output. For output, you may also specify the file disposition and expiration date.

Regardless of the input or output source, you may specify record length, blocksize, and record format. Mainframe record formats supported include

- Fixed: F, FB, FBS
- Variable: V, VB, VBS

Workstation/desktop record formats supported include

- Fixed
- Variable
- Segmented (variable, but with a segment byte as in ANSI standard)
- Text (line-feed delimited)
- NT (like Text, but carriage-return/line-feed delimited)
- MFCOBOL (Micro Focus COBOL special variable format)
- Fortran
- CSV (comma-separated values)

netCONVERT uses the COBOL File Description (FD) as the basis for data mapping. If a COBOL FD exists for an input file, netCONVERT can use that file for the record map. If a COBOL FD does not exist, the options for defining the data mapping include

- using the **`-record record-mapping`** syntax of the filter program
- converting other types of data mappings to a COBOL FD

For conversions between platform types, netCONVERT includes a standard ASCII/EBCDIC translation table that produces reversible results. You may also define your own translation tables for data or applications where the standard table does not achieve the desired results. Details are provided in ***Appendix D, Defining Translation Tables***.

The following table describes the data conversions performed by netCONVERT.

Data Type	Mainframe Format	Workstation Format
character	EBCDIC	ASCII
numeric	binary integer	binary integer (1)
numeric	IBM370 floating point	IEEE floating point
numeric	packed decimal	packed decimal
numeric	EBCDIC zoned	ASCII zoned (2)
numeric	EBCDIC zoned with leading sign (3)	ASCII zoned with leading sign (3)
numeric	EBCDIC zoned with trailing sign (4)	ASCII zoned with trailing sign (4)

(1) no conversion is performed on binary data except the reversal of byte order required for Intel-based workstations and other platforms that store such data in reverse byte order

(2) Normally an EBCDIC embedded sign is converted to the corresponding ASCII embedded sign, using the ASCII embedded sign convention. When the ALTZONE flag is specified, an EBCDIC embedded sign is instead directly translated to an ASCII equivalent of the EBCDIC sign convention standard.

(3) a workstation format in which the sign is stored in a separate character position preceding the data

(4) a workstation format in which the sign is stored in a separate character position following the data

Using netCONVERT

You use netCONVERT by executing the filter program directly as a utility program from the shell prompt. The filter program is described in detail in ***Chapter 3, Using the Filter Program***. On Unix or Windows NT platforms you may run the filter from the operating system command prompt by specifying appropriate conversion keywords and operands. Alternatively, you may create scripts for standard conversions which may be run from the OS command prompt or as background or batch jobs. On MVS or VM, certain operands of the filter program must be specified as DDnames or FILDEFs. On these systems, the filter program must be executed from a REXX script or a batch JCL stream.

On Unix platforms the filter program may also be used in conjunction with Micro Focus COBOL two-way pipes to allow a COBOL program to use the output of the conversion process directly. Using a two-way pipe in this manner is documented in the ***COBOL System Reference*** from Micro Focus. To use this technique with the netCONVERT filter program, define an environment variable in the form

```
dd_myfile="|netConv netConvert-execution-options"
```

In this case, **myfile** is a two-way pipe to the process in which the filter program is running and all READ operations in your COBOL program read stdout from that process.

Chapter 2: Using the Filter Program

The filter program is the component of netCONVERT that

- reads data from the specified input stream
- converts data according to the mapping provided
- writes the result to the specified output stream

Direct execution of the filter program is typically performed from a script. This allows conversions to be executed as part of larger jobs or to be scheduled to run without user intervention. The section entitled ***Writing netCONVERT Scripts*** later in this chapter provides details on implementing such scripts. In Unix and Windows NT environments, you may also execute the filter program directly from the operating system command prompt.

Appendix C: System Limitations and Platform Dependencies includes a chart of all conversions supported by netCONVERT.

The netConv Command

The netCONVERT filter program is executed by the **netConv** command. Keyword operands are positional and must immediately follow the keyword with which they are associated.

```
netConv   -input   filename [input-options]

          -output  filename [output-options]

         [-record  fd filename   ]
                   record-mapping

         [-number  n]

         [-errors  n]

         [-logfile filename]
```

Keyword	Purpose	Default
-input	specifies the name of the input data stream; also specifies additional *input-options* to further define the input stream For MVS, the input data set is assigned a DDname in a REXX script or batch JCL. ***filename*** is specified as the DDname. For VM, the input file is defined by a FILEDEF in a REXX script. ***filename*** is specified as the FILEDEF name. For Unix or Windows NT, ***filename*** may be any valid file name for that system; if the input stream is to be taken from stdin specify ***filename*** as "-"	
-output	specifies the name of the output data stream; also specifies additional *output-options* to further define the output stream For MVS, the output data set is assigned a DDname in a REXX script or batch JCL. ***filename*** is specified as the DDname. For VM, the output file is defined by a FILEDEF in a REXX script. ***filename*** is specified as the FILEDEF name. For Unix or Windows NT, ***filename*** may be any valid file name for that system; if the output stream is to be directed to stdout, specify ***filename*** as "-"	
-record	provides the name of the COBOL FD file containing the record mapping or defines the record mapping on the command line For MVS, the data set containing the FD is assigned a DDname in a REXX script or batch JCL. ***filename*** is specified as the DDname. For VM, the file containing the FD is defined by a FILEDEF in a REXX script. ***filename*** is specified as the FILEDEF name. For Unix or Windows NT, ***filename*** may be any valid file name for that system. Alternatively, a record mapping may be defined as an argument of the "-record" keyword using ***record-mapping*** options described below. If no record mapping is specified, the only conversion performed is from EBCDIC encoding to ASCII or the converse.	
-number	specifies the number of records to convert	all
-errors	specifies the number of conversion errors to allow	0
-logfile	specifies the name of a data stream to which log messages are directed For MVS, the logfile is assigned a DDname in a REXX script or batch JCL. ***logfile*** is specified as the DDname. For VM, the logfile is defined by a FILEDEF in a REXX script. ***logfile*** is specified as the FILEDEF name. For Unix or Windows NT, ***logfile*** may be any valid file name for that system	stderr
-asis	transfers data from the input data stream to the output data stream with no conversion	n/a

input-options may be any of the following keywords as required to define the input data stream. You may specify one or more of these keywords on the netConv command.

Keyword Option	Purpose	Defaults/Restrictions
volume *volid* [,*volid...*]	Volume serial list	No default; **must** be specified for labelled tape input For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***input-option*** in the execution parameter list, it is ignored
unit *device-name*	Name of the input device	No default; **must** be specified for tape input For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***input-option*** in the execution parameter list, it is ignored
label nl | sl | al	Label type: nl=unlabelled sl=IBM standard label al=ANSI standard label	Used for tape input only If the keyword **recfm** is specified, the default value of **label** is determined from the tape label; the setting is switched, if necessary, to match contents of existing label If the keyword **conv** is specified with option **f**, **v**, or **s**, then default for **label** is **al** For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***input-option*** in the execution parameter list, it is ignored
position *n*	File number	Default=1; used for tape input only For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***input-option*** in the execution parameter list, it is ignored

lrecl ***n***	Record length	For labelled tape input, the default is the value extracted from the tape label For all other input, the default is the value computed from record definition For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***input-option*** in the execution parameter list, it is ignored
blksize ***n***	Block size	For labelled tape input, the default is the value extracted from the tape label For all other input streams, the default is 32760 in the case of **recfm v** or, for all other formats, the largest multiple of **lrecl** that is less than or equal to 32760 For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***input-option*** in the execution parameter list, it is ignored
recfm f | v [b] [s]	Mainframe record format: f=fixed v=variable b=blocked s=spanned	You **must** specify one of **recfm**, **recorg** or **conv**; you may specify only one of these for a given data stream Use **recfm** for non-VSAM, mainframe data Specifying **recfm** implies that the data encoding is EBCDIC
recorg es | ks | rr	VSAM record format: es=entry sequential ks=keyed sequential rr=relative record	You **must** specify one of **recfm**, **recorg** or **conv**; you may specify only one of these for a given data stream Use **recorg** for VSAM data Specifying **recorg** implies that the data encoding is EBCDIC
conv ***conv-options***	Workstation record format; see ***conv-options*** below for a detailed description of each of the workstation formats supported	You **must** specify only one of **recfm**, **recorg** or **conv**; you may specify only one of these for a given data stream Specifying **conv** implies that the data encoding is ASCII

endian big|little	The byte order in which data is stored big=standard byte order little=reverse byte order	The default is the normal byte order of the platform on which netCONVERT is being executed. This is standard byte order (endian big) for most platforms and reverse byte order (endian little) for Intel-based systems.
mfcomp	Enables Micro Focus Byte-Storage Mode; see ***Micro Focus Storage Mode*** below for details	The default is Micro Focus Word Storage Mode, which causes computational (i.e., binary integer) data items to be either two or four bytes in length. When **mfcomp** is specified, such data items may have any length from one to four.
altzone	Enables EBCDIC zoned decimal sign conventions	Signs in USAGE DISPLAY data items without the SIGN IS SEPARATE clause (i.e., zoned decimal data items) contain embedde signs which are normally encoded using an ASCII-based convention. **altzone** causes an EBCDIC-based convention to be used instead.

output-options may be any of the following keywords as required to define the output data stream. You may specify one or more of these keywords on the netConv command.

Keyword Option	Purpose	Defaults/Restrictions
disp new | old | mod	Disposition for tape or disk output **new**=create new file; do not overwrite existing file **old**=overwrite existing file; do not create new file **mod**=append to existing file	If **disp** is omitted and the file does not exist, a new file is created; if the file exists, it is overwritten. For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***output-option*** in the execution parameter list, it is ignored
volume *volid* [,*volid...*]	Volume serial list	No default; **must** be specified for labelled tape output For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***output-option*** in the execution parameter list, it is ignored
unit *device-name*	Name of the output device	No default; **must** be specified for tape output For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***output-option*** in the execution parameter list, it is ignored
label nl | sl | al	Label type: nl=unlabelled sl=IBM standard label al=ANSI standard label	Used for tape output only If the keyword **recfm** is specified, the default value of **label** is determined from the tape label; the setting is switched, if necessary, to match contents of existing label If the keyword **conv** is specified with option **f**, **v**, or **s**, then default for **label** is **al** For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***output-option*** in the execution parameter list, it is ignored

expdt ***yyddd***	Expiration date	Used for tape output only Default is the value extracted from the current output tape file, if it exists, or 000000 For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***output-option*** in the execution parameter list, it is ignored
position ***n***	File number	Default=1; used for tape output only For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***output-option*** in the execution parameter list, it is ignored
lrecl ***n***	Record length	For labelled tape output, the default is the value extracted from the tape label For all other output, the default is the value computed from record definition For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***output-option*** in the execution parameter list, it is ignored
blksize ***n***	Block size	For labelled tape output, the default is the value extracted from the tape label For all other output streams, the default is the value of **lrecl** or, in the case of **recfm v**, the value of **lrecl** plus 4 For MVS or VM, **must** be specified in the REXX script or JCL used to start the filter program; if specified as an ***output-option*** in the execution parameter list, it is ignored
recfm f | v [b] [s]	Mainframe record format: f=fixed v=variable b=blocked s=spanned	You **must** specify one of **recfm**, **recorg** or **conv**; you may specify only one of these for a given data stream Use **recfm** for non-VSAM, mainframe data Specifying **recfm** implies that the data encoding is EBCDIC

recorg es | ks | rr	VSAM record format: es=entry sequential ks=keyed sequential rr=relative record	You **must** specify one of **recfm**, **recorg** or **conv**; you may specify only one of these for a given data stream Use **recorg** for VSAM data Specifying **recorg** implies that the data encoding is EBCDIC
conv *conv-options*	Workstation record format; see ***conv-options*** below for a detailed description of each of the workstation formats supported	You **must** specify only one of **recfm**, **recorg** or **conv**; you may specify only one of these for a given data stream Specifying **conv** implies that the data encoding is ASCII
endian big|little	The byte order in which data is stored big=standard byte order little=reverse byte order	The default is the normal byte order of the platform on which netCONVERT is being executed. This is standard byte order (endian big) for most platforms and reverse byte order (endian little) for Intel-based systems.
mfcomp	Enables Micro Focus Byte-Storage Mode; see ***Micro Focus Storage Mode*** below for details	The default is Micro Focus Word Storage Mode, which causes computational (i.e., binary integer) data items to be either two or four bytes in length. When **mfcomp** is specified, such data items may have any length from one to four.
altzone	Enables EBCDIC zoned decimal sign conventions	Signs in USAGE DISPLAY data items without the SIGN IS SEPARATE clause (i.e., zoned decimal data items) contain embedde signs which are normally encoded using an ASCII-based convention. **altzone** causes an EBCDIC-based convention to be used instead.

conv-options defines the workstation (ASCII) record format and may be specified as any one of the following:

fixed	fixed length records; short records are padded with blanks
variable	variable length records; the length of each record is stored as part of the record; netCONVERT uses the ANSI standard of a 4-byte length prefix in display form variable length records with the addition of a segment byte as defined in the ANSI standard
text	variable length records with trailing blanks suppressed and delimited by a newline character (`x'0a'`); typical of files created in Unix text editors
nt	same as text records but uses a carriage return/newline combination (`x'0d0a'`) as the record delimiter; typical of files created in text editors under PC operating systems
mfcobol	variable length records as implemented by the Micro Focus COBOL compiler
fortran	variable length records with both a length prefix and suffix
csv	comma separated values; a data import/export format supported by many database systems, spreadsheets, and other applications

record-mapping allows you to define the data mapping as part of the netConv command. This may be useful if there is not an existing COBOL File Description for the data to be converted. A record mapping defines the fields in a record in the form **`[type,length [type,length ...]]`**

Within each "type,length" pair, use a comma to separate ***type*** and ***length***. A blank separates the pairs. There must be a "type,length" pair specified for each field in the record.

type may be any of the following:

Type	scription	Length
binary	numeric data in computer-internal format	2, 4, or 8
character	human-readable data	1 to 32760
display	same as **character**	1 to 32760
filler	unprocessed; defines "unused space" within a record description	1 to 32760
float	floating point numeric	4 (for single precision) or 8 (for double precision)
int	same as **binary**	4
leading	signed numeric character data with the sign in a separate character position preceding the data	1 to 18
packed	packed-decimal, a mainframe data type used for exact arithmetic	1 to 16
short	same as binary	2
trailing	signed numeric character data with the sign in a separate character position following the data	1 to 18
zoned	signed numeric character data	1 to 18

Micro Focus Storage Mode

Micro Focus operates in one of two storage modes, which control how Computational (i.e., binary integer) data is stored in memory:

Mode	MFC Compiler Directive	Description
Byte-Storage Mode	NOIBMCOMP	Binary integers may have any length from one to four bytes.
Word Storage Mode	IBMCOMP	Binary integers will only have lengths two or four.

Note that Micro Focus supports binary integer lengths up to eight, but netCONVERT's support for this data type is currently limited to four bytes maximum length.

NOIBMCOMP is the default for Micro Focus Cobol. If you are running with this default, you must specify the netCONVERT **mfcobol** flag for data files associated with such Micro Focus Cobol programs, so that netCONVERT will convert binary integer data items according to the standard Micro Focus Cobol standards. Alternatively, you can use the IBMCOMP directive with your Micro Focus Cobol programs and omit **mfcomp**, and then netCONVERT will convert these data items according to IBM Cobol standards.

The following table describes the resulting field widths for binary integer data items in each case...

Number of Signed Digits	Number of Unsigned Digits	Width in Byte Storage Mode	Width in Word Storage Mode
1-2	1-2	1	2
3-4	3-4	2	2
5-6	5-7	3	4
7-9	8-9	4	4

Return Codes

Return Code	Meaning
0	Normal completion; output file written successfully
1	Program initialization error; no output file written
2	Too many conversion errors; output file was terminated prematurely
3	Program stopped by user (user pressed Ctl-C during an interactive execution or killed a running process for a background or batch job; output file terminated

Writing netCONVERT Scripts

The filter program may also be executed from a script. Scripts may be useful when you need to

- run a standard conversion repeatedly
- run a conversion as a background or batch job
- schedule a conversion to be run automatically by a job scheduler such as “cron”
- run several conversions in sequence
- include the conversion as part of a more complex process that is driven by a script
- run conversions that require a custom translation table

and possibly other situations.

Scripts to execute the filter program may be written in REXX, in any of the Unix shell languages, or in the scripting language of your choice on the platform where you use netCONVERT. The script must be able to set environment variables that are used by the filter program.

The netConv command is, itself, a script that establishes certain required settings and starts the filter program. ***Appendix C: System Limitations & Platform Dependencies*** describes the required settings for each platform type. This section shows the netConv scripts for each platform and discusses how they may be called or adapted for use at your site.

Unix Platforms

A listing of the netConv script for Unix platforms is shown below:

```
#!/bin/sh -f
#
# Copyright (c) 1996-97 The Workstation Group, Ltd. All Rights Reserved.
#
#  netConv - shell script to invoke netCONVERT filter program
#
#
#  Modification history:
#
#  11 Oct 96  pjt  Initial implementation
#
#
```

```
#
# set location of TWG license manager control files
#
LMDIR=/usr/local/twglm

PATH=${LMDIR}:${PATH}
export PATH NCPATH
#
#
# execute the filter program, taking keyword options
# from the command line arguments provided
#
${NCPATH}/Filter "$@"
```

The current setting of LMDIR and NCPATH may differ at your site if you chose to install netCONVERT in a directory other than the default.

To write your own scripts, you may copy and modify netConv or you may write scripts that call netConv. In either case, you **must** ensure that both NCPATH and LMDIR are set correctly. If you are writing a script to call the netConv script, it is not necessary to set these variables in your own script as they will be defined by netConv.

Other things you might do in a script include

- setting the NCXLATE environment variable to use a custom translation table (see ***Appendix D, Defining Translation Tables*** for details)
- defining a Micro Focus COBOL two-way pipe that uses netConv

Windows NT Platforms

A listing of the netConv BAT file for Windows NT platforms is shown below:

```
@echo off

rem

rem Copyright (c) 1996-1997 The Workstation Group, Ltd. All Rights Reserved.

rem

rem  netConv.bat - script to invoke netCONVERT filter program

rem

rem  Modification history:

rem

rem  11 Jan 97  jcs  Initial implementation

rem

rem
```

```
  shift
  goto loop
:end

@echo on
%NCPATH%\Filter %ncparams%
```

To write your own scripts, you may copy and modify netConv.bat or you may write scripts that call netConv.bat. In either case, you **must** ensure that NCPATH is set correctly. If you are writing a script to call netConv.bat, it is not necessary to set this variable in your own script as it will be defined by netConv.

Other things you might do in a script include

- setting the NCXLATE environment variable to use a custom translation table (see ***Appendix D, Defining Translation Tables*** for details)

MVS and VM Systems

The standard distributions for MVS and VM include a REXX script to set required DDnames, FILDEFs or GLOBALs and run the filter program. On MVS, this script is delivered as NETCON.DATA(NETCONV). On VM, it is NETCONV EXEC on the mini-disk where you installed netCONVERT.

```
/* REXX script to invoke netCONVERT filter program                    */
/*
 *
 * NETCONV EXEC  --  invoke netCONVERT filter program
 *
 *
 * NOTE:  You MUST set up definitions for the input and output
 *        files and the COBOL FD file prior to invoking the filter
 *        program.
 *
 *        On VM, you would create an EXEC that sets the necessary
 *        FILEDEFs and then calls NETCONV.
 *
 *        On MVS, you would create a REXX script that defines the
 *        necessary DDnamds and then calls NETCONV.
 *
 *
 *
 *
 * Modification History:
 *
 * 10 Feb 97  pjt  Initial implementation
 *
 *
 */

parse source platform .

parse arg arg_list
parse upper var arg_list . '-INPUT' inp +6 +1 inargs '-' 0,
                           '-OUTPUT' outp +7 +1 outargs '-' 0,
                           '-RECORD' rec +7 +1 fdargs '-' 0,
                           '-LOGFILE' log +8 +1 lgfile '-' .
```

```
/*
 * Some rudimentary validation of input arguments
 */

bad = ''
if inp  ¬= '-INPUT'  then bad = bad 'input'
if outp ¬= '-OUTPUT' then bad = bad 'output'

if words(inargs)  < 1 then bad = bad 'inargs'
if words(outargs) < 1 then bad = bad 'outargs'

if rec = '-RECORD' then do
   IF WORDS(FDARGS)  <  1 THEN BAD = BAD 'FDARGS'
   if words(fdargs)  >= 1 then
      if word(fdargs, 1) = 'FD'
         then if words(fdargs) ¬= 2
            then bad = bad 'fdargs'
            else nop
         else if pos(',', fdargs) = 0 then bad = bad 'fdargs'
   end

if log = '-LOGFILE' then
   if words(lgfile) ¬= 1 then bad = bad 'lgfile'

if bad \= '' then call arg_error

select
   when platform = 'CMS' then do
      'state lscrtl loadlib *'
      if rc ¬= 0 then do
         say ''
         say 'You must have access to the SAS/C run-time library LSCRTL LOADLIB'
         say ''
         say 'LSCRTL LOADLIB not found on any disk currently accessed'
         say 'Exiting now'
         say ''
         exit
         end

      'global loadlib lscrtl'/* SAS/C run-time library */
      'filedef NCKEY disk netcon key *'     /* license activation key */
      'filedef NCMSG disk convert msg *'    /* netCONVERT message file */

      'FILTER' arg_list                      /* run the filter program */
      end

   when platform = 'TSO' then do
      'alloc f(ctrans) da(c600.linklib) shr reuse'
      'alloc f(nckey) da(netcon.data(netcon)) shr reuse'
      'alloc f(ncmsg) da(netcon.data(ncmsg)) shr reuse'

      "call netcon.load(filter) '"arg_list"'" /* run the filter program */
      'free f(ctrans)'
      'free f(nckey)'
      'free f(ncmsg)'
      end

   otherwise
      say ''
      say 'Unrecognized platform name:' platform
```

```
      say 'Exiting'
      say ''
   end

exit

arg_error:
/*
 * Print proper error messages based on argument list errors found
 */

say ''
if wordpos('input', bad)  > 0
   then say '-INPUT keyword missing'
   else if wordpos('inargs', bad)  > 0
      then say '-INPUT arguments missing'

if wordpos('output', bad) > 0
   then say '-OUTPUT keyword missing'
   else if wordpos('outargs', bad) > 0
      then say '-OUTPUT arguments missing'

if wordpos('fdargs', bad)  > 0
   then say '-RECORD arguments missing or incomplete:' fdargs

if wordpos('lgfile', bad)  > 0 then say 'Log file name missing'
say ''
exit
```

Using netCONVERT on mainframe systems requires the setting of appropriate DDnames or FILEDEFs for the input, output and optional FD and log files. Several of the ***input-options*** and ***output-options*** must also be provided as part of the script or JCL that runs the filter program.

Typically, you would set these parameters in a script or JCL stream and call NETCONV, passing it an appropriate argument list for the conversion desired. As an alternative, it is possibly to modify a copy of NETCONV for individual conversions or to convert it entirely to a batch JCL stream if all conversions are to be run as batch jobs.

Examples

The netConv command and keyword arguments are always a single line of input terminated by a carriage return. When typing netConv commands at the operating system prompt, it is not unusual for the command to "wrap" onto a new line before you press Enter. When including netConv commands in a script, the entire command may be on a single line or you may use the standard line continuation conventions of your scripting language to facilitate readability.

For readability in the examples that follow, the netConv command and its arguments may appear on multiple lines.

Example 1: Convert a file on IBM Standard labelled tape to Unix text format. The input parameters are determined from the tape label. The output file contains variable-length records delimited by a newline character (`x'0a'`). The disk file "mvsdata.fd" contains the data mapping.

```
netConv -input mvs.data unit /dev/rmt/0 volser 000001 label sl
        -output unix.data conv text -record fd mvsdata.fd
```

Example 2: Convert an MVS disk file to workstation format. The input file is fixed format with 80-byte records. The output file retains these characteristics. There is no COBOL FD for this file, so the data mapping is provided by a ***record-mapping*** with the **-record** keyword.

```
netConv -input mvs.data recfm f lrecl 80
        -output unix.data conv fixed lrecl 80
        -record char,80
```

Chapter 3: Conversion of Date Fields

netCONVERT provides facilities for converting date fields between two-digit and three- or four-digit representations. The conversion may be run in either direction. This allows you to use the same data with both expanded-date- compliant and non-compliant programs.

netCONVERT supports two types of expanded date formats:

Single-digit century value

The century component is represented implicitly by a 0 or a 1 in the century position. The value is 0 if the date is to be treated as a 20th century date or 1 if the date is to be treated as 21st century.

Two-digit century value

The century component is represented explicitly as a two-digit value. The value is 19 if the date is to be treated as a 20th century date or 20 if the date is to be treated as 21st century.

In both cases, a trigger value for the year determines which century value is chosen. The default value for the trigger year in netCONVERT is 22. Year values less than or equal to 22 are treated as 21st century dates. You may change this as appropriate for each data file converted.

Date field conversions are defined by inserting comment lines in the COBOL File Description for the data to be converted. These comment lines are of the form

```
* nc directive
```

The asterisk is the normal COBOL FD comment delimiter.

nc alerts netCONVERT that this comment line includes a data conversion directive.

directive is composed of a command and its operand(s). Valid date conversion directives are shown in the following table:

Command	Operand(s)	Description
convert	***input-mask output-mask*** Both ***input-mask*** and ***output-mask*** are specified one of the following forms **`[other-data]ci[other-data]year[other-data]`** **`[other-data]year[other-data]ci[other-data]`** The input and output masks describe the data in the field as it appears in its display (PIC) representation. ***ci*** is the century indicator and designates the position within the field where the century value is to be stored. In the mask, ***ci*** is specified as **c** for single-digit century values or as **cc** for two-digit century values. ***year*** indicates the position of the year value within the field. In the mask, it is specified as **yy**. ***other-data*** is any other data in the field. This includes the month and/or day components of the date and any other data that may occur in this field. ***other-data*** may be numeric or character data depending on the field definition. In the mask, ***other-data*** is represented as zero or more **x** characters in the position(s) where such other data is stored in the field. The number of **x** characters in the input mask must match exactly the number in the output mask. The century indicator may be specified in only one of the ***input-mask*** or the ***output-mask***. It may not occur in both. When the century indicator appears in the ***output-mask***, a century value is added to the field during conversion. When the century indicator	Convert one field using the input and output masks provided

convertall	***field-name input-mask output-mask*** ***field-name*** is the simple name of an elementary item in defined in this FD. The input and output masks are specified in the same was as for conversion of a single field.	Convert all fields that match ***field-name*** using the masks provided
noconvert	N/A	Exclude this field from conversion specified by a convertall directive
setslide	***nn*** ***nn*** is a two-digit year value	Override the netCONVERT default value of 22 as the trigger year for single-digit century values

The **convert** and **noconvert** directives apply to individual fields and must be inserted immediately **following** the field definition to which they apply.

The **convertall** directive defines a global conversion for all fields that match the field name specified. You may specify zero or more **convertall** directives. Any **convertall** directives that occur must precede the first field definition in the FD.

The **setslide** is also global and applies to all conversions for this FD. You may not specify more than one **setslide** directive in an FD. The **setslide** directive, if specified, must precede the first field definition in the FD.

Date Conversion Considerations

netCONVERT converts date values in any of the following field types:

- character
- packed decimal
- zoned decimal
- binary

When netCONVERT adds century values to a field, it automatically expands the field as necessary to contain the converted data. In the case of a two-byte binary field, netCONVERT expands the field from two to four bytes as necessary to match the PIC mask. As an example, consider the conversion directive

```
* nc convert yyxx cyyxx
```

Even though the addition of a zero or 1 to the original data would not force expansion to a full word, COBOL requires a four-byte field for a display representation of PIC(99999).

For all other cases, if the converted data exceeds the maximum value that can be stored within the field, a runtime error occurs.

Examples of Data Conversion Directives

*** nc convert xxyyxx xxcyyxx**

Add a single-digit century value to the immediately preceding field using the default trigger year of 22. Store the century value immediately preceding the year value.

*** nc convert yyxxx yyccxxx**

Add a two-digit century value to the immediately preceding field. Store the century value immediately following the year value.

*** nc convert yyxxxx yyxxxxcc**

Add a two-digit century value to the immediately preceding field. The century value is not adjacent to the year value.

*** nc convert ccyyxxx yyxxx**

Remove the two-digit century value from the immediately preceding field.

*** nc convertall myfield2 xxxyy xxxyyc**

Add a single-digit century value to all fields that match the input mask.

*** nc noconvert**

Regardless of any **convertall** directives, do not convert the field immediately preceding this directive.

*** nc setslide 37**

Set the trigger year value for single-digit century values to 37.

Creating an FD for Use with Converted Data

Once the date fields in a data file have been expanded, you need a matching COBOL File Description to use with your application. netCONVERT includes a utility -- **ncfd** -- to create an expanded-date-compliant FD from the input file used for the conversion.

```
ncfd -y2k -input inputFD -output outputFD
```

-y2k specifies that the output FD should be modified to correspond to the netCONVERT date conversion directives in the input file.

-input identifies the input FD file. ***inputFD*** is the name of the input stream.

For MVS, the input data set is assigned a DDname in a REXX script or batch JCL. ***inputFD*** is specified as the DDname.

For VM, the input file is defined by a FILEDEF in a REXX script. ***inputFD*** is specified as the FILEDEF name.

For Unix or Windows NT, ***inputFD*** may be any valid file name for that system; if the input stream is to be taken from stdin specify ***inputFD*** as "-".

-output identifies the output FD file. ***outputFD*** is the name of the output stream.

For MVS, the output data set is assigned a DDname in a REXX script or batch JCL. ***outputFD*** is specified as the DDname.

For VM, the output file is defined by a FILEDEF in a REXX script. ***outputFD*** is specified as the FILEDEF name.

For Unix or Windows NT, ***outputFD*** may be any valid file name for that system; if the input stream is to be taken from stdout specify ***inputFD*** as "-".

Appendix A: netCONVERT COBOL Conventions

netCONVERT accepts any syntax that is valid in a COBOL File Description. It uses only those statements that actually map the data record. It ignores

- comments
- FD statement
- RECORDING MODE statement
- BLOCK statement

Because netCONVERT uses only the File Description (FD), there is no Procedure Division available to provide the logic for mapping redefined records. netCONVERT therefore requires that you insert special COBOL comment statements into the File Description for redefined records to provide this logic. The following rules apply:

- redefinition of a record is dependent on the value of a "trigger field" in the record
- the value of each trigger field used to remap a REDEFINEd record must be specified as a Level 88 item
- for each REDEFINEd record, comment statements must be added to specify both the default record definition and all possible re-definitions. These statements are placed AFTER the redefines block. The example will illustrate this usage.

Comment Statement	Meaning
NC WHEN *item-name	specifies redefinition when the trigger field ***item-name*** has the value specified on the Level 88 statement
***NC OTHERWISE**	specifies the default record definition

Here is an example of an FD that has been modified to provide the redefinition logic required by netCONVERT:

```
 FD  SYSIN1
     BLOCK CONTAINS 0 RECORDS
     RECORDING MODE IS V.

 01  SYSIN1-REC.
     05 PATIENT-NO           PIC 9(8)  COMP SYNC.
     05 PATIENT-REC-ID       PIC 9(6)  COMP SYNC.
        88 PNOTIFY           VALUE 1.
        88 PTESTS            VALUE 2.
        88 PDOC              VALUE 3.
*
     05 PATIENT-AREA         PIC X(200) OTHERWISE.
*NC  OTHERWISE

     05 PATIENT-NOTIFICATION REDEFINES PATIENT-AREA.
        10 PATIENT-LFMNAME  PIC X(40).
        10 PATIENT-LFMADD   PIC X(60).
        10 PATIENT-NOKNAME  PIC X(40).
        10 PATIENT-NOKADD   PIC X(60).
*NC  WHEN PNOTIFY

     05 PATIENT-TESTS REDEFINES PATIENT-AREA.
        10 PATIENT-SMAC20   OCCURS 40 TIMES
           PIC 9(3) COMP-3.
        10 PATIENT-CBC      OCCURS 60 TIMES
           PIC 9(3) COMP-3.
*NC  WHEN PTESTS

     05 PATIENT-DOC-NOTES REDEFINES PATIENT-AREA.
        10 PATIENT-TIMESTAMP PIC X(16).
        10 PATIENT-OBSERVER  PIC X(40).
        10 PATIENT-COND      PIC X(144).
*NC  WHEN PDOC
```

Appendix B: Diagnostic Messages

Message Number	Message Text Explanation
001	**No Message File Available** On Unix and Windows NT, netCONVERT locates the message file (Convert.msg) by the current definition of the NCPATH environment variable. This variable must be set to the name of the directory containing this file, usually the directory where netCONVERT is installed. Check the setting of NCPATH in the script that you are using: “nc” for the graphical tool; “netConv” or your own custom script for the filter program. On MVS, netCONVERT locates the message file by the current definition of the DDname NCMSG. This DDname must be defined to point to the location of a valid message file (default location: NETCON.DATA(NCMSG)). Check the setting of NCMSG in the NETCONV script or your own custom script or JCL for running the filter program. ON VM, netCONVERT locates the message file by the current setting of the FILEDEF NCMSG. This FILEDEF must be defined to point to the location of a valid message file, usually CONVERT MSG *. Check the setting of NCMSG in the NETCONV EXEC or in your own custom EXEC for running the filter program.
002	**Message Open Error** Unable to open a valid netCONVERT message file. The file was found, but is it corrupted in some manner. On Unix and Windows NT systems, the file is Convert.msg and is found in the location specified by NCPATH in the script you are using to run netCONVERT. On MVS or VM systems, the file is identified by the DDname or FILEDEF for NCMSG in the script you are using to run netCONVERF. Contact The Workstation Group or your authorized distributor for assistance if you are unable to resolve this problem.

003	**Logfile Open Error** netCONVERT is unable to write to the logfile you specified. The conversion proceeds, but no messages are logged. On Unix systems, one of the following may have occurred: • the file exists and you do not have write permission • the file is new and you do not have write permission for the specified directory • a fully qualified path name to the file was specified and that file system is not mounted Check file and directory permissions before attempting to use this logfile name again. On Windows NT systems, file attributes may have been set in such a way that you do not have write permission for the file. Other network security issues may prevent you from writing to the file. On MVS, you may have allocated a dataset for which you do not have write permission. Check the RACF, ACF2 or other security rules associated with the dataset. On VM, you may have specified a file on a mini-disk that is not accessed in write mode or you may have specified a file on a shared file system for which you have not been granted write authority.
004	**No Message File Available, check NCPATH setting** **Messages will only have error numbers** The netCONVERT graphical tool locates the message file by the current definition of the NCPATH environment variable. This variable must be set to the name of the directory where netCONVERT was installed. Check the setting of NCPATH in the "nc" startup script (or ncStartup.bat for Windows NT systems. This is only a warning message; the graphical tool will run normally but messages will be reported only by message number (without text).
101	**File Read Error** Unable to read the specified File Description * (FD) file. This may occur if you do not have read permission for the file or if some other system error occurs. Verify that you have read permission for the file and its parent directory(s) and that its file system is mounted and re-run the conversion. If this error is reliably reproducible, contact The Workstation Group or your authorized distributor for assistance.
102	**File Write Error** Unable to write the output File Description (FD) file. This may occur if you do not have write permission for the file or if some other system error occurs. Verify that you have write permission for the file and its parent directory(s) and that its file system is mounted and re-run the conversion. If this error is reliably reproducible, contact The Workstation Group or your authorized distributor for assistance.
103	**No Memory Error** Insufficient real or virtual memory to complete the conversion. Verify that you have adequate swap space allocated on your system. It may not be possible to perform complex conversions on systems with very small amounts of real memory.

105	**No Enclosing Group** The File Description (FD) contains a definition of an elementary item with a level number greater than the immediately preceding statement; and the immediately preceding statement is also an elementary item. Thus there is no group defined for the statement with the higher level number. Correct the file description before re-running the conversion.
106	**No Error Blocks Available** An internal error has occurred in netCONVERT error handling. This should not occur. Please contact The Workstation Group or your authorized distributor for assistance.
107	**Invalid Redefine Target** The trigger field specified for a redefined record is invalid. Either it is not a field in the current group or some other COBOL syntax error has been encountered.
108	**Duplicate Item Name** This item name is a duplicate of another item that has the same name and the same parent item.
109	**Invalid Level Number** A level number is invalid if it fails to meet the following criteria: • must be in the range 1-49 or 88 • must be greater than or equal to the level number of the preceding item definition • if level 88, it must be specified within an elementary item
110	**Invalid Parameter For This Level Number** The following conditions are not valid for this level number: • level 88 item may not be redefined • level 88 item may not be the target of an OCCURS specification • SYNC, SIGN, USAGE, and PIC are not valid with a level 88 item • VALUE may be specified only for an elementary item
111	**Invalid VALUE Type** The value specified for a level 88 item must be of the same type as the elementary item it references.
112	**Invalid Lex Input** This message should not occur. Contact The Workstation Group or your authorized distributor for assistance.
115	**Value Magnitude too large for format** Error in conversion of floating point data
116	**Value Magnitude too small for format** Error in conversion of floating point data.

117	**Value Range Error** The value of the current data will not fit in the field specification.
121	**Yacc Generated Error (Init)** Internal error. Contact The Workstation Group or your authorized distributor for assistance.
122	**Yacc Generated Error (Stack Overflow)** Internal error. Contact The Workstation Group or your authorized distributor for assistance.
123	**Yacc Generated Error (Syntax)**
124	**Yacc Generated Error (Generic)** Internal error. Contact The Workstation Group or your authorized distributor for assistance.
125	**Item Name Not Found** The trigger field specified for a redefined record is not defined as a Level 88 item in this FD.
126	**Item Name Ambiguous** The field name specified as the trigger field for a redefined record cannot be determined unambiguously. It is likely that the same field name exists within more than one block in this FD. You must qualify the trigger field name more completely as in ***t-field* IN *group***
127	**Missing Pic string** The PIC string (display specification) for this field is missing. Correct the FD.
128	**Expected subitem not encountered** The sequencing of levels of groups and elementary items (fields) is incorrect. Correct the COBOL syntax in the FD.
129	**Elementary item has subitem** An elementary item (field) may not have a sub-item specified. Correct the FD specification.
130	**Internal error: no parent** This error should not occur. Contact The Workstation Group or your authorized distributor for assistance.
131	**Internal error: no usage determined** This error should not occur. Contact The Workstation Group or your authorized distributor for assistance.
132	**Invalid OCCURS count value specified** The value to indicate how many times a field occurs within a record is invalid. It must be an integer greater than zero.
133	**Invalid OCCURS DEPENDING target** The field specified in an OCCURS DEPENDING clause of not of integer type.

134	**FILLER must be elementary item** The item specified for FILLER must be an elementary item (field). It may not be a group. Correct the FD specification.
135	**Invalid reference to a FILLER item**
136	**Only one item per set may have OTHERWISE** OTHERWISE is part of the netCONVERT special syntax to provide the logic for REDEFINEd records. The OTHERWISE keyword denotes the default mapping for this record. Only one default may be specified.
137	**Unsupported USAGE operand** The keyword following "USAGE IS" is invalid. Correct the COBOL syntax in the FD.
138	**Typedef feature not supported** netCONVERT does not support the TYPEDEF COBOL feature at this time.
139	**WHEN argument must be a level 88 item** The argument following a WHEN keyword (special netCONVERT syntax for REDEFINEd records) must have previously been defined as a Level 88 item.
140	**File Access Error** An error occurred trying to access the file containing the COBOL FD. For Unix or Windows NT systems, check the permissions on the file and on the directory in which it is located. For MVS or VM systems, check that the DDname or FILEDEF specification is correct and that you have proper security privileges to access the file.
141	**File Open Error** An error occurred attempting to open the file containing the COBOL FD. Check to see that you have specified the correct file and that it is not damaged in any way.
142	**File Close Error** An error occurred attempting to close the file containing the COBOL FD. It is possible that the file may be damaged.
150	**Invalid OCCURS DEPENDING value**
151	**Buffer Bounds Error** A internal conversion error has occurred. Contact The Workstation Group or your authorized distributor for assistance.
152	**Floating Point Overflow**
153	**Floating Point Underflow**
155	**A member of a redefinition set could not be chosen** The current value of the trigger field for redefined records does not match any of the WHEN or OTHERWISE specifications. The record cannot be converted.
156	**Invalid packed decimal data encountered**

157	**Invalid zoned data encountered**
158	**Century conversion encountered invalid year** This can occur for zoned or character fields when the year portion of a date to be converted is not decimal.
165	**Previous item specified Y2K mask inconsistent with PIC spec** The digit count in a Y2K conversion mask is not consistent with the digit count in the field being converted.
166	**Previous item's Y2K mask implies invalid output format** Output integer size exceeds 9 digits
167	**Print operation overflowed buffer size** Internal processing error, please report to vendor tech support.
168	**Internal inconsistency, please report to tech support** Internal processing error, please report to vendor tech support.
169	**Invalid digit count for numeric fields implied in PIC string** Numeric fields are limited to no more than 18 digits.
170	**Unsupported integer usage** Either a PIC digit count or the result of Y2K conversion would create an integer whose decimal value would have more than nine digits.
157	**Invalid zoned data encountered**
301	**You must specify an input file** You must specify the name of a file that contains data to be converted.
302	**You must specify an output file** You must specify the name of a file to which converted data is to be written.
305	**You must specify an input device** When the input data stream is from tape, you must specify a tape device name.
306	**You must specify an output device** When the output data stream is tape, you must specify a tape device name.
307	**You must specify a Record FD** You must specify the name of a file containing the COBOL File Description for the record to be converted.
308	**You must specify a Record Mapping** Because you have not specified an FD for this conversion, you must define a record mapping.

310	**The file does not exist** The file name specified for input data or File Description does not exist in the specified directory. In the “Load Data” dialog or the “Load FD” dialog, you specified a filename or file path that does not exist. In the “Load Data” dialog you may activate a file selection dialog by clicking on the button labelled “...” to the right of the Filename text box.
311	**The file is not readable** You do not have read permission for the file name specified for input data. In the “Load Data” dialog, you specified a filename or file path for which you do not have read access. Check permissions on the file.
312	**You must specify a file** In the “Load Data” dialog, you may not leave the filename blank. You must specify the name or full path of a file that contains input data. You may type the name in the text box or click on the button labelled “...” for a file selection dialog.
313	**You must specify a tape filename** When the input or output data stream is tape, you must specify a filename on that tape.
314	**You must specify a tape volser** For input or output to a labelled tape, you must specify the tape volser.
315	**You must specify a tape unit** For input or output to tape, you must specify the tape device name.
316	**You must specify a tape position** For input or output to tape, you must specify the number of the file to be read or written.
317	**You must specify a Lrecl** If you have not loaded an FD before loading input data, you must specify an lrecl for the input data.
318	**Lrecl must be 0** You must specify record length as a whole number greater than zero.
320	**Could not open logfile** In the “Run Conversion” dialog, you specified as the logfile name a file for which you do not have write permission. The conversion proceeds, but log messages are not captured. Check the permissions on the filename you specified. If the file does not already exist, be sure you have write permission for the directory in which the file is to be created.
321	**An I/O error occurred while reading the FD** An unexpected error occurred when attempting to read the file description (FD) file. This may be caused by a transient disk error or other temporary condition on your system. Re-run the conversion. If this error is reliably reproducible, contact The Workstation Group or your authorized distributor for assistance.

322	**An I/O error occurred while writing the FD** An unexpected error occurred when attempting to write the file description (FD) file. This may be caused by a transient disk error or other temporary condition on your system. It may also occur if the file system in which the output file resides becomes full. Check your file system status and re-run the conversion. If this error is reliably reproducible, contact The Workstation Group or your authorized distributor for assistance.
330	**No more records, End-of-file** Encountered "End-of-file" in the input data stream; no more records can be converted.
331	**The file is not writeable** You do not have write permission for the file name specified for output or logfile. Correct the permissions or specify a different file name.
332	**Year field needs 2 digits** When specifying an expiration date for a file, you must specify 2 digits for the "year" component.
333	**Day field must be 1-366** When specifying an expiration date for a file, you must specify the day of the year as a number between 1 and 366.
340	**recordClose() failed** An unexpected internal error occurred in recordClose(). Please contact The Workstation Group or your authorized distributor for assistance.
341	**Could not exec Filter program** The netCONVERT graphical tool was unable to access the filter program to run the conversion. Check your installation -- the directory in which you installed netCONVERT must contain a file named "Filter". This file must be an executable binary appropriate for your platform and must have read and execute permission for all users. Verify that you can run the filter program directly using the "netConv" script (or netConv.bat on Windows NT). If you cannot resolve this problem, contact The Workstation Group or your authorized distributor for assistance.
342	**Environment variable "NCPATH" not set, Filter program may not be locatable** The netCONVERT graphical tool was unable to access the filter program to run the conversion. This program is normally located using the NCPATH environment variable. If this variable is not set, the graphical tool may not be able to locate the filter program. Check your "nc" script (or "ncStartup.bat" on Windows NT) to be sure that NCPATH is set to the directory where you installed netCONVERT.
343	**Could not find Filter program** The environment NCPATH is set, but no executable program named Filter was found in the directory to which NCPATH points. Either the file does not exist or it does not have execute permission for all users.

401	**Execution Starting Input:*filename* Output:*filename*** Beginning execution of the netCONVERT filter program; input and output data streams for this conversion are as indicated.
402	**Signal(2) Error:** The signal system call failed. This should not occur. Please contact The Workstation Group or your authorized distributor for assistance.
404	**Input Open Failed:** Unable to open input data stream; check file permissions or state of input device
405	**Output Open Failed:** Unable to open output data stream; check file permissions or state of output device
406	**Conversion Init Error** The **-record** keyword was specified but the record definition compiler could not process it. Preceding messages indicate the specific error.
408	**Input Read Error:** Unable to read from input data stream; check file permissions or state of input device
409	**Output Write Error:** Unable to write to output data stream; check file permissions, state of output device, or write-protect mechanism on tape
419	**Data Conversion Error Limit Exceeded** The limit for allowable conversion errors was exceeded. Preceding messages indicate the specific errors encountered. By default, no errors are allowed; the conversion stops when the first error is encountered. Use the **-errors** keyword to specify the number of conversions allowed.
420	**Invalid DDNAME Specification**
421	**Invalid record length on write**
459	**Keyword *keyword* value invalid** The value specifed for ***keyword*** is not valid; see ***Chapter 4, Using the Filter Program*** for valid values for this keyword.
460	**Keyword *keyword* invalid** The keyword specified is not a valid keyword argument for the filter program; see ***Chapter 4, Using the Filter Program*** for valid syntax.
461	**Keyword *keyword* value must be numeric** The value specified in conjunction with this keyword must be numeric; see ***Chapter 4, Using the Filter Program*** for valid syntax.
462	**Value missing for keyword *keyword*** You **must** specify a value for the keyword indicated; see ***Chapter 4, Using the Filter Program*** for valid syntax.

463	**Disp must be NEW|OLD|MOD** The value specified with the **disp** keyword is invalid; must be one of "new", "old", or "mod".
464	**Recfm *format* invalid** The value specified with the **recfm** keyword is invalid; must be one of "f", "fb", "fbs", "v", "vb", or "vbs".
465	**Label must be SL|AL|NL** The value specified with the **label** keyword is invalid; must be one of "sl", "al", or "nl".
466	**Conv value *format* invalid** The value specified with the **conv** keyword is invalid; must be one of "fixed", "variable", "segmented", "text", "nt", "mfcobol", "fortran", or "csv".
467	**Recfm and Conv options mutually exclusive** You may not specify both **recfm** and **conv** for the same data stream. Use **recfm** for mainframe data (EBCDIC encoding); use **conv** for workstation data (ASCII encoding).
468	**One of Recfm, Conv must be specified** You **must** indicate the type of data for both the input and output data streams. Use **recfm** for mainframe data (EBCDIC encoding); use **conv** for workstation data (ASCII encoding).
470	**Endian must be BIG|LITTLE** The value specified with the **endian** keyword is invalid; must be one of "big" or "little". "big" indicates standard byte order used on most hardware platforms. "little" indicates reverse byte order used on the Intel architecture.
471	**Could not acquire buffer:** The malloc failed for an internal buffer. This should not occur. Please contact The Workstation Group or your authorized distributor for assistance.
472	**Invalid field type *type*** The field type specified in the FD file or in the ***record-mapping*** specification is invalid.
473	**Invalid field length *length* for type *type*** In the FD file or the ***record-mapping*** specification, the length is invalid for the data type specified.
479	**Argument processing terminated by "-"**
480	**Error occurred on record number: *n*** A conversion error occured on the specified record.
491	***n* records read** The specified number of records were read from the input data stream.

492	***n* bytes read** The specified number of bytes of data were read from the input file.
493	***n* records written** The specified number of records were written to the output data stream.
494	***n* bytes written** The specified number of bytes of data were written to the output file.
495	***n* conversion errors** The specified number of error occurred during this conversion.
498	***n* records converted, *n* errors** The specified number of records were converted, with the number of conversion errors shown.
503	**Open Failed, File Missing** Unable to open the input file or the output file with disp=old because the file could not be found. Verify that the file exists and that its name is spelled correctly in the **netConv** command arguments. For some platforms, file names are case sensitive.
504	**Open Failed, File Missing**
505	**Open Failed, File not writeable**
508	**Record Read System Call error**
509	**Record Read Logic error**
513	**Data Error in Input File** A file with variable-length record format contained a record with an invalid length specification. Either the internal record length exceed the LRECL for the file or a segment of a SPANNED format file is missing.
514	**Data Error, Read Incorrect Length**
530	**Open failed, RECFM not F or V** The **conv** keyword was not specified and the record format specified was not "F" or "V". This can only occur if a tape label has incorrect information (as for a bad tape label).
531	**Open failed, blksize not multiple of lrecl** Could not open input or output data stream; the block size specified is not a multiple of the record length. This applies only to files with fixed record format.
532	**Open failed, for recfm f, lrecl must equal blksize** Could not open input or output data stream; the block size specified is not the same as the record length. This applies only to files with fixed record format that are not blocked.
533	**Open failed, blksize too small for lrecl** Could not open input or output data stream; for files with variable record length, the blocksize must be at least 4 bytes greater than the record length.

534	**Open failed, DISP not valid for input** The keyword **disp** is not valid in conjunction with the keyword **-input**; see ***Chapter 4, Using the Filter Program*** for valid syntax.
535	**Open failed, UNIT not valid for this environment** The keyword **unit** is not valid on the platform on which you are running netCONVERT.
540	**Tape volume mounted does not match volume requested** The volume id (***volid***) of the tape currently mounted does not match the value specified with the **volume** keyword; mount the correct tape.
541	**Tape label type mounted does not match type requested** The label of the currently mounted tape does not match the value specified with the **label** keyword; mount the correct tape.
542	**Tape dsname mounted does not match dsname requested** The file name specified could not be found on the currently mounted tape; mount the correct tape.
543	**Tape Error: Volume label write failed** Unable to write volume label. Check the write protect mechanism on the tape.
544	**Tape Error: Volume not expired** The current volume has not yet expired; unable to write output.
545	**Tape Error: File label not present** The currently mounted tape is mounted for input and has a volume label but no file label (HDR1/HDR2). The tape is bad and cannot be used.
546	**Tape Error: EOF label error** An error occurred reading the tape End-of-File label. The tape is not written correctly or has a media error.
549	**Tape Error: Rewind failed** An error occurred rewinding the tape. The tape may be written correctly but it is not rewound and may have a media error.
550	**Close Failed:** The close call failed for the input or output device. Preceding messages indicate the specific problem.
570	**Open failed, Input lrecl missing** Unable to open input data stream; record length not specified and could not be determined from the data mapping
580	**MFCOBOL file read error**
581	**MFCOBOL file read error**
582	**MFCOBOL file header format rror**
583	**MFCOBOL file compression not supported**
585	**MFCOBOL file writing header**

701	**Demo license obtained.** You are running the demo version of netCONVERT, which will convert only 10 records. Contact The Workstation Group or your authorized distributor for purchasing information.
702	**License expires in *n* days.** Your license to run netCONVERT will expire in the number of days indicated.
790	**Internal error, terminating.** An internal error occurred in obtaining a license for netCONVERT. Contact The Workstation Group or your authorized distributor for assistance.
791	**Software version not licensed. Contact vendor.** You netCONVERT authorization code is not valid for this release. Contact The Workstation Group or your authorized distributor for assistance.
792	**License key file not found or not readable.** A valid activation key for netCONVERT could not be located. Check your installation. If further assistance is required, contact The Workstation Group or your authorized distributor.
793	**Storage not available to read license key.** There is insufficient memory available on your system to read the license activation key. You may need to increase the swap space on your system or terminate some other processes prior to running netCONVERT.
794	**Read error on license key file.** netCONVERT could not read the file containing the license activation key. Be sure that it has read permission for all users.
795	**Error decoding license key.** netCONVERT was unable to decode your license activation key. The key may be invalid. Contact The Workstation Group or your authorized distributor for assistance.
796	**License for wrong product.** netCONVERT found a valid activation key file but the key it contains is not correct for this product. Contact The Workstation Group or your authorized distributor for assistance.
797	**Hostname not found.** netCONVERT is licensed to run on a specific host in your network. This hostname could not be found on your network. Check name resolution on your network and on the host where you are running netCONVERT. For further assistance, contact The Workstation Group or your authorized distributor.

798	**License for wrong host.** netCONVERT is licensed to run on a specified host. You are attempting to run it on a different host. Contact The Workstation Group or your authorized distributor if you need to change the license to a different host or if you need to acquire licenses for additional hosts.
799	**License has expired.** Your license to run netCONVERT has expired. Contact The Workstation Group or your authorized distributor for purchasing information.

Appendix C: System Limitations & Platform Dependencies

This appendix describes any limitations or restrictions enforced by netCONVERT on the data to be converted. It also describes platform-specific requirements or limitations in the use of the product. Whenever possible, system dependencies have been accommodated in the netCONVERT program itself or in the scripts that run its components.

System Limitations

Converting data with netCONVERT is subject to the following limitations or conventions:

Maximum length of an input record	32760
CSV Format	Supported for output only
CSV Output	Strings enclosed in double quotes; numbers not enclosed in quotes; records truncated to the length of the actual data
Maximum length of a PIC string	30
Maximum length of a symbol name	30
Maximum field size (`PIC X(n)`)	250
Maximum COBOL data level	49 (Level 88 items also permitted; see ***Appendix A: netCONVERT COBOL Conventions*** for use of level 88 items by netCONVERT)

Conversions Supported

	Output												
Input	**recfm**						**conv**						
		f					fixed	variable		text			fortran	csv
	f	s	s		s	s	s	s	s	s	s	s	s	s
	fb	s	s		s	s	s	s	s	s	s	s	s	s
	v	s	s		s	s	s	s	s	s	s	s	s	s
	vb	s	s		s	s	s	s	s	s	s	s	s	s
	vbs	s	s		s	s	s	s	s	s	s	s	s	s
	conv													
	fixed	s	s		s	s	s	s	s	s	s	s	s	s
	variable	s	s		s	s	s	s	s	s	s	s	s	s
	seg	s	s		s	s	s	s	s	s	s	s	s	s
	text	s	s		s	s	s	s	s	s	s		s	
	nt	s	s		s	s	s	s	s	s	s		s	
		s	s		s	s	s	s	s			s	s	s
	fortran	s	s		s	s	s	s	s	s	s	s	s	s
	csv													

Required Settings

The settings shown in the table below are used to locate various components of the netCONVERT system. Failure to set one or more of them will cause execution of **nc**, **ncStartup.bat** or **netConv** to fail.

For Unix or Windows NT systems, appropriate values are set through the definition of environment variables. For MVS, appropriate values are set through the definition of DDnames in a REXX script or JCL stream. For VM, appropriate values are set through FILEDEFs in a REXX script.

Setting	Purpose	Comments
CLASSPATH	Location of the netCONVERT Java classes (normally the directory where netCONVERT is installed); required to execute the Graphical tool	Required only for platforms or sites where Java is installed
CTRANS	Location of the SAS/C run-time library	MVS only
JAVADIR	Location where the Java virtual machine is installed; required to execute the netCONVERT Graphical Tool	Required only for platforms or sites where Java is installed
LD_LIBRARY_PATH	Location of netCONVERT shared libraries and required shared system libraries	Solaris 2 systems only

LIBPATH	Location of netCONVERT shared libraries and required shared system libraries	RS/6000 systems only
LMDIR	Location of License Manager files	Unix platforms only
NCFILTER	Full path and file name of the netCONVERT filter program; required to use the Graphical Tool	Required only on Unix systems where Java is installed
NCKEY	Location of License Manager files	MVS or VM only
NCMSG	Location of the netCONVERT message file	MVS or VM only
NCPATH	Location of the netCONVERT message file and Java classes; for Windows NT, also identifies the location of License Manager files	Unix or Windows NT platforms only

Platform Dependencies

For portability across a wide variety of windowing systems, the graphical tool is implemented in Java. Use of this tool is therefore limited to operating system releases that support the Java virtual machine. We require Java release 1.1 or higher, and recommend release 1.1.3 or higher when running on Windows/NT.

The filter program is implemented in ANSI standard C. Whenever possible, it is compiled on the earliest version of an operating system that is binary-compatible with all current versions.

HP/9000 (HP-UX)

Filter program	Compatible with HP-UX 9.x or HP-UX 10.x
Graphical tool	Built under HP-UX 10.20 Compatible with HP-UX 10.10 or 10.20
Java Download Location	http://www.hp.com/gsyinternet/hpjdk/prel112.html

IBM RS/6000 (AIX)

Filter program	Compatible with AIX 3.2.5 or AIX 4.x
Graphical tool	Built under AIX 4.1.5 Compatible with AIX 4.1.1 or later
Java Download Location	http://ncc.hursley.ibm.com/javainfo/latest/answers/faq0.html

Sun Solaris 2

Filter program	Compatible with Solaris 2.x
Graphical tool	Built under Solaris 2.5 Compatible with Solaris 2.4 or later
Java Download Location	http://www.javasoft.com/products/jdk/1.1/index.html

Microsoft Windows

Filter program	Compatible with 95 and NT 3.51 or later
Graphical tool	Built under NT 4.0 Compatible with 95 and NT 4.0 or later
Java Download Location	http://www.javasoft.com/products/jdk/1.1/index.html

Appendix D: Defining Translation Tables

netCONVERT provides a standard translation table to convert data encoding from EBCDIC to ASCII or the converse. This table provides fully reversible results and should be sufficient for most conversions.

In the event that you require a different translation, you may define your own translation table. A translation table is an ordinary file of the following form:

```
    x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 xA xB xC xD xE xF

0x  00 01 02 03 37 2d 2e 2f 16 05 25 0b 0c 0d 0e 0f
1x  10 11 12 13 3c 3d 32 26 18 19 3f 27 1c 1d 1e 1f
2x  40 5a 7f 7b 5b 6c 50 7d 4d 5d 5c 4e 6b 60 4b 61
3x  f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 7a 5e 4c 7e 6e 6f
4x  7c c1 c2 c3 c4 c5 c6 c7 c8 c9 d1 d2 d3 d4 d5 d6
5x  d7 d8 d9 e2 e3 e4 e5 e6 e7 e8 e9 ad e0 bd 9a 6d
6x  79 81 82 83 84 85 86 87 88 89 91 92 93 94 95 96
7x  97 98 99 a2 a3 a4 a5 a6 a7 a8 a9 c0 4f d0 5f 07
8x  20 21 22 23 24 15 06 17 28 29 2a 2b 2c 09 0a 1b
9x  30 31 1a 33 34 35 36 08 38 39 3a 3b 04 14 3e e1
Ax  41 42 43 44 45 46 47 48 49 51 52 53 54 55 56 57
Bx  58 59 62 63 64 65 66 67 68 69 70 71 72 73 74 75
Cx  76 77 78 80 8a 8b 8c 8d 8e 8f 90 6a 9b 9c 9d 9e
Dx  9f a0 aa ab ac 4a ae af b0 b1 b2 b3 b4 b5 b6 b7
Ex  b8 b9 ba bb bc a1 be bf ca cb cc cd ce cf da db
Fx  dc dd de df ea eb ec ed ee ef fa fb fc fd fe ff
```

The topmost row and leftmost column are coordinates that represent the hexadecimal value of an ASCII code point. The hexadecimal values in the grid represent the EBCDIC equivalent code point for that ASCII value. For example, the ASCII encoding for the digit “2” is represented as x`32’. To locate the corresponding EBCDIC encoding, find the intersection of the row identified by “3x” and the column identified by “x2”. This gives you the value “f2”, which is the EBCDIC hexadecimal representation of the digit “2”.

Customized translation tables should be constructed in this form. You may store your translation table using any filename and in any location that is meaningful to you.

netCONVERT locates custom translation tables by the current value of NCXLATE. For Unix or Windows NT systems, set the environment variable NCXLATE to the full path name of the file containing the translation table. For MVS systems, define the DDname NCXLATE to point to the data set containing the translation table. For VM systems, define a FILEDEF for NCXLATE to point to the file containing the translation table.

If NCXLATE is unset, netCONVERT uses the default translation shown above.