Migration Specialties International, Inc.

217 West 2nd Street, Florence, CO   81226-1403

+1 719-784-9196

E-mail: *Info@MigrationSpecialties.com*

www.MigrationSpecialties.com

CVTFILE UTILITY
USER'S GUIDE

Version 1.1

The following are trademarks of Compaq Corporation:

DEC	DECwriter	Professional	VAX	DECmate	DIBOL
Rainbow	VMS	DECnet	MASSBUS	RSTS	VT
DECsystem-10	PDP	RSX	Work Processor		
DECSYSTEM-20	P/OS	UNIBUS	[digital logo]	DECUS	
Alpha AXP	OpenVMS				

TABLE OF CONTENTS

FIGURES

TABLES

1.   INSTALLING THE CVTFILE UTILITY

This document describes the automated procedure for installing the CVTFILE Utility on the OpenVMS operating system. The procedure requires a minimum of 1000 blocks of free disk space and a OpenVMS operating system running at Version 6.1 or higher.

1.1   VMSINSTAL

VMSINSTAL should be used to install this product. To install using VMSINSTAL, the user should do the following:

1) Log into the system manager's account. Use a hard-copy device, if available, since you may need to examine the listing for errors.

2) Invoke VMSINSTAL by typing:

```
$ @SYS$UPDATE:VMSINSTAL CVTFILvvu
```

where *vvu* is the version number of the kit being installed. For example, version 1.1 of the CVTFILE Utility would be represented as CVTFIL011.

If your system has DECnet and it is running, VMSINSTAL displays the following message:

```
%VMSINSTAL-W-DECNET, Your DECnet network is up and running
```

VMSINSTAL will tell you if there are any other active processes by displaying the following:

```
%VMSINSTAL-W-ACTIVE, The following processes are still active:…
```

VMSINSTAL will then ask:

```
Do you want to continue anyway [NO]?
```

Enter YES if you want to continue; otherwise, VMSINSTAL will stop the installation process.

3) VMSINSTAL will then ask:

```
Are you satisfied with the backup of your system disk [YES]?
```

If you haven't made a fresh backup before the beginning of the installation, you should type NO to this question, which will terminate the installation and allow you to do a backup. Otherwise, press the <RETURN> key to continue.

4) VMSINSTAL will then ask:

```
Enter installation options you wish to use (none):
```

There are no installation options for the CVTFILE Utility kit. Simply press the <RETURN> key and continue.

5) VMSINSTAL will then ask:

```
Where will the distribution volumes be mounted?
```

Enter the name of the device on which you will mount the distribution media (Example: MUA0:).

6) Before VMSINSTAL tries to mount the distribution medium, it asks you to mount the medium on the device that you specified when you responded to the device prompt. The prompt is:

```
Please mount the first volume of the set on ddcu:
```

where *ddcu:* is the device that you specified.

This message is followed by the prompt asking if you are ready to proceed with the installation. When you have mounted the medium, respond by typing YES to the prompt:

If VMSINSTAL succeeds in mounting this volume, it displays the following message:

```
%MOUNT-I-MOUNTED, CVTFIL mounted on _ddcu:

The following products will be processed:

CVTFIL V1.1
Beginning installation of CVTFIL V1.1 at HH:MM

%VMSINSTAL-I-RESTORE, Restoring product saveset A...
```

VMSINSTAL then copies the distribution kit files from the first volume of the distribution medium.

7) The installation procedure asks if you want to purge all the CVTFILE Utility files when the installation is completed. Unless you have a special reason for wanting to preserve the old CVTFILE Utility files on the system, respond to this prompt with a YES.

```
Do you want to purge the files replaced by this installation [YES]?
```

8) The installation procedure then displays the following message:

```
%VMSINSTAL-I-SYSDISK, This product creates system directory
SYS$SYSDEVICE:[S36_TOOLS]
```

```
                    NOTICE: SYSTEM MANAGER

Place the following command in the SYS$MANAGER:SYSTARTUP_V5.COM
procedure to create the system logical S36$TOOLS at system startup.

            @SYS$STARTUP:S36_TOOLS_LOGICALS.COM

Place the following command in the SYS$MANAGER:SYLOGIN.COM
procedure or individual user LOGIN.COM procedures to set up the
symbols used by the CVTAPE Utility.

            @S36$TOOLS:S36_TOOLS_SETUP.COM

%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target
directories
```

9) The following message is displayed when the installation has completed:

```
Installation of CVTFIL V1.1 completed at HH:MM

VMSINSTAL procedure done at HH:MM
```

1.1.1.1.1.1   CVTFILE Utility Setup

The user needs to execute the command procedure S36$TOOLS:S36_TOOLS_SETUP.COM to set up the local symbols and logicals necessary to use the CVTFILE Utility. It is recommended that the command @S36$TOOLS:S36_TOOLS_SETUP.COM be placed in the system SYLOGIN.COM procedure or each user's LOGIN.COM so that the CVTFILE Utility will be set up for use automatically when a user logs in.

1.2   RESTORING ACCESS TO THE CVTFILE UTILITY IF OPENVMS IS REINSTALLED OR UPGRADED

If it becomes necessary to reinstall or upgrade OpenVMS, the DCL command tables containing the CVTFILE command and the DCL help library containing the CVTFILE help text may get replaced. In order to restore access to the CVTFILE Utility should this occur, run the following procedure from the system manager's account:

```
@SYS$SYSDEVICE:[S36_TOOLS]CVTFILE_CLD
```

This will restore the CVTFILE Utility commands to the DCL tables and replace the CVTFILE help text in the online help library. If this procedure does not restore access to the CVTFILE Utility after a change to OpenVMS, reinstall the CVTFILE Utility kit using the VMSINSTAL Utility.

1.3   INSTALLATION EXAMPLE

The following text is a sample of an actual installation of the CVTFILE Utility on a VAX system.

```
@SYS$UPDATE:VMSINSTAL CVTFIL011

VAX/VMS Software Product Installation Procedure V5.4

It is 16-JAN-1999 at 08:12.

Enter a question mark (?) at any time for help.
* Are you satisfied with the backup of your system disk [YES]? <RET>
* Where will the distribution volumes be mounted: MUA0:<RET>
* Enter installation options you wish to use (none): <RET>

Please mount the first volume of the set on  MUA0:.
* Are you ready? Y<RET>
%MOUNT-I-MOUNTED, CVTFILE mounted on _MUA0:

The following products will be processed:

   CVTFIL V1.1

          Beginning installation of CVTFIL V1.1 at 08:13

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
* Do you want to purge files replaced by this installation [YES]? <RET>

%VMSINSTAL-I-SYSDIR, This product creates system disk directory
SYS$SYSDEVICE:[S36__TOOLS].

                    NOTICE: SYSTEM MANAGER

Place the following command in the SYS$MANAGER:SYSTARTUP_V5.COM
procedure to create the system logical S36$TOOLS at system startup.

                @SYS$STARTUP:S36_TOOLS_LOGICALS.COM

Place the following command in the SYS$MANAGER:SYLOGIN.COM
procedure or individual user LOGIN.COM procedures to set up the
symbols used by the CVTAPE Utility.

                @S36$TOOLS:S36_TOOLS_SETUP.COM

%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories.

SYS$STARTUP:S36_TOOLS_LOGICALS.COM - Setting up S36$TOOLS logicals.
16-JAN-1999 08:20:18.29

          Installation of CVTFILE V1.1 completed at 08:20

          VMSINSTAL procedure done at 08:21.
```

2.   THE CVTFILE UTILITY
CONVERTING FILES BETWEEN EBCDIC & ASCII

The CVTFILE Utility is used to convert files from EBCDIC to ASCII, ASCII to EBCDIC, and to rebuild logical record structures. The utility outputs a sequential file. If a relative or indexed file organization is needed, use the OpenVMS CONVERT Utility to reorganize the file.

2.1   OPERATION

The CVTFILE Utility processes an input file as an unstructured string of ASCII or EBCDIC data. The number of bytes at the beginning of the input file indicated by the /SKIP qualifier are bypassed before data conversion begins. This allows file header information that may have carried over from another system to be skipped. Then the number of bytes indicated by the /INPUT_RECORD_LENGTH qualifier are read into a buffer and converted from EBCDIC to ASCII or ASCII to EBCDIC. An optional translation file can be specified with the /TRANSLATION_FILE qualifier to identify record formats and packed-decimal and binary data to the utility. Finally, the number of bytes indicated by the /RECORD_LENGTH qualifier are output from the buffer to disk.

If a translation parameter file is present, the CVTFILE Utility will parse it before beginning the file conversion. If errors are encountered, the utility will display the offending lines and the error messages to the user's terminal.

In most cases, the input record size (/INPUT_RECORD_LENGTH) and output record size (/RECORD_LENGTH) are equal. It is possible for the output record size to be less than the input record size if the utility that was used to create the file padded the records. The output record size should never be greater than the input record size.

As CVTFILE runs, a record count is logged to the terminal every 100 records. At the end of processing, a final count is displayed indicating the total number of records written to disk. The count can be used to verify that the correct number of records were converted.

The utility outputs a sequential file with fixed-length records. If a relative or indexed file organization is needed, use the OpenVMS CONVERT Utility to reorganize the file.

The CVTFILE Utility uses null records to determine where the end of the data file is located. If CVTFILE encounters a null record while converting a file, it will treat it as an end-of-file condition and terminate the file conversion. If null records exist in a file that is to be converted, remove them or replace the null characters with blanks or zeros. To avoid null record problems when transferring a file from an IBM System/34 or System/36, REORGANIZE each data file before it is moved to the OpenVMS system.

2.2   RUNNING THE CVTFILE UTILITY

The CVTFILE Utility is invoked using the following command:

```
$ CVTFILE /RECORD_LENGTH  [/qualifiers]  input-file-spec
```

2.2.1   Input Parameters

input-file-spec

> Name of the input data file. This can be a fully qualified OpenVMS file description. The file can contain either ASCII or EBCDIC data. The CVTFILE Utility assumes a data format of EBCDIC for the input file if the data format is not specified. If an input file is not provided, the user will be prompted for it. If no extension is supplied on the input file name, an extension of .EBC will be assumed.

2.2.2   Command Qualifiers

The following qualifiers can be used to control the output and functions of the CVTFILE Utility. These qualifiers are position-independent, meaning they can be specified in any order following the CVTFILE command or the input file name.

The /ASCII and /EBCDIC qualifiers are mutually exclusive. The default input file data format is /EBCDIC. The /RECORD_LENGTH qualifier is required. If no translation file is provided, the CVTFILE Utility will assume that all data is alphanumeric and is to be converted.

/ASCII

> Indicates that the input file is an ASCII file and that the output file is to be an EBCDIC file. This qualifier cannot be specified with the /EBCDIC qualifier.

/EBCDIC (*default*)

> Indicates that the input file is an EBCDIC file and that the output file is to be an ASCII file. If /ASCII is not specified, then this is the default condition assumed by the CVTFILE Utility. This qualifier cannot be specified with the /ASCII qualifier.

/INPUT_RECORD_LENGTH=*nn*

> Specifies the number of bytes that must be read from the input file in order to get one data record plus any trailing bytes that may precede the next data record. If /INPUT_RECORD_LENGTH is not specified, then the input file record length is assumed to be equivalent to the output file record length (/RECORD_LENGTH).

/OUTPUT[=*file-spec*]

> This qualifier allows the user to specify an output file name for the converted file. This can be a fully qualified OpenVMS file description. By default, the CVTFILE Utility will use the input file name and a .SEQ or .EBC extension, depending upon whether the output file is to be ASCII (.SEQ) or EBCDIC (.EBC).

The utility outputs a sequential file with fixed-length records. If a relative or indexed file organization is needed, use the OpenVMS CONVERT Utility to reorganize the file.

/RECORD_LENGTH=nn (*required*)

This qualifier is required. It specifies the length of the output records. If the /INPUT_RECORD_LENGTH is not specified, then it is set equal to the output record length.

/SKIP=*nn*

This qualifier allows the CVTFILE Utility to skip the designated number of bytes in the input file before beginning data translation. This qualifier is useful for files which may have a header record included at the beginning of the file.

/TRANSLATION_FILE[=*file-spec*]

This qualifier allows the user to specify a translation file which is used by the CVTFILE Utility to identify various record formats within the file and to differentiate between alphanumeric, packed-decimal, and binary data. If the /TRANSLATION_FILE qualifier is not present, the CVTFILE Utility will attempt to locate a translation file using the input file name and a .TRN extension. If no translation file is found, all data in the input file is assumed to be alphanumeric and will be converted accordingly.

/VERSION

This qualifier causes version and distribution information for the CVTFILE Utility to be displayed.

2.3   THE TRANSLATION PARAMETER FILE

The translation parameter file contains parameters that define which sections of each input record are to be converted and how each section should be converted. Translation parameters fall into two categories: those that define record formats and those that define record layouts. These parameters may be created using a text editor such as EDT. The translation parameter file is used by the CVTFILE Utility to determine record formats and types.

2.3.1   Comments in a Translation Parameter File

Any record in a translation parameter file which begins with a exclamation point (!) or a blank is treated as a comment line by the CVTFILE Utility.

2.3.2   Record Format Parameters

Record format parameters are used to define a record's layout. They are used the same way that the field description entries in the input record specifications are used in an RPG program to define a record. The record format parameter layout is described in *Table 2-1*.

Table 2-1: Translation Parameter File Layout - Single Record Format

Parameter Spec /Column(s)	Entry	Description
Control Code 1	X	Must be in the first record of a translation parameter file if the data file being converted has only one record format.
	T	Must be used for all transaction parameter records other than the case above.
	blank or !	Comment line.
Start Position 2 - 5	0001 - 9999	A 4-digit, right-adjusted number defining the starting position of a section of data in the record for conversion.
6	,	Comma (delimiter).
End Position 7 - 10	0001 - 9999	A 4-digit, right-adjusted number defining the ending position of a section of data in the record for conversion. The end position must be greater than or equal to the start position.
11	,	Comma (delimiter).
Format Code 12	A	**Alphanumeric:** Identifies the data in the defined columns as being alphanumeric. Alphanumeric data is always converted. Contiguous blocks of alphanumeric data can be defined as a single field within the translation parameter file.
	B	**Binary:** Identifies the data in the defined columns as being binary. Binary fields must have a length of 2 or 4 bytes. Binary data must be defined on an individual field basis.
	P	**Packed-Decimal:** Identifies the data in the defined columns as being packed-decimal. Packed-decimal data is always converted. Contiguous blocks of packed-decimal data can be defined as a single field within the translation parameter file.

Translation Parameter Notes

- The start position (2 - 5) and the end position (7 - 10) may contain values defining blocks of data for alphanumeric or packed-decimal data.

- Binary data must be specified by individual field because the byte order in each field must be converted.

2.3.3   Record Identification Parameters

Record identification parameters are used to identify different record types for conversion in a data file containing multiple record formats. They are used the same way that the input record specifications are used in an RPG program to identify a record. They may be omitted if a file uses only one record layout. The record identification parameter layout is described in *Table 2-2*. The record layout translation parameters are described in *Table 2-3*.

Table 2-2: Translation Parameter File layout - Multiple Record Format Identification Parameters

Parameter Spec /Column(s)	Entry	Description
Control Code 1	R	Identifies the translation parameter as the beginning of a record identification group.
	A	Logical AND designation used to continue the identification of the record format that started with the previous R specification.
	O	Logical OR designation used to continue the identification of the record format that was started with the previous R specification.
	blank or !	Comment line.
Target Position 2 - 5	0001 - 9999	A 4-digit, right-adjusted number defining the position in the input record of the target character that will be used in the comparison.
6	,	Comma (delimiter).
Comparison Code 7	I	The comparison is true if the target character (columns 2 - 5) *matches* the comparison character (column 11).
	N	The comparison is true if the target character (columns 2 - 5) *does not match* the comparison character (column 11).
8	,	Comma (delimiter).
Format Code 12	C	**Character:** Comparison is performed on entire character.
	A	**Zoned:** Comparison is only performed on the zoned portion of the character.
	D	**Digit:** Comparison is only performed on the digit portion of the character.
Comparison Character 11	*character*	The character that is used to match against the target character position. The comparison character can be any legal ASCII character.

Translation Parameter Notes

- Records that fail to match any of the specified record formats will not be converted, but will be written to the output file. A message warning the user that this has occurred will be displayed on the terminal.

Table 2-3: Translation Parameter File Layout - Multiple Record Format Record Layout

Parameter Spec /Column(s)	Entry	Description
Control Code 1	T	Identifies the translation records in translation files defining multiple record layouts.
	blank or !	Comment line.
Start Position 2 - 5	0001 - 9999	A 4-digit, right-adjusted number defining the starting position of a section of data in the record for conversion.
6	,	Comma (delimiter).
End Position 7 - 10	0001 - 9999	A 4-digit, right-adjusted number defining the ending position of a section of data in the record for conversion. The end position must be greater than or equal to the start position.
11	,	Comma (delimiter).
Format Code 12	A	**Alphanumeric:** Identifies the data in the defined columns as being alphanumeric. Alphanumeric data is always converted. Contiguous blocks of alphanumeric data can be defined as a single field within the translation parameter file.
	B	**Binary:** Identifies the data in the defined columns as being binary. Binary fields must have a length of 2 or 4 bytes. Binary data must be defined on an individual field basis.
	P	**Packed-Decimal:** Identifies the data in the defined columns as being packed-decimal. Packed-decimal data is always converted. Contiguous blocks of packed-decimal data can be defined as a single field within the translation parameter file.

Translation Parameter Notes

- The Start position (2 - 5) and the End position (7 - 10) may contain values defining blocks of data for alphanumeric or packed-decimal data. However, binary data requires a field-by-field definition of the data using the From and To columns.

- Binary data must be specified by individual field because the byte order in each field must be converted.

2.4   TRANSLATION PARAMETER FILE EXAMPLES

The following sections list examples of translation parameter files and describe how the translation parameters are interpreted by the CVTFILE Utility.

2.4.1   Single Record Format Example

This is an example of the translation parameter file that would be used to convert a file using a single record format.

```
$ CVTFILE /RECORD_LENGTH=160 /SKIP=128 SCOUT.EBC
```

The translation file SCOUT.TRN contains the following translation parameters:

```
! SCOUT.TRN
!
X0001,0010,A
T0011,0099,P
T0100,0160,A
```

Figure 2-1: Single record format translation parameter file.

In this example, the CVTFILE Utility will convert the EBCDIC data file SCOUT.EBC to ASCII using the parameters in the translation file SCOUT.TRN. The output file will be labeled SCOUT.SEQ. The utility will convert the alphanumeric data specified in columns 1 - 10 and 100 - 160 from EBCDIC to ASCII. The packed decimal data in columns 11 - 99 will be left alone and transferred to the output file as is.

The X in column 1 of the first translation parameter denotes that all records in the file have the same layout and data conversion pattern. Thus, there is no need to specify record format parameters and each record is converted using the same translation specification. The file in this example was loaded from a diskette created on a System/36 with the SAVE procedure. Therefore, the first 128 bytes of the file are skipped because they will contain file header information instead of data.

2.4.2   Multiple Record Formats Example

This is an example of the translation parameter file which would be used to convert a file containing multiple record formats.

```
$ CVTFILE /ASCII /RECORD_LENGTH=20 /TRAN=BIGBLK.TRN ENGINE.DAT
```

BIGBLK is a translation parameter file containing the following records:

```
! BIGBLK.TRN
!
! Master record layout.
R0001,I,C,A
A0002,I,C,B
A0004,I,C,D
O0001,I,C,C
A0004,I,C,J
T0001,0004,A
T0005,0009,P
T0010,0011,B
T0012,0014,A
T0015,0020,P

! Detail record layout
R0001,I,C,X
T0001,0003,A
T0004,0007,B
T0008,0009,B
T0010,0020,P

! Summary record layout
R0001,N,C,A
A0001,N,C,C
A0001,N,C,X
T0001,0008,A
T0009,0014,P
T0015,0018,A
T0019,0020,B
```

Figure 2-2: Multiple record format translation parameter file.

In this example, the translation parameter file BIGBLK.TRN is used to convert the ASCII file ENGINE.DAT to EBCDIC. The output file will be labeled ENGINE.EBC. The master record layout definition uses AND and OR clauses to identify master records. It reads as follows:

```
R0001,I,C,A      If  position 1 IS equal to character A
A0002,I,C,B      AND position 2 IS equal to character B
A0004,I,C,D      AND position 4 IS equal to character D
O0001,I,C,C      OR  position 1 IS equal to character C
A0004,I,C,J      AND position 4 IS equal to character J
                    then convert the record.
```

The master record contains alphanumeric data in columns 1 - 4 and 12 - 14; packed-decimal data in columns 5 - 9 and 15 - 20; and a 2-byte binary field in columns 10 - 11. Detail records are identified using a single identification clause:

```
R0001,I,C,X    If position 1 IS equal to character X
               then convert the record
```

The detail record contains alphanumeric data in columns 1 - 3; packed-decimal data in columns 10 - 20; a 4-byte binary field in columns 4 - 7; and a 2-byte binary field in columns 8 - 9.

Summary records are identified using an AND clause:

```
R0001,N,C,A    If  position 1 is NOT equal to character A
A0001,N,C,C    AND position 1 is NOT equal to character C
A0001,N,C,X    AND position 1 is NOT equal to character X
               then convert the record.
```

The summary record contains alphanumeric data in columns 1 - 8 and 15 - 18; packed-decimal data in columns 9 - 14; and a 2-byte binary field in columns 19 - 20.

2.5   CVTFILE UTILITY MESSAGES

The following sections contain descriptions of the messages output by the CVTFILE Utility.

2.5.1   Informational Messages

INFO-010 - No translation file found. Assuming all data is alphanumeric

If a translation parameter file is not specified using the /TRANSLATION_FILE qualifier and a default translation parameter file is not found, then this message is issued and the CVTFILE Utility assumes that all data in the file is alphanumeric and eligible for conversion.

2.5.2   Warning Messages

WARNING-010 - Unidentified record type in translation file. Data not converted

A record in the data file did not match the record layout descriptions in the translation parameter file. The record will be written to the output file without being converted.

WARNING-012 - No field trans parameters found for record. Data not converted

A record has failed to match any of the specified record formats and will not be converted. It will be written to the output file as is.

2.5.3   Error Messages

ERROR-010 - Input file not found

The input file cannot be found or a channel to the file cannot be opened. Verify the location of the input file, the syntax of the input file name, the process file limit, and the file and directory protection set in the current work area. Read the associated OpenVMS error message and correct the problem.

ERROR-015 - Record length (/RECORD_LENGTH=) not specified in command line

The output record length (/RECORD_LENGTH=*nn*) has not been specified. Correct the command line and try again.

ERROR-020 - Error opening output file

The output file cannot be opened or a channel to the file cannot be established. Verify the syntax of the output file name, the amount of free disk space, the process file limit, and the file and directory protection set in the current work area. Read the associated OpenVMS error message and correct the problem.

ERROR-030 - Error reading input file

The utility is unable to read a record from the input file. Read the associated OpenVMS error message and correct the problem.

ERROR-040 - Translation file not found

The translation parameter file specified in the /OUTPUT qualifier could not be found. Verify the location of the file and the syntax of the file name. Read the associated OpenVMS error message and correct the problem.

ERROR-042 - Error opening translation file

A channel to the translation parameter file specified in the /OUTPUT qualifier could not be opened. Verify the process file limit and the file and directory protection set in the current work area. Read the associated OpenVMS error message and correct the problem.

ERROR-044 - Error(s) encountered parsing translation file

Errors were encountered while parsing the translation parameter file. Correct the errors and try again.

ERROR-046 - Error reading translation file

An error was encountered while reading the translation parameter file. Read the associated OpenVMS error message and correct the problem.

ERROR-050 - IS/NOT invalid

Code specified in column 7 of a record identification parameter is not valid. Column 7 should contain an I or an N.

ERROR-060 - C/Z/D invalid

Code specified in column 9 of a record identification parameter is not valid. Column 9 should contain a C, Z, or D.

ERROR-070 - Commas misplaced in translation record description

The commas in the record identification translation parameter being displayed are not in the correct columns. Commas should appear in columns 6, 8, and 10 in a record identification parameter.

ERROR-080 - Invalid column number in translation file

A column number in the translation parameter file contains invalid characters.

ERROR-082 - Column number exceeds input record size in translation file

A column number in the translation parameter file exceeds the output record size specified in the /RECORD_LENGTH qualifier.

ERROR-090 - Invalid record code in translation file

The code specified in column 1 of the translation parameter is not valid.

ERROR-100 - Commas misplaced in translation field entry

The commas in the data translation parameter being displayed are not in the correct columns. Commas should appear in columns 6 and 11 in a data translation parameter.

ERROR-110 - Translation type not A (alphanumeric), B (binary), or P (packed)

The code entered in column 12 is not valid. Valid entries are A, B, or P.

ERROR-120 - Invalid start column in translation file

The start column contains invalid characters.

ERROR-130 - Invalid end column in translation file

The end column contains invalid characters.

ERROR-140 - Start column greater than End column in translation file

The start column in the displayed translation parameter is greater than the end column.

ERROR-150 - Record description not preceded by data translation parameter

A translation parameter beginning with an R code has appeared immediately following an parameter beginning with X, R, A, or O. At least one data translation parameter (T) must appear between each set of record identification parameters.

ERROR-160 - Embedded AND/OR entry in translation file

An And or Or translation parameter has appeared out of sequence in the translation parameter file.

ERROR-170 - Translation entry not preceded by an R, A, O, or X parameter

The data translation parameter was not preceded by a record (R), And (A), Or (O), or single record format (X) parameter.

ERROR-180 - Invalid ASCII record ID character

The utility is unable to convert a character encountered in the file. ERROR-200 - Binary field not 2 or 4 bytes long A binary field translation definition has been encountered which is not 2 or 4 bytes long.

3.   THE CVTAPE UTILITY - DOWNLOADING IBM SYSTEM/34 AND SYSTEM/36 9-TRACK TAPES

The CVTAPE Utility is used to read 9-track tapes created on an IBM System/34 or System/36 directly into an OpenVMS system. The SAVE procedure is recommended for copying data files to tape on the IBM system. There may be multiple files on a single tape or a single file may span several tapes. The data found on the OpenVMS system disk after copying a SAVE-created file from tape will be distinguished by the following characteristics:

The data will be in EBCDIC format and must be converted to ASCII format.

The data will exist as an unstructured string of bytes and must be converted back to the correct logical file format.

The CVTFILE Utility can be used to convert the data and restructure the records.

If a utility other than SAVE is used on the System/34 or System/36, or if the tapes are created on another machine (such as the System/3 or a key-to-disk data entry system), the CVTAPE Utility may not be able to read the tape. If this is the case, see the Copying Non-Standard Tapes to Disk section later in this chapter. Once the files have been downloaded, the CVTFILE Utility can be used to convert and reformat the data.

3.1   CVTAPE OPERATION

The CVTAPE Utility reads tapes created by an IBM system and then copies the information to disk on an OpenVMS system. The utility begins by reading the tape directory and validating the format. If the format is correct, a directory listing is displayed showing the names of all files on the tape, the volume number (if it is a multi-volume file), and the number of blocks occupied by each file on the tape. A file-by-file copy is then performed from tape to disk, with the output file going to the user's default device and directory.

The name under which the file will be stored on disk is composed by selecting the first eight valid alphanumeric characters from the name as shown in the directory listing. Characters that may have been valid on the IBM system, but will not be valid on the OpenVMS system, are replaced with the underscore (_) character. Valid characters for file names on the OpenVMS system are "A" through "Z", "0" through "9", the dollar sign ($), and the underscore (_). Periods and blanks in the IBM file name will be omitted on the OpenVMS system. The file name will be given a file extension of `.EBC' if it is a SAVE file.  If not, the file extension will be `.D00'. The file name is displayed on the terminal as the file is being copied. Care should be taken that duplicate names are not created for different data files during file name translation.

A full 2400-foot reel at 1600 bpi will require approximately thirty (30) minutes to copy using the CVTAPE Utility.

The data stored on the OpenVMS system will be an image copy of the file as it resided on the tape. This means that the data is still in EBCDIC format. To utilize this data, it must be converted from EBCDIC to ASCII and have its logical record structure rebuilt. The CVTFILE Utility may be used for this purpose.

3.2   RUNNING CVTAPE

Use the following instructions to execute the CVTAPE Utility:

1) Before using CVTAPE, it is a good idea to do a SHOW DEVICE/FULL tape-device to check if the tape device to be used using is already allocated to another process. If the device is already allocated to another process, use a different tape drive or try again later.

2) Use the SET DEFAULT command to move to the directory to which the files on tape are to be restored. The CVTAPE Utility will restore the files to the directory from which it is called.

3) When you are ready to proceed, type in the following line at the DCL prompt:

```
$ CVTAPE tape_device
```

The following message will be displayed:

```
S/36 Convert Tape Utility
```

If a tape device is not supplied on the command line, the utility will display the following prompt:

```
Tape Device>
```

Enter in the name of the tape drive to be used to download the files.

4) CVTAPE automatically allocates the requested tape drive and mounts the tape with a density of 1600 bpi. It then examines the contents of the tape, displaying the message:

```
Examining tape...
```

5) CVTAPE will restore all data files that were saved via the SAVE command.

As CVTAPE restores each data file, the following message will display.

```
                 Record  Block   Save Proc
File Label    Seq #  Length  Length  Used
-------------------------------------------------
EMPMSTR       0001   00256   24576    SAVE
EMPHIST       0002   00512   24576    SAVE
```

This report will also be sent to the printer defined by SYS$PRINT when CVTAPE finishes downloading the files.

6) When CVTAPE has completed restoring files from the tape, it will display a directory listing of the restored files.

```
Directory $DISK1:[PAYROLL.DATA]
EMPMSTR.EBC;1  EMPHIST.EBC;1

Total of 2 files.

End S/36 Convert Tape Utility
```

7) The tape drive is automatically dismounted and deallocated when all data files have been read.

3.3   ABORTING CVTAPE

Aborting the CVTAPE procedure while it is running may result in temporary work files being left on the disk. These files begin with TMP_ and have a time/date stamp as part of the extension or a .TMP extension. The files can be deleted to recover disk space.

3.4   CVTAPE MESSAGES

The following sections contain descriptions of the messages output by the CVTAPE Utility.

3.4.1   Warning Messages

CVTAPE - WARNING - Above file was not created using SAVELIBR or SAVE

You may not be able to convert it. If a utility other than SAVE is used on the System/34 and System/36 or if the tapes are created on another machine (such as the System/3 or a key-to-disk data entry system), the CVTAPE Utility may not be able to read the tape. See the Copying Non-Standard Tapes to Disk section later in this chapter for information on how to copy the data to disk manually.

3.4.2   Error Messages

CVTAPE - Error detected, CVTAPE Utility aborting

An error has caused the CVTAPE Utility to abort execution. Look for another error message preceding this one.

CVTAPE - Tape-device does not exist

The device name entered does not exist. Check and correct spelling of device name entered.

CVTAPE - Tape-device is not a tape drive

The device name entered is not a tape drive. Correct the device name.

CVTAPE - Tape-device is allocated to another process

The tape drive is already in use. Enter a different tape drive device name or try again later when the device is available.

CVTAPE - Tape-device is not available

The tape drive is not available. Enter a different tape drive device name or try again later when the device is available.

CVTAPE - Tape-device is already mounted

Device name entered is already mounted. Enter a different tape drive device name or try again later when the device is available.

CVTAPE - End-of-tape encountered unexpectedly

CVTAPE could not find the end-of-tape marker or what appears to be an end-of-tape marker was encountered unexpectedly. Clean the tape drive and try reloading the tape with CVTAPE. If this does not work, recreate the tape or load it manually.

CVTAPE - You must be at VMS V5.4 or higher. You are at VMS v.u

OpenVMS Version 5.4 or higher is required for the execution of CVTAPE. See your system manager regarding upgrading OpenVMS to Version 5.4 or higher.

3.5   CREATING MEDIA FOR DATA TRANSFER FROM THE SYSTEM/34 AND SYSTEM/36

Before a System/36 tape can be loaded, it must be correctly created on a System/36. Review the following instructions to ensure that your tape has been correctly built.

These instructions describes how to create tapes to transfer data from an IBM System/34 or System/36 to a Digital OpenVMS system via 9-track tape. Follow the steps on the attached pages to ensure a successful transfer of data from the System/36 to the OpenVMS system. In addition, please keep in mind the following key point:

When transferring data files onto tape from the System/34 or System/36, use the SAVE procedure. Do not use the TRANSFER procedure.

3.5.1   Steps to Transfer Files from an IBM System/34 or System/36 to 9-Track Tape

Use the following steps to transfer data files from an IBM System/34 or System/36 to a 9-track tape:

1) Initialize the tapes on the IBM system before using them. The following command will initialize a tape on the IBM system with a volume identification of IBMIRD in drive T1.

Note: Drive T1 or T2 may be substituted according to a site's specific needs.

```
TAPEINIT T1,SL,IBMIRD,CLEAR
```

Repeat this step as often as needed for each tape.

2) Transfer each data file to tape using the SAVE procedure. Use the following command, substituting the name of the desired file for filename:

```
SAVE filename,,,,T1,NOAUTO
```

3) Use the following command to verify that the data files have been successfully transferred to the tape:

```
CATALOG ALL,T1
```

Once a tape has been created on an IBM system using these instructions, it can be downloaded on an OpenVMS system using the CVTAPE Utility.

3.5.2   Copying Non-Standard Tapes to Disk

If the CVTAPE Utility will not copy files from a tape created on a System/34 or System/36, then the data can usually be copied to disk manually. Use the following instructions to manually copy files from a tape to disk.

1) The tape must first be mounted as a foreign volume on the OpenVMS system. Use the following command to accomplish this:

```
MOUNT/FOREIGN/BLOCK_SIZE=64000 dduu:
```

where *dduu:* is the tape drive device designation.

2) Once the tape has been mounted as a foreign volume, the files can be copied to disk using the COPY command. Use the following command to copy data from the tape to disk:

```
COPY dduu: file-spec
```

where *dduu:* is the name of the tape drive and *file-spec* is the name of the file as it will be stored on disk.

This type of copy command will display a record count of the number of records copied from the tape to disk. This record count in no way accurately reflects the true number of records in the file. The CVTFILE Utility will display an accurate record count when the file is converted from an EBCDIC to ASCII format.

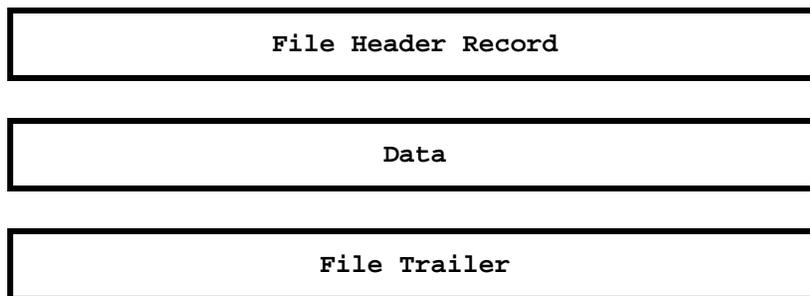3) Files are often stored on tape in the following format:

```
File Header Record
```

```
Data
```

```
File Trailer
```

Figure 3-1: File storage on tape.

When they are copied to the OpenVMS system, each portion of the format will be copied down as a separate file. Thus, three copy commands must be issued to completely copy one file from the tape. The following commands demonstrate how this can be done.

```
$ MOUNT/FOREIGN/BLOCK_SIZE=64000 MUA0:
$ COPY/LOG MUA0: HEADER.TMP
%COPY-S-COPIED, _MUA0: copied to
$DISK1:[PAYROLL.DATA]HEADER.TMP;1        (1 record)
$ COPY/LOG MUA0: EMPHIST.EBC
%COPY-S-COPIED, _MUA0: copied to
$DISK1:[PAYROLL.DATA]EMPHIST.EBC;1       (100 records)
$ COPY/LOG MUA0: TRAILER.TMP
%COPY-S-COPIED, _MUA0: copied to
$DISK1:[PAYROLL.DATA]TRAILER.TMP;1       (1 record)
```

Figure 3-2: Sample Code to Copy a File From a Non-Standard Tape.:

The header and trailer records have no meaning on the OpenVMS system and can be deleted. The data portion of the file can be kept and will need to be converted to ASCII and reformatted. The CVTFILE Utility is used for this purpose.

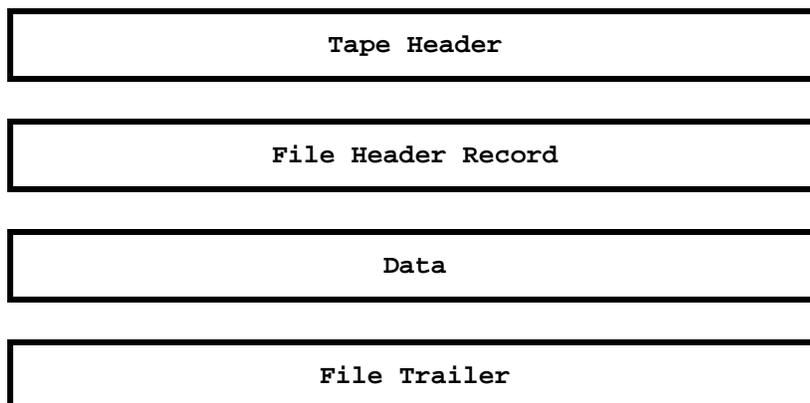In some cases, a tape also contains a tape header record that precedes all of the file entries.

```
Tape Header
```

```
File Header Record
```

```
Data
```

```
File Trailer
```

Figure 3-3: File storage format on tape including tape header record.

If this is the case, the tape header will also copy to the OpenVMS system as a data file, requiring an extra copy command to pull down the first file. The tape header can also be ignored and deleted.

To confirm the organization of data on a tape as it is copied to the OpenVMS system, use the DMP Utility to dump the contents of each file after it has been copied to disk. The DMP/EBCDIC command will allow you to examine the contents of each file and confirm that it is a header record, trailer record, or actual data. See The DMP Utility chapter in this manual for more information on using the DMP Utility.

4.   THE DMP UTILITY
DUMPING ASCII AND EBCDIC FILES

The Dump Utility is used to create a hexadecimal dump of an ASCII or EBCDIC file. The dump can be made to the terminal or a listing file. The DMP Utility allows the user to dump all or part of a file.

4.1   RUNNING THE DUMP UTILITY

The Dump Utility is invoked with the command DMP.

```
DMP [/qualifiers] input-file-spec
```

4.1.1   Input Parameters

input-file-spec

> The name of the ASCII or EBCDIC input file which is to be dumped. This is a required parameter and will be prompted for if it is not supplied. Any type of file can be submitted to the DMP Utility, but it is most useful when dumping data files containing packed or binary data fields or EBCDIC data. The file name can be a fully qualified OpenVMS file description.

4.1.2   Command Qualifiers

The following qualifiers can be used to control the output and functions of the DMP Utility. These qualifiers are position-independent, meaning they can be specified in any order following the DMP command or the input file name.

The /ASCII and /EBCDIC qualifiers are mutually exclusive, as are the /LAST and /RANGE qualifiers. The default input file data format is /ASCII. If no qualifiers are specified, the DMP Utility assumes an ASCII input file and dumps the first 10 records in the file.

/ASCII (*default*)

> Specifies that the input file is an ASCII file. This is the default unless /EBCDIC is specified. This qualifier cannot be specified with the /EBCDIC qualifier.

/EBCDIC

> Specifies that the input file is an EBCDIC file. The DMP Utility will expect ASCII input unless this qualifier is specified. This qualifier cannot be specified with the /ASCII qualifier.

/LAST

> Specifies the last 10 - 20 records in the file are dumped. By default, the DMP Utility dumps the first 10 records in the file. The /LAST qualifier cannot be specified with the /RANGE qualifier.

/OUTPUT[=*file-spec*]

   Specifies that the output is to be directed to a file. By default the output is directed to the user's terminal. If /OUTPUT is specified without a file-spec, then the input file name and a *.DMP* extension will be used to label the output file. The default extension for an output file is *.DMP*. If an output file name is specified, it can be a fully qualified OpenVMS file description.

/RANGE=*nn*[:*mm*]

   Specifies that a range of records is to be dumped. By default the DMP Utility dumps the first 10 records in the file. The /RANGE qualifier cannot be specified with the /LAST qualifier.

   nn - Numeric entry. The record number of the first record in the range to be dumped. If no last record number is specified, only this record will be dumped.

   mm - Numeric entry. The record number of the last record to be dumped. This number must be greater than or equal to the starting record number.

/VERSION

   This qualifier causes version and distribution information for the DMP Utility to be displayed.

4.2   DMP EXAMPLES

1) In this example, the first 10 records of the EBCDIC file ACCTSPAY.EBC will be dumped to the user's terminal.

```
$ DMP/EBCDIC ACCTSPAY.EBC
```

2) In this example, the last 10 - 20 records in the ASCII file AUDIT.DAT will be dumped to the output file AUDIT.DMP.

```
$ DMP /LAST /OUTPUT AUDIT.DAT
```

3) In this example, the EBCDIC file CAMPAIGN.EBC will have records 100 - 200 dumped to the output file TMP.LIS.

```
$ DMP /EBCDIC /RANGE=100:200 /OUTPUT=TMP.LIS CAMPAIGN.EBC
```

4) These are equivalent DMP commands. Both will dump a single record, the 10th record in the FOXBAT.DAT file, to the user's terminal.

```
$ DMP /RANGE=10 FOXBAT.DAT
$ DMP /RANGE=10:10 FOXBAT.DAT
```

4.3   DUMP LISTING FORMAT

A record dump consists of three lines of information. The first line is the image of each character in the data record. In the case of packed, binary, and EBCDIC data, many of these characters may not be printable. Immediately under each data character is the hexadecimal representation of the character, with the high four bits being displayed on the second line and the low four bits being displayed on the third line.

4.3.1   DMP Command Output

The following is sample output of a dump from an ASCII file generated using from the DMP Utility:

```
** RECORD NUMBER 000001 **
FELIX   29100   478       TRX A
44445445333332233322222555224
65C98D52291000047800000428001

** RECORD NUMBER 000002 **
OSCAR   29200   499       TRX B
45445445333332233322222555224
F3312D52292000049900000428002
```

Figure 4-1: Example of DMP Utility output.

In reading this dump, the character F in the word FELIX in the first record of the dump is represented by a Hex 46. The value of the high four bits (4) is displayed on the second line and the value of the low four bits (6) is displayed on the third line. The Hex value of the character E is 45; the Hex value of the character L is 4C; and so forth.

The DMP Utility will display a maximum of 100 characters on a line. Records exceeding 100 characters in length are continued on the following lines.

4.4   DMP UTILITY MESSAGES

The following sections contain descriptions of the messages output by the DMP Utility.

4.4.1   DMP Error Messages

ERROR-010 - Cannot open input file

The input file cannot be found or a channel to the file cannot be opened. Verify the location of the input file, the syntax of the input file name, the process file limit, and the file and directory protection set in the current work area.

ERROR-020 - Cannot open output file

The output file cannot be opened or a channel to the file cannot be established. Verify the syntax of the output file name, the amount of free disk space, the process file limit, and the file and directory protection set in the current work area.

ERROR-022 - Cannot open channel to terminal

The DMP Utility is unable to open a channel to the user's terminal. Read the accompanying OpenVMS error message and take corrective action.

ERROR-030 - Invalid value specified in /RANGE qualifier

A non-numeric entry has been made in the /RANGE qualifier.

ERROR-040 - Start range exceeds end range

The starting record number in the RANGE qualifier must be less than or equal to the ending record number.

ERROR-050 - Error writing output record

The DMP Utility is unable to write to the output file. Read the accompanying OpenVMS error message and take corrective action.