HP 64700 Operating Environment

Absolute File Translator

User's Guide

HEWLETT PACKARD

HP Part No. B1471-97008
Printed in U.S.A.
March 1992

Edition 2

Notice

Hewlett-Packard
P.O. Box 2197
1900 Garden of the Gods Road
Colorado Springs, CO 80901-2197, U.S.A.

Printing History

New editions are complete revisions of the manual. The date on the title page changes only when a new edition is published.

A software code may be printed before the date; this indicates the version level of the software product at the time the manual was issued. Many product updates and fixes do not require manual changes, and manual corrections may be done without accompanying product changes. Therefore, do not expect a one to one correspondence between product updates and manual revisions.

Edition 1	B1471-97005, July 1991
Edition 2	**B1471-97008, March 1992**

In This Book

This book describes the **xlate** aboslute file translator utility. Included are description of the command line options and possible error messages. The book contains the following appendices describing file formats.

Appendix A — **I_hex Absolute File Format**. This appendix describes the Intel Hexadecimal Intellec 8/MDS (I_hex) file format.

Appendix B — **Intel 286_OMF File Format**. This appendix describes the Intel 80286 bootloadable object module file format.

Appendix C — **M_hex Absolute File Format**. This appendix describes the Motorola hexadecimal S-Record (M_hex) file format.

Appendix D — **T_hex Absolute File Format**. This appendix describes the Tektronix hexadecimal (T_hex) file format.

Appendix E — **Extended T_hex Absolute File Format**. This appendix describes the extended Tektronix hexadecimal (T_hex) file format.

Appendix F — **Hewlett-Packard Absolute File Format**. This appendix describes the HP absolute file format.

Contents

1 Absolute File Translator

1

Absolute File Translator

Introduction

The absolute file translator (***xlate***) is a utility that converts Intel, Motorola, and Tektronix absolute files into HP absolute files, and vice-versa.

You can convert Intel, Motorola, or Tektronix absolute files into HP absolute files, load the converted files, and execute them in the HP 64000 emulation environments.

You also can convert HP absolute files into Intel, Motorola, or Tektronix absolute files. This may come in handy, for instance, when you wish to program ROMs with code developed in the HP 64000 or environments with a prom programmer which does not support HP absolute files, but does support Intel, Motorola, or Tektronix absolute files.

The specific absolute file formats that can be converted into HP absolute files are:

- Intel Hexadecimal Intellec 8/MDS (I_hex).
- Intel 286_OMF (bootloadable modules only).
- Motorola Hexadecimal (M_hex).
- Tektronix Hexadecimal (T_hex).
- Extended Tektronix Hexadecimal.

Note

HP absolute files can be converted into all file formats above **except** Intel 286_OMF files.

Note

The *xlate* utility will **not** convert symbol records.

Xlate Command Syntax

The following pages describe the ***xlate*** command syntax in the UNIX operating environment. Descriptions of the error messages issued by ***xlate*** appear at the end of the chapter.

Table 1. Bus and Data Width of Microprocessors

Microprocessor	Data Width	Bus Width
8080/85, Z80	8 bits	8 bits
6800/01/03/05/09	8 bits	8 bits
68HC11	8 bits	8 bits
650X	8 bits	8 bits
68000/08/10	16 bits	8 bits
8021/22, 8041/48	8 bits	8 bits
9900/40/85/89/99	16 bits	8 bits
9980	8 bits	8 bits
99XXX	16 bits	16 bits
1802	8 bits	8 bits
F8	8 bits	8 bits
Z8	8 bits	8 bits
8086/88/89/186/188/286	16 bits	8 bits
Z8001/2	16 bits	8 bits
8051	8 bits	8 bits
1750A	16 bits	16 bits
TMS320	16 bits	16 bits
8096	16 bits	8 bits

xlate(1)

Absolute File Translator

Synopsis

```
xlate -t<type> [-f] [-d<data width>]
[-b<bus width>]  [-o<output_file>]  input_file
.  .  .
```

Description

The **xlate** utility is used to convert absolute load files from any of several formats to the HP 64000 absolute file format or to convert HP 64000 absolute files to any of the other file formats. These formats include Intel Hexadecimal, Tektronix Hexadecimal, Extended Tektronix Hexadecimal, Motorola S Record, and Intel 286_OMF.

Note

The **xlate** utility does not convert HP 64000 absolute files into 286_OMF files.

The legal ASCII data in any hex file can only be separated by spaces, tabs, or newlines; otherwise, translation will stop and an appropriate error message will be printed.

Parameters

Definitions for syntactical terms are as follows:

-t< type>

Specifies the type of file to be converted where the valid types are:

ih80	I_hex format (8080/8085).
ih86	I_hex format (8086/8088).
iomf	Intel 286_OMF format.
tek	T_hex format.

etek	Extended T_hex format.
mot	Motorola S Record format.

input_file

The name of the file to be converted (any valid UNIX file specification). The output file has the same name with a ".X" extension if it is an HP absolute file, or a ".H" extension otherwise. Specifying multiple input files is the same as entering the **xlate** command for each of the files.

Options

The **xlate** command line options are defined as follows:

-f

This option specifies that the translation will be from the HP absolute file format.

If the **-f** option is not used, the translation will be to an HP 64000 absolute file by default.

-o< output_file>

This option allows you to specify an output file name. By default, the output file has the same base name as the input file with a ".X" appended for HP 64000 format files and a ".H" appended otherwise.

-d< data width>

Data Width Base: This is the size of the smallest addressable unit of the target microprocessor. Legal values are 8 and 16. The default is 8 if this option is not used.

-b< bus width>

Data Bus Width: This is the data bus width of the target microprocessor. Legal values are 8 and 16. The default is 8 if this option is not used.

-s

Scale Address. This option is for use with processors with a 16-bit basic addressable unit (data width base). Normally, when addresses for these processors are encountered in a hex file, they are treated as word addresses; that is, the address is incremented once for each two bytes of data in the file. This is not a completely standard practice, and some vendors treat the addresses as byte addresses. Under these circumstances, the **-s** option must be used to read or produce files that are compatible with other software.

Examples

Here are two examples of the **xlate** command:

```
xlate -ttek -b16 tek_abs_file.H <RETURN>
```

The command above converts a Tektronix Hexadecimal Format absolute file (for a 16-bit microprocessor) into an HP 64000 absolute file.

```
xlate -ftmot -b16 hp_abs_file.X <RETURN>
```

The command above converts an HP 64000 absolute file (for a 16-bit microprocessor with a 16-bit data width) into a Motorola S Record format file.

Note

The file format converter is contained in public directory ***/usr/hp64000/bin***. If */usr/hp64000/bin* is in the user's directory path, a conversion can be run by using only the command *xlate*.

Error Messages

This section lists, in alphabetical order, the error messages which can occur when using the **xlate** utility. There are two types of messages:

- Informational Messages. The **xlate** utility will issue these messages to warn you about certain situations. The file format conversion will continue.
- Fatal Errors. These errors will cause file format conversion to be halted.

The error messages, along with descriptions of why the error occurred, are listed below.

Informational Messages

Abort record encountered

This message is informational and indicates that an abort record was read at some point during the processing of the input file. The translator will continue translation, and the output file should be usable.

Bad checksum value, data may be incorrect

This message indicates that an incorrect checksum value was read from the file. The translator will print this message once for each record with an incorrect checksum. The translation does not stop, and the output file will be usable provided that the data read from the file was correct despite the incorrect checksum.

No end record encountered

This message indicates that no end record was read from a file format that supports end records. This will not affect translation, except that any information contained in the end record, usually a transfer address, will not be available.

Fatal Errors

Attempt to find current file position was unsuccessful

An attempt to locate the current position in output file failed.

Attempt to move to new file position failed

An attempt to reposition in the output file was unsuccessful.

Attempted to read end of file

The translator encountered end of file while not on a record boundary.

Bad or missing translation file type specification

The filetype specification was omitted or incorrectly specified.

Could not complete translation from HP 64000 format

The translator was unable to translate one of the records in the input file. Other messages will be combined with this message to better indicate the cause of the failure.

Could not complete translation to HP 64000 format

The translator was unable to translate one of the records in the input file. Other messages will be combined with this message to better indicate the cause of the failure.

Data width base must be 8 or 16

The data width base specification is incorrect. Check that the specification is either 8 or 16.

Input absolute file name exceeds 253 characters

The input file name specification is too large. If the specification is correct then you must rename the file to allow the translator to translate the file.

Invalid address, address too large for processor

An attempt was made to create an address (load or transfer) that was to large for the output file format to accommodate.

I/O error on HP 64000 absolute file during close

The HP 64000 format file could not be closed.

I/O error on HP 64000 absolute file during create

The HP 64000 output file could not be created. Check that the file protections allow you to create and write to the output file.

I/O error on HP 64000 absolute file during open

The HP 64000 input file could not be opened. Check that the specified file exists and that the file protections allow you read access to the file.

I/O error on HP 64000 absolute file during read

The HP 64000 input file could not be read. Check that the file protections allow you to read the file.

I/O error on HP 64000 absolute file during rewrite

An attempt to modify the processor information record in the HP 64000 file failed.

I/O error on HP 64000 absolute file during write

The HP 64000 output file could not be written to. Check that the file protections allow you to write to the output file.

I/O error on HP 64000 binary file during close

The HP 64000 format file could not be closed.

I/O error on HP 64000 binary file during create

The HP 64000 output file could not be created. Check that the file protections allow you to create and write to the output file.

I/O error on HP 64000 binary file during open

The HP 64000 input file could not be opened. Check that the specified file exists and that the file protections allow you read access to the file.

I/O error on HP 64000 binary file during read

The HP 64000 input file could not be read. Check that the file protections allow you to read the file.

I/O error on HP 64000 binary file during rewrite

An attempt to modify the processor information record in the HP 64000 file failed.

I/O error on HP 64000 binary file during write

The HP 64000 output file could not be written to. Check that the file protections allow you to write to the output file.

Load file not in correct format for processor

The translator read invalid characters or characters in the wrong location for the specified file type. Check that the specified file type matches the type of file being translated and that the direction of translation is properly specified.

Non hex characters in inappropriate location of file

The translator encountered invalid characters when reading a hex file. This message usually indicates that an attempt was made to translate a file that was not a hex absolute file of the appropriate type.

Translation not supported by this utility

The translator could not perform the requested translation. Check that the file format type is correctly specified. Also, note that translation from HP 64000 format files to Intel 80286 OMF files is not supported.

Unable to close absolute file

The translator could not close one of the files used in the translation.

Unable to close the non-HP file

The translator could not close one of the files used in the translation.

Unable to create absolute file

The translator could not create the specified file. Check that file protections allow the creation of the output file.

Unable to open the non-HP file

The translator could not open one of the files required for the translation. You should check that the input file exists as specified to the translator and that you have the ability to create the output file (i.e., the correct permissions).

Unable to perform a file read

The translator could not read the specified file.

Unable to perform a write to a file

The translator could not write to the output file. Check that file protections allow you to write to the output file.

Unable to read absolute file

The translator could not read the specified file. Check that the file protections allow you to read the input file.

Unable to read record

The translator could not read one of the records in the input file. If this message is accompanied by a message indicating that the input file could not be read, that is the cause of the error; otherwise, there is some problem with the record structure of one of the records in the file.

Unable to read XLATE load file

The translator could not read the specified file. Check that the file protections allow you to read the input file.

Unable to rewind file

An attempt to rewind the input file failed.

Unable to translate record

The translator was unable to translate one of the records in the input file. Other messages will be combined with this message to better indicate the cause of the failure.

Unable to write to absolute file

The translator could not write to the output file. Check that file protections allow you to write to the output file.

Word count exceeds 125 16 bit words or 250 bytes

This is an internal error in the translator contact HP.

A

I_hex Absolute File Format

Introduction

Intel Hexadecimal Intellec 8/MDS (I_hex) File Format for paper tape records consists of several fixed fields. An example follows:

`: ll aaaa tt dddd..dd xx`

(spaces are inserted for clarity, each letter represents one hex character)

Where:

:	record header character
ll	record length (number of data bytes)
aaaa	16-bit load address (or offset for 8086)
tt	record type number (see table A-1)
dddd..dd	hex data (if any)
xx	checksum (xx = 2's complement (ll+ aa+ aa+ tt+ dd+ dd+ ...+ dd)).

The various Intel record types and their field contents are shown in table A-1.

The older 8080 type transfers consisted of record types 00 and 01 only (see table A-1). The 8086 and 8088 transfers use all four types of records.

Examples of different types of records are shown below.

Table A-1. Intel Record Types

Data Record:

Record type = 00
Record length = nn (two hex digits) # data bytes
Load address = load address
Data = N data bytes

End Record: (:00 0000 01 FF)

Record type = 01
Record length = 00
Load address = *transfer address
Data = none

Extended Address Record: (used with 8086 only)

Record type = 02
Record length = 02
Load address = 00 (meaningless)
Data = USBA (16-bit upper segment base address)

Start Record: (only output for 16-bit processors but is recognized by terminal)

Record type = 03
Record length = 04
Load address = 00 (meaningless)
Data = CS and IP of 8086 processor (Start ADDR)

*NOTE: This field contains the transfer address only when 8080-type format is specified. This field is meaningless when using the 8086/8088 format.

Start Record Example

:04 0000 03 0000 1234 B3

(spaces added for clarity)

Where:		
	04	byte count of data bytes
	0000	load address (meaningless)
	03	record type = start record
	0000	8086/8088 CS register value
	1234	8086/8088 IP register value
	B3	Checksum = two's complement of: 04+ 00+ 00+ 03+ 00+ 00+ 12+ 34.

Extended Address Record Example

:02 0000 02 1000 EC

(spaces added for clarity)

Where:		
	02	number of data bytes
	0000	load address (meaningless)
	02	record type = extended address record
	1000	USBA value. SBA = USBA*10H = 1 0000H
	EC	Checksum = two's complement of: 02+ 00+ 00+ 02+ 10+ 00.

Data Record Example

:03 0A00 00 E# 7C 47 4D

(spaces added for clarity)

Where:

03	byte count
0A00	16-bit load address
00	record type = data record
E3,7C,47	data
EC	Checksum = two's complement of: 03+ 0A+ 00+ E3+ 7C+ 47.

B

Intel 286_OMF File Format

Introduction

The **xlate** utility will convert Intel 286_OMF bootloadable module object files into HP absolute format files. **However, the "xlate" utility does not convert HP absolute files into Intel 286_OMF files.** The file format for Intel 286_OMF bootloadable module object files is described below. (The number of bytes in a field, if constant, is shown in parentheses following the field descriptions.)

Bootloadable Module Object Files

A2	Bootloadable Module Header	Partition	. . .	Partition	Checksum (1)

A2 — The file header for bootloadable files.

Checksum — Equal to the one’s complement of the (8-bit) sum of all bytes in the file.

Bootloadable Module Header

Total Space (4)	Date (8)	Time (8)	Creator (41)	GDT Limit	GDT Base (4)	IDT Linit (4)	IDT Base (4)	TSS Selector (2)

Total Space — The total number of bytes of the bootloadable file.

Date — Date of module creation in the form: MM/DD/YY.

Time — Time (military) of module creation in the form: HH:MM:SS.

Creator — Name of the program which created the module.

Partition

Table of Contents	Absolute Text Section	Debug Text Section

Section — Either an absolute text section or a debug text section. Both are described below.

Table of Contents

Abs. Text Loc. (4)	Debug Text Loc. (4)	Last Loc. (4)	Next Partition	Reserved (4)

Abs. Text Loc. — The byte offset of the absolute text section from the start of the file.

Debug Text Loc. — The byte offset of the debug text section from the start of the file (zero if there is no debug text sextion).

Last Location — The location of the first byte after the partition.

Next Partition — The location of the next partition's table of contents, or zero if there is no following partition.

Reserved — These bytes should be zero.

Absolute Text Section

Read Address (3)	Length (2)	Text	. . .	Real Address (3)	Length (2)	Text

Real Address — The load address of the text.

Length — The number of following text bytes.

Text — The absolute bytes to be loaded.

Debug Text Section

No. of Segments (2)	Length (2)	Location (2)	. . .	Location (2)	Text

No. of Segments	The number of debug segments in the file.
Length	The number of location bytes which follow.
Location	The byte offset of a debug segment from the start of the file. The location equals 0 if the debug segment does not exist.
Text	Contains a concatenation of the bytes from the debug segments in the order that their locations are previously listed.

C

M_hex Absolute File Format

Introduction

Eight Motorola hexadecimal (M_hex) record types are supported. They are distinguished by the eight record headers S0, S1, S2, S3, S5, S7, S8, and S9. The record types are discussed in the paragraphs below.

S0 Type

The S0 type is the start record and may contain miscellaneous information. The format for an S0 type record is as follows:

`S0 bb yyyyyyyy xx`

(spaces are inserted for clarity, each letter = one hex character)

Where:

bb	is the byte count
yy...yy	is arbitrary hex data
xx	is the checksum (see S1 record checksum).

S1 Type

The S1 type is a data record beginning on a 16-bit load address. The format for an S1 type record is as follows:

`S1 bb aaaa dd...dd xx`

(spaces are inserted for clarity, each letter = one hex character)

Where:

Where:	bb	# of data bytes + # of address bytes + 1 (checksum)
	aaaa	16-bit load address
	dd...dd	Data bytes (or words) up to 24 bytes per record
	xx	Checksum = one's complement of: bb+ aa+ aa+ dd+ ...+ dd.

Example: **`S1 05 A000 6C 01 ED`**

(spaces added for clarity)

S2 Type

The S2 type is a data record starting on a 24-bit load address. The S2 type format is the same as an S1 type record except for a 24-bit load address (6 hex characters versus 4). The format for an S2 type record is as follows:

`S2 bb aaaaaa dd...dd xx`

(spaces are inserted for clarity, each letter = one hex character)

Example: **`S2 06 A00000 6C 01 EC`**

(spaces added for clarity)

S3 Type

The S3 type is a data record starting on a 32-bit load address. The S3 type format is the same as an S1 type record except for a 32-bit load address (8 hex characters versus 4). The format for an S3 type record is as follows:

`S3 bb aaaaaaaa dd...dd xx`

(spaces are inserted for clarity, each letter = one hex character)

Example: **`S3 07 A0000000 6C 01 EB`**

(spaces added for clarity)

S5 Type

The S5 type contains the number of S1, S2, and S3 records which are in the record block. (A record block begins with an S0 record and ends with an S7, S8, or S9 record.) The format for an S5 type record is as follows:

`S5 bb cccccccc xx`

(spaces are inserted for clarity, each letter = one hex character)

Where:

bb	is the byte count
cccccccc	is a count of the number of S1, S2, and S3 records in the block
xx	Checksum = one's complement of: bb+ cc+ cc+ cc+ cc.

Example: **`S5 05 00008000 F2`**

(spaces added for clarity)

S7 Type

The S7 type is an end record for a block of S3 records and has the following format.

`S7 bb tttttttt xx`

(spaces are inserted for clarity, each letter = one hex character)

Where:

bb	is the byte count
tttttttt	is an optional 32-bit transfer address
xx	Checksum = one's complement of: bb+ tt+ tt+ tt+ tt, or the one's complement of "bb" if the transfer address is not present.

Example: **`S7 05 80008000 FA`**

(spaces added for clarity)

S8 Type

The S8 type is an end record for a block of S2 records and has the following format.

`S8 bb tttttt xx`

(spaces are inserted for clarity, each letter = one hex character)

Where:		
	bb	is the byte count
	tttttt	is an optional 24-bit transfer address
	xx	Checksum = one's complement of: bb+ tt+ tt+ tt, or the one's complement of "bb" if the transfer address is not present.

Example: **`S8 04 006600 95`**

(spaces added for clarity)

S9 Type

The S9 type is an end record for a block of S1 records and has the following format.

`S9 bb tttt xx`

(spaces are inserted for clarity, each letter = one hex character)

Where:

bb	is the byte count
tttt	is an optional 16-bit transfer address
xx	Checksum = one's complement of: bb+ tt+ tt, or the one's complement of "bb" if the transfer address is not present.

Example: **`S9 03 0000 FC`**

(spaces added for clarity)

D

T_hex Absolute File Format

Introduction

The Tektronix Hexadecimal (T_hex) File Format is used to transfer 8-bit processor absolute information. The definition of T_hex format includes a data transfer protocol in addition to specifying the absolute data representation. This protocol provides for positive and negative acknowledgement of records received and for retransmission of erroneous records.

The T_hex format specifies three types of records: data, terminating, and abort. The Tektronix format specifies that a maximum of 30 data bytes may be transmitted in any one data record. The record types are discussed in the following paragraphs.

Data Record

The format for a data record is as follows:

`/ aaaa bb cc dddd...dd xx`

(spaces are inserted for clarity, each letter represents one hex character)

Where:

/	record header character
aaaa	4 hex digits specifying load address
bb	byte count (# data bytes)
cc	first checksum (cc = a+ a+ a+ a+ b+ b)
dd..dd	hex data
xx	second checksum (data checksum).

Example of a Data Record:

`/ 0A00 02 0C 2A C3 1B`

(spaces are inserted for clarity)

Where:

0A00	load address (hex)
02	data byte count
0C	1st checksum = 0+ A+ 0+ 0+ 0+ 2
2A,C3	two data bytes
1B	2nd checksum = 2+ A+ C+ 3.

Terminating Record

The format for a terminating record is as follows:

`/ aaaa bb cc`

(spaces are inserted for clarity)

Where:

/	record header character
aaaa	transfer address
bb	byte count (always 0)
cc	checksum (same as 1st checksum in data record).

Abort Record

The format for an abort record is as follows:

```
//text
```

An abort record is identified as having two header characters. The text is optional and may be used to describe the condition which caused the abort. The terminal does not send any text with an abort block. Only the device transmitting a file may send an abort. An acknowledgement is not expected after an abort record has been sent.

E

Extended T_hex Absolute File Format

Introduction

The extended T_hex file format is made up of three types of records; data records, symbol records, and termination records. Data and symbol records can be up to 255 characters long (not counting the end-of-line or carriage return).

Note

The **xlate** absolute file translator will not convert symbol records.

Variable-Length Fields

Certian fields of extended T_hex file format records can be variable lengths, i.e., from 2 to 17 characters in length. The first character of a variable-length field specifies the number of characters which follow (0 specifies that 16 characters follow in the particular field). For example, some variable-length symbols and values are: 4DONE, 3END, 5DELAY, 2FF, 3FFF, and 40000.

Valid Characters

Because symbols can be a part of the extended T_hex file format, characters other than hexadecimal characters can be a part of the file. The valid symbol characters are shown in table E-1 and equated with decimal values used for checksum computation. Data and termination records of the extended T_hex file format are described below. The character lengths of the various record fields, if known, are shown in parentheses after the field descriptions.

Table E-1. Extended T_hex Character Values

CHARACTER	DECIMAL VALUES
0 . . 9	0 . . 9
A . . Z	10 . . 35
$	36
%	37
. (period)	38
_ (underscore)	39
a . . z	40 . . 65

Data Record

Header Field (6)	Load Address (2 - 17)	Bytes of Hex Data

Load Address — This variable-length address specifies where the code is to be loaded.

Data Bytes — This part of the data record contains the object code.

Header Field

%	Record Length (2)	T	Checksum (2)

% — The "%" character specifies that the record is an extended T_hex record.

Record Length — This is the number of characters in the record (excluding the %) represented in two hex digits.

T — This character specifies the type of record where:

6 = Data Record.
3 = Symbol Record.
8 = Termination Record.

Checksum — This is a two digit hex number which equals the sum of all characters except the %, the checksum digits, and the end-of-line (or carriage return), mod 256.

Symbol Record

The **xlate** absolute file converter will not convert symbol records.

Header Field (6)	Section Name (2 - 17)	Section Def. Field *	Symb. Field	Symb. Field	

Header Field — See the previous description.

Section Name — This is a variable-length name of the section in which the symbols that follow (in the record) are found.

Section Definition Field

* The section definition field will appear in exactly one symbol record for each section.

0	Base Address (2 - 17)	Length (2 - 17)

Base Address — This variable-length address is the starting address of the section.

Length — This variable-length number equals 1 + (high address - base address).

Symbol Field

T	Symbol (2 - 17)	Value (2 - 17)

T — This character specifies the type of symbol where:

1 = Global Address.
2 = Global Scalar.
3 = Global Code Address.
4 = Global Data Address.
5 = Local Address.
6 = Local Scalar.
7 = Local Code Address.
8 = Local Data Address.

Symbol — A variable-length symbol.

Value — This is a variable-length value that is associated with the symbol.

Termination Record

Header Field (6)	Transfer Address (2 - 17)

Header Field — See the previous description.

Transfer Address — This is a variable-length address where program execution is to begin.

F

Hewlett-Packard Absolute File Format

Introduction

The absolute file consists of a variable number of records, the first of which, called the Processor Information Record, provides information about the microprocessor for which the file is intended. All subsequent records, called Data Records, are of variable length up to 128 sixteen-bit words and contain header information about the record along with data words.

For a pictorial representation of the absolute file format, see figures F-1, F-2, and F-3.

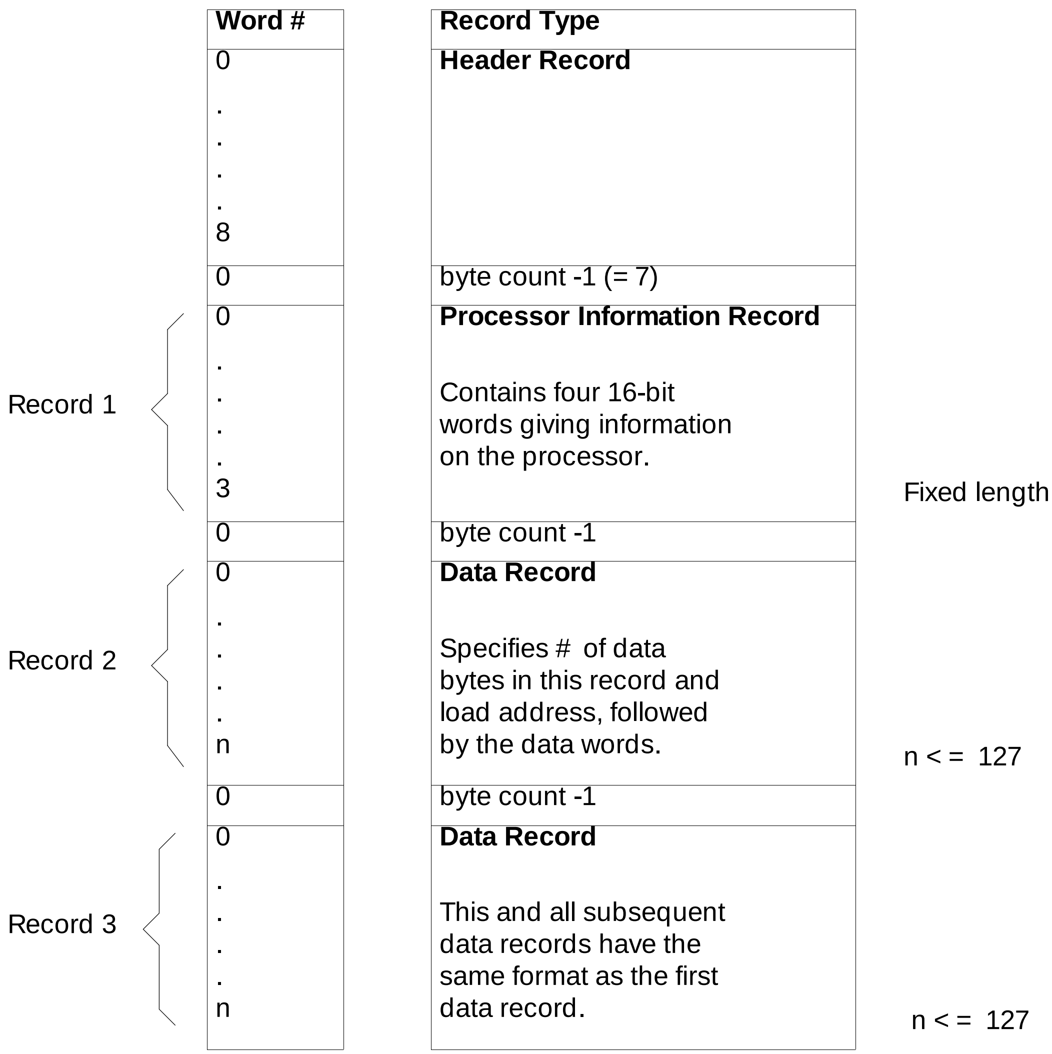

Figure F-1. HP Absolute Files - Overall Structure

Processor Information Record

WORD 0. Data Bus Width is the width of the processor data bus (8 or 16, for example).

WORD 1. Data Width Base is the minimum addressable entity (group of bits) used by the microprocessor. Usually this will be 8, but not always.

WORD 2-3. Transfer Address is the value to be loaded into the microprocessor Program Counter by the emulator. It is generated only by the linker and is set to zero when an absolute file is created by storing memory from the emulator. The Most Significant Word of the Transfer Address should be set to zero if it is not needed by the processor.

Word #	B15 B8	B7 B0
0	Data Bus Width	
1	Data Width Base	
2	Transfer Address LS Word	
3	Transfer Address MS Word	

Fixed Length = 4 words

Figure F-2. HP Absolute File - Processor Info. Record

Data Record

WORD 0. number of Data Bytes in this record expressed in binary.

WORD 1-2. Load Address is the binary address in the microprocessor memory space into which the first data byte (from WORD 3) should be loaded. Subsequent data from this record is loaded into the following microprocessor memory space. The Most Significant Word of the Load Address should be set to zero if it is not needed by the processor.

WORD 3 through n. Data Words are the binary representation of the absolute data to be loaded into microprocessor memory space.

WORD n. If the last byte of word n is not used, it should be set to 0.

Word #	B15 B8	B7 B0
0	# of Data Bytes in Record	
1	Load Address LS Word	
2	Load Address MS Word	
3	Data Word 1	
.	.	
.	.	
.	.	
n	Data Word m	

n < = 127

Figure F-3. HP Absolute File - Data Record Format

Index

Certification and Warranty

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Bureau of Standards, to the extent allowed by the Bureau's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

This Hewlett-Packard system product is warranted against defects in materials and workmanship for a period of 90 days from date of installation. During the warranty period, HP will, at its option, either repair or replace products which prove to be defective.

Warranty service of this product will be performed at Buyer's facility at no charge within HP service travel areas. Outside HP service travel areas, warranty service will be performed at Buyer's facility only upon HP's prior agreement and Buyer shall pay HP's round trip travel expenses. In all other cases, products must be returned to a service facility designated by HP.

For products returned to HP for warranty service, Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country. HP warrants that its software and firmware designated by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environment specifications for the product, or improper site preparation or maintenance.

No other warranty is expressed or implied. HP specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

Exclusive Remedies

The remedies provided herein are buyer's sole and exclusive remedies. HP shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.