

HP Computers



BASIC/260 to HP Business BASIC

Conversion Guide



**HEWLETT
PACKARD**

19111 Pruneridge Avenue, Cupertino, CA. 95014-9974

Part No. 32115-90005
E0886

Printed in U.S.A. 08/86

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company.

Printing History

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

The software code printed alongside the date indicates the version level of the software product at the time the manual or update was issued. Many product updates and fixes do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one to one correspondence between product updates and manual updates.

| | | |
|--------------------------|----------------------|---------------|
| First Edition | June 1985. | 32115.A.00.00 |
| Second Edition | August 1986. | 32115.A.01.00 |



Preface

This document describes the BASIC/260 to HP Business BASIC Conversion Utility (conversion package) and provides the instructions necessary to convert an HP 260 application for use in HP Business BASIC (BASIC in all subsequent references). This document does not give details on hand conversions, nor does it attempt to describe related subsystems on the HP 3000 (such as IMAGE). References are made to the appropriate manuals for these subsystems.

NOTE

All descriptions in this document referring to the HP260 also apply to the HP250.

Manual Description

This guide contains the following chapters and appendices:

- Chapter 1, the Introduction, gives a brief overview of the conversion process.
- Chapter 2 contains general instructions for converting files.
- Chapter 3 provides instructions for converting each specific file type.
- Chapter 4 contains instructions for converting HP260 FORM files.
- Appendix A contains instructions for using the TRNSFR program.
- Appendix B contains instructions for using DSN/DS with the PERFORM DROM.
- Appendix C lists the messages that are produced by the conversion package.
- Appendix D describes differences between BASIC/260 and BASIC, unimplemented BASIC/260 features, and the conversions that are performed automatically.

Preface (Continued)

Other Documents Referenced in this Guide

This guide refers you to a number of other documents for more detailed descriptions of products related to the conversion package. You may find it helpful to obtain a copy of the following documents before beginning the conversion process:

- *BASIC/260 Utilities Manual*
- *Business BASIC Reference Manual*
- *HP/260 Console Operator's Guide*
- *DSN/DS/260 User's Manual*
- *FORMS/260 Reference Manual*
- *HP 3000 Intrinsic Manual*
- *IMAGE/260 Programming Manual*
- *TurboIMAGE Reference Manual*
- *Report Writer/260 Programming Manual*
- *MPE Commands Reference Manual*
- *SORT-MERGE/3000 Reference Manual*
- *TIO/260 Programming Manual*
- *VPLUS/3000 Reference Manual*

Syntax Conventions

| Notation | Description |
|----------------------|---|
| Boldface | Represents literals; they are to be entered exactly as shown. |
| lowercase nonbold | Represents substitutable argument names, program names, or user-designated strings. |
| [] | Specifies that an element inside brackets is optional. Several elements stacked inside brackets means the user can select any one or none of these elements. For example: |

$\begin{bmatrix} A \\ B \end{bmatrix}$ $\begin{bmatrix} A \\ B \end{bmatrix}$ User can select A or B or neither.

When brackets are nested, parameters in inner brackets can be specified only if parameters in outer brackets or comma placeholders are specified. For example:

[parm1 [, parm2 [, parm3]]] can be entered as:

parm1, parm2, parm3 or
parm1, , parm3 etc.

| | |
|-----|---|
| { } | When several elements are stacked within braces in a syntax statement, the user must select one of those elements. For example: |
|-----|---|

$\begin{Bmatrix} A \\ B \\ C \end{Bmatrix}$ $\begin{Bmatrix} A \\ B \\ C \end{Bmatrix}$ User must select A or B or C.

| | |
|-----|---|
| ... | A horizontal ellipsis in a syntax statement indicates that a previous element can be repeated. For example: |
|-----|---|

[, itemname]...

In addition, vertical and horizontal ellipses can be used in examples to indicate that portions of the example have been omitted.

| | |
|------------|---|
| -, +, or = | Represents a flag argument, even if it appears in a position where a file name could appear. This is usually used with HP-UX. |
|------------|---|

| | |
|--|--|
| | When several elements are separated vertically by bars in a syntax statement, the user must select one of those elements. For example: |
|--|--|

A | B | C User must select A or B or C.

General Conventions

italics


Are used within text to represent substitutable argument names, program names, or user-designed strings.

underlining

When necessary for clarity in an example, user input can be underlined. For example:

NEW NAME? ALPHA



The symbol  can be used to indicate a key on the terminal keyboard. For example, RETURN indicates the carriage return key.

CONTROL char

Control characters are indicated by CONTROL followed by the character. For example, CONTROLY means the user presses the control key and the character Y simultaneously.

Contents

Chapter 1 Introduction

| | |
|---|-----|
| Purpose of the Conversion Package | 1-1 |
| Conversion Process Overview | 1-1 |

Chapter 2 The Conversion Process

| | |
|--|-----|
| Transferring the Files. | 2-1 |
| DSN/DS | 2-1 |
| TRNSFR | 2-1 |
| Using the Conversion Program | 2-3 |
| Using a STREAM File. | 2-3 |
| Commands to Convert Files | 2-3 |
| Additional Commands | 2-5 |

Chapter 3 Converting Files to BASIC

| | |
|---|------|
| Programs. | 3-1 |
| Softkeys | 3-4 |
| RESPONSE Built-in Function | 3-4 |
| OPTION INPUT LOOPS and OPTION NO INPUT LOOPS. | 3-4 |
| The WAIT Statement | 3-5 |
| Data Files | 3-6 |
| Data Bases. | 3-7 |
| Forms. | 3-10 |
| General Information | 3-10 |
| Some Differences Between FORMS/260 and VPLUS/3000 | 3-10 |
| Converting the Forms to BASIC | 3-10 |
| Using the CONVERT Command from BBCT250. | 3-11 |

Chapter 4 FORMS/260 to JOINFORM Conversion

| | |
|--|-----|
| Converting HP260 Form Files Into JOINFORM Files. | 4-1 |
|--|-----|

Appendix A Using TRNSFR

| | |
|---|-----|
| Some Considerations When Using TRNSFR | A-1 |
| Getting Started. | A-2 |
| Archive Transfer | A-4 |
| Data File Transfer. | A-6 |
| Batch File Transfer | A-7 |

Contents (Continued)

Appendix B

Messages and Their Meanings

| | |
|--|------|
| SET 1: General Messages | B-2 |
| SET 2: Program Conversion Warning Messages | B-9 |
| SET 3: Help Facility | B-11 |

Appendix C

Differences and Conversions

| | |
|--|-----|
| Differences | C-1 |
| Command Statement Only Allowed in Programs | C-1 |
| FOR-NEXT is Statically Matched | C-1 |
| CALL Line is Remembered for Return | C-1 |
| Common Areas | C-2 |
| The MERGE Statement | C-2 |
| Programmatic GETs | C-2 |
| Memory Management | C-2 |
| Display Control Characters | C-2 |
| File Names | C-2 |
| Volume Specifiers | C-3 |
| Printer Specifications | C-3 |
| IMAGE Data Base Management | C-4 |
| The BASIC Compiler | C-6 |
| Untranslatable Code | C-6 |
| Forms | C-6 |
| Data Base Utilities | C-6 |
| Advanced Data Base Statements | C-7 |
| TIO statements | C-7 |
| System dependent | C-7 |
| Media | C-8 |
| Type SHORT Numbers | C-8 |
| Miscellaneous Statements and Functions | C-8 |
| Automatic Conversions | C-9 |

Appendix D

Using PERFORM and DSN/DS

| | |
|-------------------------------|-----|
| The PERFORM File | D-2 |
| Sample PERFORM File | D-2 |

Tables

| | |
|--|------|
| Table 4-1. JOINFORM Conversions | 4-1 |
| Table A-1. LK3000 and TRNSFR Softkeys. | A-2 |
| Table C-1. Automatic Conversions | C-10 |



Chapter 1

Introduction

Purpose of the Conversion Package

The BASIC/260 to HP Business BASIC conversion package enables you to transfer an HP 260 application to an HP 3000 and then convert the application for use in BASIC. An application consists of programs, data files, forms, data-base schemas, and data-base data sets. The conversion package performs the tedious and simple conversions, but it does not guarantee that the application will immediately run in BASIC. Some hand conversion may still be required on your part before the application can run successfully.

Conversion Process Overview

The following general steps should be followed to ensure a smooth transition from BASIC/260 to BASIC:

1. Prepare the conversion site.

Both the HP 260 and HP 3000 should have the latest software releases. A link must be established between the two machines. All accounting structures should be set up on both machines. The HP 260 and the HP 3000 files should be backed up.

2. Prepare the files for transfer.

Each type of file (program, data, form, data-base schema, data-base data sets) must be in the appropriate format before it can be transferred to the HP 3000.

3. Transfer the files from the HP 260 to the HP 3000.

File transfers may be accomplished in two different ways, depending upon the hardware available. Both methods make the actual transfer relatively easy.

4. Convert the files to BASIC.

Once on the HP 3000, all of the transferred files must be run through the conversion program. This produces files for use with BASIC and other subsystems (e. g. IMAGE).

5. Hand convert any unconverted features.

After all automatic processes are done, some programs may still need manual conversion before they run correctly. The manual conversion is the user's responsibility, as well as optimization of code.

6. Make enhancements and do performance tuning in your application (optional).

Chapter 2

The Conversion Process

This chapter describes the two basic steps of the conversion process:

1. Moving the files that you want to convert to **BASIC** from the **HP260** to the **HP3000**.
2. Converting those files from the **BASIC/260** format into the **BASIC** format.

Transferring the Files

Two packages are available for transferring files from the **HP260** to the **HP3000**: (1) **DSN/DS** and **PERFORM** and (2) **TRNSFR**. Each package is described below.

DSN/DS

DSN/DS is a program that provides virtual terminal access to the **HP3000** and file transfer capabilities between the **HP260** and the **HP3000** via **DS/3000** facilities. It requires an **Intelligent Network Processor (INP)** on both systems.

The **DSN/DS** program allows you to use an **HP260** work station to do the following:

- Use the **HP260** as a remote terminal to the **HP3000**.
- Execute **HP260** commands locally.
- Transfer data files between the **HP260** and **HP3000** in either direction.
- Store **HP260** files on the **HP3000** and retrieve them on any **HP260**.
- Batch transfer multiple files from the **HP260** to the **HP3000** (you must have the **PERFORM DROM**).

For a detailed description of **DSN/DS** and for instructions on using this program, refer to the *DSN/DS User's Manual*.

TRNSFR

TRNSFR is a modified version of the **LK3000 Utility** (**LK3000** is described in Appendix C of the *TIO/260 Programming Manual*).

NOTE

Use TRNSFR instead of LK3000; LK3000 is only adequate for transferring simple ASCII files.

TRNSFR is not as quick and powerful as DSN/DS, but it is adequate for the one-time transfers in the HP260 to HP3000 conversion process. This package requires an RS232 interface on the HP260 and HP3000 computers. Set the baud rate on the HP3000 to 4800.

TRNSFR allows you to do the following:

- Execute HP260 commands locally.
- Use the HP260 as a remote terminal to an HP3000.
- Transfer ASCII and binary data files to the HP3000.
- Batch transfer multiple files from the HP260 to the HP3000.

If you are using TRNSFR, refer to Appendix A for more detailed instructions.

Using the Conversion Program

After the files have been transferred from the HP 260 to the HP3000, they must be converted into the **BASIC** format. One conversion program converts all five file types.

To run the conversion program, enter the following MPE command:

```
:RUN BBCT250.PUB.SYS
```

The program prompts you with a ">". At this point, enter the appropriate command for the type of file you want to convert. (Refer to Chapter 3 for more information on converting specific file types.)

Using a STREAM File

If you have multiple files to convert, you should transfer them to the HP3000 all at once. Then, set up a **STREAM** file on the HP3000 to run the conversion program on all of the files at once.

The **STREAM** command allows you to initiate jobs on the HP3000 while in an interactive session. MPE processes them independently in the background, allowing you to continue with your session. Thus, you can convert multiple files without tying up the terminal and interrupting higher priority tasks.

Refer to the *MPE Commands Reference Manual* for a more detailed description of the **STREAM** command.

Commands to Convert Files

The conversion program contains five commands, each of which corresponds to the specific file type it converts. This section gives the syntax of each command, and a description of the parameters.

- **PROG** <input file> [, <output file> [, <report file>]]

Use the **PROG** command to convert program files.

<input file> should be the file name of a program that was transferred from the HP260.

<output file> is the file into which the new **BASIC** program is placed. This is optional; if not specified, the **BASIC** program is listed out at the terminal.

<report file> is the file into which the warning messages are placed. This is also optional and, if not specified, the warnings are printed to the terminal.

NOTE

If the output and report files specified in the **PROG** command have the same name, only one file is opened. Also, note that you can perform a "trial" conversion on a sample file by omitting the output and report file names from the **SCHEMA** and **PROG** commands. This allows you to preview the output of the conversion program on the terminal.

- **SCHEMA** <input file> [, <output file>]

Use the SCHEMA command to convert schema files.

<input file> is the schema file that was transferred from the HP260.

<output file> is the file into which the new BASIC schema file is placed. This is optional; if not specified, the BASIC schema is listed out at your terminal.

| |
|-------------|
| NOTE |
|-------------|

When no input file is given for the PROG or SCHEMA commands, the program prompts you with a "=>". Simply type the lines to be converted. Type "//", "E", or "e" to end the command.

- **DATA** <input file>, <output file>

Use the DATA command to convert data files.

<input file> is the file name of a data file that was transferred from the HP260.

<output file> is the file into which the new BASIC data file is placed.

- **DATASET** <input file>, <data base>

Use the DATASET command to convert data bases.

<input file> is the HP260 data-base back up file that was transferred from the HP260.

<data base> is the name of the existing HP3000 IMAGE data base .

- **FORM** <input file>, <output file>, [<report file>]

Use the FORM command to convert forms files.

<input file> is the forms file that was transferred from the HP260.

<output file> is the file into which the new BASIC forms file is placed.

<report file> is the file into which all output messages are placed when the program FORMSPEC.PUB.SYS is run. The report file is optional; if not specified, all messages are printed to the terminal. This is used only for VPLUS forms.

- **CONVERT** <input file>,[<output file>],[<report file>]]

Use the CONVERT command to convert any of the five previously mentioned file types.

<input file> can be either a program file, a schema file, a data file, a data-base backup file or a forms file that was transferred from the HP260.

<output file> is the file into which the new converted file will be placed or it must be the name of the existing HP3000 IMAGE data-base if <input file> is a data-base backup file.

<report file> is the file into which all output messages are placed if the <input file> is a program file or a forms file. The report file is optional. The report file is not used, and the parameter is ignored, if the <input file> is not a program file or a forms file.

Additional Commands

The following paragraphs describe various commands that can be used in conjunction with the conversion program.

- **<mpe command>**

This command allows you to execute MPE commands from the conversion program. The colon is part of an MPE command. Type it before the command name.

- **HELP or H**

This command prints a brief summary of the conversion program commands on your screen, as well as the commands described here. Refer to Appendix C for a detailed description.

- **EXIT or E**

Use this command to exit from the conversion program and return to MPE.

- **!**

The exclamation point (!) is the comment command. It is used primarily for entering comments in a job stream. In a job stream, the ! must be preceded by a blank to prevent MPE from changing it to a colon (:). The conversion program ignores comment lines and simply places them in the output file.

- **OPTION INPUTLOOPS**

This command causes program conversion to place loops around all WAIT, INPUT, TINPUT, ACCEPT, and LINPUT statements. Refer to Chapter 3 for more information on using this command. OPTION INPUTLOOPS remains in effect until an OPTION NOINPUTLOOPS command is executed.

- **OPTION NOINPUTLOOPS**

This command causes program conversion not to place loops around all WAIT, INPUT, TINPUT, and ACCEPT statements. Refer to Chapter 3 for more information on using this command. OPTION NOINPUTLOOPS remains in effect until an OPTION INPUTLOOPS is executed. OPTION NOINPUTLOOPS is in effect when you first run the conversion program.

- **OPTION DECIMAL**

This command causes all floating point numbers and declarations to be converted into floating point decimal numbers and declarations. This effects the PROG, DATA, SCHEMA and DATASET commands. OPTION DECIMAL remains in effect until an OPTION REAL is executed. OPTION DECIMAL is in effect when you first run the conversion program.

- **OPTION REAL**

This command causes all floating point numbers and declarations to be converted into floating point real numbers and declarations. This effects the PROG, DATA, SCHEMA and DATASET commands. OPTION REAL remains in effect until an OPTION DECIMAL is executed. OPTION DECIMAL is in effect when you first run the conversion program.

- **OPTION JOINFORM**

This command causes the FORM command to convert HP250 FORMS into JOINFORM FORMS. Refer to Chapter 3 for general information on converting forms. Refer to Chapter 4 for more information on converting to JOINFORM forms. OPTION JOINFORM remains in effect until an OPTION VPLUS is executed. OPTION VPLUS is in effect when you first run the conversion program.

- **OPTION VPLUS**

This command causes the FORM command to convert HP250 FORMS into VPLUS FORMS. Refer to Chapter 3 for general information on converting forms. OPTION VPLUS remains in effect until an OPTION JOINFORM is executed. OPTION VPLUS is in effect when you first run the conversion program.

Chapter 3

Converting Files to BASIC

This chapter contains detailed instructions for converting programs, data files, data bases, and forms files to BASIC.

Programs

The following steps are required to convert program files from BASIC/260 to BASIC.

1. Back up the files using BACKUP, DUPL, or FVBACK (as described in *BASIC/260 Utilities Manual*).
2. Save an ASCII version of the program on the HP260. To do this, enter the following commands:

```
SCRATCH P
LOAD "<prog file>"
SAVE "<data file>"
```

The file <data file> contains the ASCII version of the program in <prog file>.

| |
|-------------|
| NOTE |
|-------------|

It is very important to maintain a distinction between files that contain the ASCII version of a program, and files that contain data to be read by a program. These two types of files look the same on the HP260. However, the first type of file is copied using the DATA FILE TRANSFER softkey, while the ARCHIVE TRANSFER softkey is used for the second. When you run the conversion program, you must use the PROG command to convert program files, and the DATA command to convert data files. Otherwise, unexpected results will occur.

3. Transfer the data file (created in step 1) from the HP260 to the HP3000 using one of the file transfer programs described in Chapter 2. Use the DATA FILE TRANSFER softkey. (Refer to the *DSN/DS User's Manual* for instructions on using DSN/DS. For instructions on using TRNSFR, refer to Appendix A.)

4. Convert the program file into BASIC.

To do this, run the conversion program, then enter the PROG command as described in Chapter 2. Here is an example of the PROG command and its screen output:

```
:RUN BBCT250.PUB.SYS

HP260 TO HP Business BASIC/3000 Conversion Utility
(HP32115Y.01.00). (1,2) Copyright Hewlett-Packard Co. 1985.
(1,16)

=> PROG MYFILE.CVRTIN, MYFILE.CVRTOUT, MYFILE.CVRTRPT

Records read : 138

=> EXIT

END OF PROGRAM.
:
```

There are some features of BASIC/260 for which the syntax of the equivalent feature in BASIC is slightly different. For example, in the multiple LET statement (shown below), BASIC/260 uses an equal sign (=) to separate the variables receiving the value. This statement assigns the value 5 to the variables A, B, and C.

```
10 A=B=C=5
```

BASIC uses commas to separate the variables:

```
10 A,B,C=5
```

When the conversion program encounters statements such as this, the translation is done automatically, and no warning message is issued. However, the BASIC lines will have a comment of the form "！** SYNTAX CHANGE" appended to them. This makes them easy to locate for reference by using the FIND command.

Other features in BASIC/260 either do not currently exist in BASIC, or the translation cannot be done automatically. When the conversion program encounters a feature of this type, a warning message is issued to the <report file> and a comment of the form "！** UNTRANSLATABLE" is appended to the line.

The conversion utility converts display control characters into escape sequences. However, if the resulting line is longer than 252 ASCII characters, it is not converted. Instead, the conversion utility returns the line to the state it was in before any display control characters were converted. The line is commented out even though it has the correct syntax.

NOTE

The output file created by the conversion program is an MPE ASCII file, not an BASIC data file.

5. Bring the converted program file into BASIC. To run the BASIC interpreter, type in the MPE command "BBASIC":

```
:BBASIC
```

```
HP Business BASIC/3000 HP32115A.00.00 (c) Hewlett-Packard Co. 1986
THU, JAN 16, 1986, 11:07 AM
```

6. When the BASIC prompt (>) appears on the screen, enter the GET command to access the program file that was created by the conversion program:

```
>GET "myfile.cvrout"
```

Any lines with syntax problems are displayed on the screen and entered into the program as comments. Here is an example:

```
10 SEND OUTPUT TO 17
Improper statement.
20 SEND SYSTEM OUTPUT TO 17
Improper statement.
>list

10 ! SEND OUTPUT TO 17
20 ! SEND SYSTEM OUTPUT TO 17
```

At this point you may list the program, run it, modify it, etc.

NOTE

You should SAVE the program in a new file that is a BSAVE file, rather than an ASCII file. This greatly speeds up the time it takes to GET the program.

Softkeys

Softkeys work differently in BASIC than they do on the HP260. This not only affects the general program flow, but also responses to any of the input statements in BASIC. The `OPTION INPUTLOOPS` command has been added to help you convert applications that rely upon softkeys.

Softkeys on the HP260 can interrupt a program at any time, just as the `HALT` key can. In particular, a softkey can interrupt a program when input has been requested. After processing the softkey, the HP260 program continues to wait for input. In other words, a softkey does not satisfy the input statement.

In BASIC, however, softkeys can be used only when input is requested. The softkeys may not be used for general program interrupts. Additionally, a softkey terminates an input in BASIC. Any data typed before the softkey is lost.

RESPONSE Built-In Function

BASIC has the `RESPONSE` function to help in converting softkey dependent programs. This built-in function is set by the following statements:

| | |
|------------------------|--|
| <code>INPUT</code> | Library routine <code>BB__PUSH__KEY</code> |
| <code>LINPUT</code> | Program start |
| <code>TINPUT</code> | Program stop |
| <code>ACCEPT</code> | <code>PRESS KEY</code> |
| <code>READ FORM</code> | <code>FLUSH INPUT</code> |

This `RESPONSE` function returns the following values:

- 255: `HALT` was pressed
- n to -1: `KEY n` was pressed
- 0: No input has been done or `FLUSH INPUT` was executed
- 1: Hard `HALT` was pressed
- 2: Timeout occurred
- 10: Input accepted
- 11: Input accepted without carriage return

OPTION INPUT LOOPS and OPTION NO INPUT LOOPS

The program conversion has a configurable option to indicate how the input statements are translated. You specify `INPUT LOOPS` or `NO INPUT LOOPS` in the configuration file. (The default is `NO INPUT LOOPS`.) This option can also be specified as a command to the conversion program. When `NO INPUT LOOPS` is specified, all input statements are converted into the correct BASIC syntax, with no other changes.

If option `INPUT LOOPS` is specified, all input statements are put into a loop that uses the `RESPONSE` function to control the input. This simulates the actions of the HP260. The following program segment demonstrates the conversion:

260 Program

```
100 ON KEY #1,#2 GOTO Sfkey
200 INPUT "Command: ";Com$
```

Converted BASIC Program

```
1000 ON KEY 1,2 GOTO Sfkey
2000 LOOP
2001 INPUT "Command: ";Com$
2002 EXIT IF RESPONSE > 0
2003 ENDLOOP
```

The converted program remains in the loop and continues to ask for a command if you respond with HALT or a softkey. As soon as a command is entered, the loop exits. If TINPUT or ACCEPT is used with a timeout clause, a timeout will exit the loop as well.

The following statements are converted as shown above if the option INPUT LOOPS is used:

```
INPUT
LINPUT
TINPUT
ACCEPT
```

The WAIT Statement

The WAIT statement provides an easy mechanism for using softkeys on the HP260. However, on the HP3000, softkeys cannot be used when a WAIT is active. In order to simulate the actions of the HP260 in BASIC, an infinite WAIT statement (i.e. WAIT with no time limit given) is translated as follows:

| <u>HP260</u> | <u>BASIC</u> |
|--------------|-------------------------|
| 100 WAIT | 1000 LOOP |
| | 1001 ACCEPT |
| | 1002 EXIT IF RESPONSE=1 |
| | 1003 ENDLOOP |

Data Files

The following steps are required to convert a data file.

1. Back up the file using BACKUP, DUPL, or FVBACK (as described in the *BASIC/260 Utilities Manual*).
2. Make an unprotected copy of the data file. An example of using the COPY command is shown below:

```
COPY "MYFILE,PASS" TO "MYDATA"
```

3. Transfer the data file from the HP260 to the HP3000, using the ARCHIVE TRANSFER softkey. (Refer to Appendix A for instructions on using TRNSFR, or refer to the *DSN/DS User's Manual* for instructions on using DSN/DS.)
4. Convert the data file into BASIC. Use the DATA command of the conversion program, as described in Chapter 2.

```
:RUN BBCT250.PUB.SYS
```

```
HP250 TO HP Business BASIC/3000 Conversion Utility  
(HP32115Y.01.00). (1,2)  
Copyright Hewlett-Packard Co. 1985. (1,16)
```

```
=> DATA MYFILE.CVRTIN, MYFILE.CVRTOUT
```

```
Records read: 138
```

```
*****  
Done with 250 BDATA conversion  
*****  
=> EXIT
```

```
END OF PROGRAM.
```

```
:
```

| |
|-------------|
| NOTE |
|-------------|

The output file created by the conversion program will be a BASIC data file with the file code, BDATA.

Data Bases

This section provides instructions for converting a data base. A data base consists of two parts: the description of the data structure (known as the schema) and the actual data to be organized according to that schema. This is true for both IMAGE/260 and TurboIMAGE. Thus, converting an IMAGE/260 data base is a two part process. First the schema is converted, then the data is converted and placed into the specified structure.

The following steps are required to convert a data base.

1. Back up the data base using DBSTORE or DUPL (as described in the *IMAGE/260 Programming Manual*).
2. Use the DBUNLD program to create BKUP files for the data set entries. These are the files that are transferred to the HP3000. Instructions for running this program are found in the *IMAGE/260 Programming Manual*.
3. Obtain an unprotected copy of the schema describing the data base. If a copy is not available, use the DBMODS utility on the HP260 to print a listing of the data base schema to an unprotected data file. Follow these steps:
 - a. Enter the command RUN "DBMODS,SYSTEM".
 - b. On the initial menu, fill in the DATA BASE NAME and ROOT FILE VOLUME (plus a MAINTENANCE WORD, if used).
 - c. On the main menu, press the PRINT SCHEMA key.
 - d. On the Schema Lister menu, press the PRINT TO FILE key.
 - e. Enter the name of the file to which the listing should be sent.
 - f. Press the EXIT key twice to leave DBMODS.
4. Transfer the SCHEMA listing file using the DATA FILE TRANSFER softkey.
5. Transfer the individual BKUP files using the ARCHIVE TRANSFER softkey.
6. Convert the schema file to TurboIMAGE. To do this, run the conversion program and use the SCHEMA command, as described in Chapter 2. The following example shows the command to use, along with the screen output.

```
:RUN BBCT250.PUB.SYS
```

```
HP250 TO HP Business BASIC/3000 Conversion Utility/  
(HP32115Y.01.00). (1,2) Copyright Hewlett-Packard Co. 1985.  
(1,16)
```

```
=> SCHEMA MYSCHMIN, MYSCHM
```

```
Records read: 47
```

```
=> EXIT  
  
END OF PROGRAM.  
:
```

NOTE

There is no direct support in TurboIMAGE for the floating point decimal data types available in BASIC/260 and BASIC. The BASIC/260 REAL and SHORT types (which map to DECIMAL and SHORT DECIMAL in BASIC if you use the default, OPTION DECIMAL) are represented in the converted schema as K2 and K4. This simply implies that the data is either 2-word binary coded decimal or 4-word binary coded decimal. When accessing a TurboIMAGE data base, BASIC recognizes that these data types indicate floating point decimal numbers and processes the numbers accordingly. If you choose OPTION REAL, the BASIC/260 REAL and SHORT types are mapped to REAL and SHORT REAL in BASIC and will be represented as R2 and R4 in the converted schema.

7. Create the data base from the schema.

- a. Use the DBSCHEMA utility to process the schema and create the root file.
- b. Use the MPE :FILE command to specify the schema name and the file to which the listing from DBSCHEMA should be sent.

```
:FILE DBSTEXT=MYSCHM  
:FILE DBSLIST;DEV=PP (sends the listing to the printer)
```

- c. Run the DBSCHEMA utility, using the PARM keyword to let the program know that you have specified its input file and listing file.

```
:RUN DBSCHEMA.PUB.SYS;PARM=3
```

```
HP32215B.04  
NUMBER OF ERROR MESSAGES: 0  
ROOT FILE B CREATED
```

```
END OF PROGRAM
```

The root file now exists with the name specified in the data base line of the schema. For example, assume the name is DBCUST.

```
BEGIN DATA BASE DBCUST
```

- d. Run the DBUTIL program.

```
:RUN DBUTIL.PUB.SYS  
HP3221B.04.00 IMAGE/3000: DBUTIL...
```

- e. Use the CREATE command of DBUTIL to create the empty data sets. (Refer to the *TurboIMAGE Reference Manual* for a detailed description of DBSCHEMA and DBUTIL.)

```
>>CREATE DBCUST/PASS
data base DBCUST has been CREATED.
```

- f. Enter the EXIT command to leave the program.

```
>>EXIT

END OF PROGRAM
```

8. Run the conversion program to convert and load the data sets. Use the DATASET command, as described in Chapter 2. The following example illustrates the use of this command.

```
:RUN BBCT250.PUB.SYS

HP260 TO HP Business BASIC/3000 Conversion Utility/
(HP32115Y.01.00). (1,2) Copyright Hewlett-Packard Co. 1985.
(1,16)

=> DATASET BAKUP1IN, DBCUST

Records read: 104

=> DATASET BAKUP2IN, DBCUST

Records read: 71

=> DATASET BAKUP3IN, DBCUST

Records read: 19

=> EXIT

END OF PROGRAM.

:
```

NOTE

If the data set has a user password, you are prompted for it before the conversion takes place. The terminal does not echo the password as it is entered.

The data base, DBCUST has now been created and loaded with data and can be accessed by BASIC programs or other utilities.

Forms

These paragraphs contain information pertaining to HP260 forms conversion. Included here is some general information that you should know about forms. You may choose to convert your 260 forms files into VPLUS forms or into JOINFORM forms. Please refer to Chapter 4 for information on converting to JOINFORM. This chapter explains conversion to VPLUS/3000. Some fundamental differences between FORMS/260 and VPLUS/3000 are also described. This information is followed by instructions for preparing, transferring, and converting forms.

General Information

Forms manipulation using VPLUS/3000 is probably the area of BASIC that is most different from BASIC/260. VPLUS is a combination of utility programs and procedures, used by programmers in all languages on the HP3000 to assist in obtaining and presenting data to application end users.

Forms are created and modified on the HP3000 using the utility known as FORMSPEC.PUB.SYS. This is analogous to the CFORM and MFORM utilities of FORMS/260. Forms are displayed and data is transferred between the form and the program through the use of BASIC form statements or *intrinsic*s. Intrinsic are callable routines included as part of the Fundamental Operating Software of the HP3000. They are documented in the *VPLUS/3000 Reference Manual*.

VPLUS/3000 has many features that are not in FORMS/260. Powerful field editing specifications are available. Multiple forms may exist in a single forms file, and the order may be specified for how these forms are presented to the user.

Some Differences Between FORMS/260 and VPLUS/3000

There are some fundamental aspects of FORMS/260 that are different from, and therefore incompatible with, VPLUS/3000. FORMS/260 uses the normal I/O statements of BASIC/260 (PRINT, INPUT, etc.) to access the individual fields of a form and to return control to the program after each field is entered by the user. VPLUS/3000 uses the Block Mode feature of the terminal to transmit an entire screen of information to or from the program. The normal I/O statements of BASIC cause unpredictable results if used while a form is "active". In addition, VPLUS/3000 does not support the option of changing the input, output, or tab order in a form.

Obviously, BASIC/260 programs require substantial manual conversion in the way forms are handled. The purpose of the forms portion of the conversion package is to convert the FORMS/260 form itself into a VPLUS/3000 form, which can then be manipulated by intrinsic called from the BASIC program. The layout of the screen image is retained as well as the location and attributes of the fields.

Converting the Forms to BASIC

The following procedure is recommended for converting a forms file to BASIC.

1. Back up the forms file using BACKUP, DUPL, or FVBACK (as described in the *BASIC/260 Utilities Manual*).
2. Make listings of the forms using the PFORM utility on the HP260 (as described in the *FORMS/260 Reference Manual*).

3. Create an unprotected copy of the forms file, as shown in the example below.

```
COPY "MYFILE" , "PASS" TO "MYFORM"
```

4. Transfer the forms file to the HP3000 using the ARCHIVE TRANSFER softkey.
5. Run the conversion program to convert FORMS/260 forms to VPLUS/3000. The conversion program automatically calls FORMSPEC.PUB.SYS to create the VPLUS/3000 form file. Use the FORM command, as shown in the example below.

```
:RUN BBCT260.PUB.SYS
```

```
HP260 TO HP Business BASIC/3000 Conversion Utility/  
(HP32115Y.01.00). (1,2) Copyright Hewlett-Packard Co. 1985.  
(1,16)
```

```
=> FORM MYFORMIN, MYFORM, FORMLIST
```

```
Records read: 62
```

```
=> EXIT
```

```
END OF PROGRAM.
```

```
:
```

Although FORMS/260 allows only one form per file, VPLUS allows multiple forms per file. Each form is identified by a unique form name. All converted HP260 forms are given the name of the input file and placed in separate forms files.

In the example above, the third parameter in the FORM command is the report file to which all output from FORMSPEC is sent. Any error messages that occur are also sent to this file.

Using the CONVERT Command from BBCT250

The previous sections describe the recommended procedures to follow to convert HP260 files to the HP3000 and BASIC. However, there may be some times when you want to convert many files, in batch mode, without having to determine the file type of each file. In these cases, you may want to use the CONVERT command.

When BBCT250 receives the CONVERT command, the program checks to see what the file type of the input file is. If the input file is a program file, a data file, a schema file, or a forms file, the output file is then created with the appropriate file type. But, if the input file is a data base backup file, the output file must already exist and must be the HP3000 IMAGE data base into which the data from the file will be unloaded. If the output file does not exist or the output file is not an IMAGE data base file, an error occurs and the input file (the data base backup file) is not converted.

The report file is an optional parameter, and if it is specified, it is only used if the input file is either a program file or a forms file. If the report file is specified and the input file is a schema file, a data file, or a data base backup file, a warning message is issued, the report file name is ignored, and the input file is converted.

Converting Files to BASIC

Here is an example of the CONVERT command and its screen output:

```
:RUN BBCT250.PUB.SYS
```

```
HP260 TO HP Business BASIC/3000 Conversion Utility/ (HP32115Y.01.00).  
(1,2) Copyright Hewlett-Packard Co. 1985. (1,16)
```

```
=> CONVERT MYSCHEM.CVRTIN, SCHEMOUT.CVRTOUT
```

```
SCHEMA
```

```
Records read: 53
```

```
=> EXIT
```

```
END OF PROGRAM.
```

```
:
```

| |
|-------------|
| NOTE |
|-------------|

When the CONVERT command is used, the conversion utility prints the file type of the input file to the terminal.

Chapter 4

FORMS/260 to JOINFOURM Conversion

This chapter describes the conversion of HP260 FORM files into JOINFOURM files. It also contains a table of the FORMS/260 statements and the resulting conversions to BASIC.

JOINFOURM is a FORMS/260 compatible forms package that is available in BASIC. JOINFOURM is not available to any other languages on the HP 3000. It is only supported on the HP150 and HP2392 terminals. It is intended to provide an easy-to-use alternative to VPLUS forms for current HP260 users.

Converting HP260 Form Files Into JOINFOURM Files

To convert FORMS/260 files to JOINFOURM files, run BBCT250.PUB.SYS. Use the OPTION JOINFOURM command to specify that you want to convert the forms to JOINFOURM format. Then use the FORM command (see Chapter 3) to convert your files.

The conversions in table 4-1 are all performed by BBCT250.PUB.SYS. If a GET FORM statement does not use a string literal ("formfile") or a string variable ("formfile\$"), manual conversion may be required.

Table 4-1. JOINFOURM Conversions

| HP260 | BASIC |
|--|---|
| GET FORM F\$ | OPEN FORM F\$+"."+F\$ |
| CLEAR FORM | CLEAR FORM |
| EXIT FORM | CLOSE FORM; REMAIN |
| DELETE FORM | CLOSE FORM |
| CURSOR IF#N, CURSOR OF#N, CURSOR CF#N. | CURSOR IFLD(N), CURSOR OFLD(N), CURSOR CFLD(N). |
| TFNUM | TFLD (see NOTE) |

NOTE

TFLD returns the fieldnumber of the last input field that was read. The actual cursor position and the fieldnum returned by TFLD are only identical when the fields were walked through using the RETURN key. The TAB key moves the cursor to the next (or the previous field when BACKTAB is depressed) in screen order. This is not recognized by TFLD since TAB and BACKTAB are local to the terminal. Moving the cursor using the cursor positioning keys is also not recognized by TFLD.

The following statements are not converted because they are regular BASIC statements, as well. When JOINFORM forms are active, these statements interact.

| | |
|-------|-------------|
| DISP | LDISP |
| ENTER | LENTER |
| INPUT | LINPUT |
| PRINT | PRINT USING |

Appendix A

Using TRNSFR

This appendix provides detailed instructions on using the TRNSFR program to transfer files from the HP260 to the HP3000.

Some Considerations When Using TRNSFR

When using TRNSFR, there are some important points that you should be aware of. The following suggestions will enable you to use the TRNSFR program more effectively.

- Be sure to set the baud rate on the HP3000 and the HP260 to 4800. This ensures greater accuracy in transferring files from the HP260 to the HP3000.
- Be sure the HP260 files are unprotected.
- When specifying the destination file, enter only the filename. Lockwords, groupnames, and account names are not allowed.
- TRNSFR requires the TIO and TOOLS DROM from the B.07 operating system on the HP260.



Getting Started

To get started using TRNSFR, follow this procedure on the HP260:

1. Run "CONFIG" to load the T10 DROM and configure an RS 232 port for use with the HP3000.
2. Enter the command RUN "TRNSFR".
3. Enter the number of the port that is connected to the HP3000.
4. When the HP3000 prompt (:) appears, log onto the HP3000. All MPE commands are now available.

```
:HELLO MYUSER.MYACCT,MYGROUP
```

| |
|-------------|
| NOTE |
|-------------|

Be sure to set the baud rate on the HP3000 and HP260 to 4800. This ensures greater accuracy in transferring files from the HP260 to the HP3000.

If you are accustomed to the LK3000 program, you will notice that the softkey labels used by TRNSFR are similar to the softkeys used in the LK3000. The softkey labels and corresponding numbers have been changed as shown in the table below. Softkeys not listed here are the same as those in LK3000. Refer to the *TIO/260 Programming Manual* for information on LK3000.

Table A-1. LK3000 and TRNSFR Softkeys

| LK3000 Softkey | | TRNSFR Softkey | |
|--------------------|----------|--------------------|-----------|
| Label | Numbers | Label | Number(s) |
| BAUD RATE 2400 | 4 and 12 | (Removed) | |
| TRANSFER FROM 3000 | 5 and 13 | TRANSFER FROM 3000 | 21 |
| TRANSFER TO 3000 | 6 and 14 | TRANSFER TO 3000 | 22 |
| | | BATCH | 4 and 12 |
| | | ARCHIVE TRANSFER | 5 and 13 |
| | | DATA FILE TRANSFER | 6 and 14 |

Softkeys 5, 13, 6 and 14 correspond to the labels used by the DSN/DS program. The DATA FILE TRANSFER softkey is actually used to transfer data files into HP3000 ASCII files, while the ARCHIVE TRANSFER softkey is used to transfer exact copies of files.

The following paragraphs contain descriptions of the ARCHIVE TRANSFER, DATA FILE TRANSFER, and BATCH softkeys, along with instructions for using them in the TRNSFR program.

Archive Transfer

The ARCHIVE TRANSFER softkey is used to transfer HP260 data-base data sets, data files, and forms files. More generally, the ARCHIVE transfer mode is used to transfer exact copies of files.

To execute an archive transfer, do the following:

1. Type RUN "TRNSFR".
2. Enter the number of the port connected to the HP3000.
3. Log onto the HP3000.
4. Press the ARCHIVE TRANSFER softkey.
5. Enter the HP260 source file name.
6. Enter the HP3000 destination file name. Enter only the file name; group and account names are not allowed. Each record is converted to hexadecimal and transferred to the HP3000. Afterward, the entire file is converted from hexadecimal into binary.

The example below shows the execution of an archive transfer.

```
RUN"TRNSFR"
```

```
HP 250/3000 INTERACTIVE LINK, Rev 7.A  
Enter port number (1..10): 4
```

```
:HELLO MYUSER.MYACCT,MYGROUP  
HP3000 / MPE IV F.B0.00. F.B0.00.FRI, JAN 3, 1986, 3:19PM
```

```
:(Press softkey #5)  
HP 250 TO HP3000 ARCHIVE TRANSFER  
HP250 source file name: MYDAT
```

```
HP3000 destination file name: MYDAT1  
RECORD 9 TRANSFERRED  
END OF HP250 TO HP3000 FILE TRANSFER
```

```
TRNSFR/250 HEX to BINARY or ASCII Conversion Program (HP32115A.00.00) (4, 1)
```

```
ARCHIVE file "MYDAT1" with 1 records which are 256 words long. (4, 11)  
The file "MYDAT1" was converted from HEX to BINARY. (4, 19)
```

```
END OF PROGRAM
```

```
:
```

NOTE

The HP3000 file "MYDAT1" must not exist, or the TRANSFR program reports an error and does not send the file.

Data File Transfer

As stated above, the DATA FILE TRANSFER softkey is actually used to transfer data files into ASCII files. These include program source, data-base schemas, and documentation files.

CAUTION

Do not use this key for HP260 DATA files that will be read by BASIC program or the other files transferred with the ARCHIVE TRANSFER softkey. Otherwise, undesirable results will occur.

To execute a data file transfer, do the following:

1. Type RUN"TRANSFR".
2. Enter the number of the port connected to the HP3000.
3. Log onto the HP3000.
4. Press the DATA FILE TRANSFER softkey.
5. Enter the HP260 source file name.
6. Enter the HP3000 destination file name. Enter only the file name; group and account names are not allowed. Each record is converted to hexadecimal and transferred. Afterward, the entire file is converted from hexadecimal into ASCII.

The example below shows the execution of a data file transfer.

```
RUN"TRANSFR"
```

```
HP250/3000 INTERACTIVE LINK, Rev 7.A
```

```
Enter port number (1..10): 4
```

```
:HELLO MYUSER.MYACCT,MYGROUP
```

```
HP3000 / MPE IV F.BO.00. F.BO.00.FRI, JAN 3, 1986, 3:19PM
```

```
:(Press softkey #6)
```

```
HP 250 TO HP3000 DATA FILE TRANSFER
```

```
HP250 source file name: MYDAT
```

```
HP3000 destination file name: MYDAT2
```

```
RECORD 9 TRANSFERRED
```

```
END OF HP250 TO HP3000 FILE TRANSFER
```

```
TRANSFR/250 HEX to BINARY or ASCII Conversion Program (HP32115A.00.00). (4, 1)
```

```
ASCII file "MYDAT2" with 9 records which are 160 Characters long. (4, 22)
```

```
The file "MYDAT2" was converted from HEX to ASCII. (4, 20)
```

```
END OF PROGRAM
```

```
:
```

Batch File Transfer

The BATCH softkey is used to transfer multiple files from the HP260 to the HP3000 in batch mode. Using the batch mode allows transfers to occur over long periods of time without human intervention. The batch file must be created in advance.

The HP260 editor can be used to create the batch file. The file may contain any MPE commands desired. The example shows a batch file created with the editor:

```

1  SHOWME
2  PURGE MYDATAF
3  ARCHIVE_TRANSFER MY250 TO MYDATAF
4  PURGE MYASCII
5  DATA_FILE_TRANSFER MYDAT TO MYASCII
6  LISTF MY@
7  PURGE SCHEMA
8  DATA_FILE_TRANSFER DBSCMA TO SCHEMA
9  LISTF

```

NOTE

Do not type a colon in front of MPE commands.

In addition to MPE commands, there are two special commands available for file transfers:

`ARCHIVE__TRANSFER <250-file-name> TO <3000-file-name>`

`DATA__FILE__TRANSFER <250-file-name> TO <3000-file-name>`

The above commands are equivalent to using the ARCHIVE TRANSFER and DATA FILE TRANSFER softkeys. All HP260 files must be in the correct format before the batch file is executed.

To execute a batch file transfer, do the following:

1. Type RUN "TRANSFR"
2. Enter the number of the port connected to the HP3000.
3. Press the BATCH softkey.
4. Enter the name of the batch file. From this point, the TRANSFR program begins to execute the commands in the batch file.

Using TRNSFR

The example shows you how to execute a batch transfer using the batch file shown above.

RUN"TRNSFR"

HP 250/3000 INTERACTIVE LINK, Rev 7.A

Enter port number (1..10): 4

:

:(Press softkey #4)

HP 250 TO HP3000 BATCH FILE MODE

HP250 Batch file name: BATCH

:SHOWME

USER: #S7,MARK.BASIC,MYGROUP (NOT IN BREAK)

MPE VERSION: HP32033F.B0.00. F.B0.00.

CURRENT: MON, JAN 6, 1986, 12:37 PM

LOGON: MON, JAN 6, 1986, 10:34 AM

CPU SECONDS: 19 CONNECT MINUTES: 123

\$STDIN LDEV: 23 \$STDLIST LDEV: 23

:PURGE MYDATAF

:ARCHIVE TRANSFER MY250 TO MYDATAF

HP 250 TO HP3000 ARCHIVE TRANSFER

RECORD 9 TRANSFERRED

END OF HP250 TO HP3000 FILE TRANSFER

TRNSFR/250 HEX to BINARY or ASCII Conversion Program.

ARCHIVE file "MYDATAF" with 1 records which are 256 words long.

The file "MYDATAF" was converted from HEX to BINARY.

END OF PROGRAM

:PURGE MYASCII

:DATA FILE TRANSFER MYDAT TO MYASCII

HP 250 TO HP3000 DATA FILE TRANSFER

RECORD 9 TRANSFERRED

END OF HP250 TO HP3000 FILE TRANSFER

TRNSFR/250 HEX to BINARY or ASCII Conversion Program.

ASCII file "MYASCII" with 9 records which are 160 Character long.

The file "MYASCII" was converted from HEX to ASCII.

END OF PROGRAM

:LISTF MY@

FILENAME

MYASCII MYDATA MYDATAF

:PURGE SCHEMA

:DATA FILE TRANSFER DBSCMA TO SCHEMA

HP 250 TO HP3000 DATA FILE TRANSFER

RECORD 9 TRANSFERRED
END OF HP250 TO HP3000 FILE TRANSFER

TRANSFR/250 HEX to BINARY or ASCII Conversion Program.

ASCII file "SCHEMA" with 9 records which are 160 Characters long.
The file "SCHEMA" was converted from HEX to ASCII.

END OF PROGRAM
:LISTF

FILENAME

MYASCII MYDATA MYDATAF SCHEMA

:



Appendix B

Messages and Their Meanings

This is the message appendix for the BASIC/260 to HP Business BASIC Conversion Utility. It contains the following message sets:

- Set 1: General messages for all aspects of the conversion program.
- Set 2: Warnings printed out during the conversion of programs.
- Set 3: Help facility listing.

SET 1: General Messages

These messages are used in all aspects of the conversion program. Some of these aspects include the following:

- Parsing the command line.
- Opening, reading, writing, closing, and purging files.

**** ERROR: Could not allocate enough memory. (1,1)

The program tried to get memory from the PASCAL Heap and could not. The command is aborted and the main prompt is returned to the user's screen.

HP 250 to HP Business BASIC/3000 Conversion Utility. (1,2)

This is the Banner message.

**** ERROR: Missing File name. (1,3)
**** ERROR: Missing Lock word. (1,3)
**** ERROR: Missing Group name. (1,3)
**** ERROR: Missing Account name. (1,3)
**** ERROR: File name is too long. (1,4)
**** ERROR: Lock word is too long. (1,4)
**** ERROR: Group name is too long. (1,4)
**** ERROR: Account name is too long. (1,5)
**** ERROR: File name does not begin with a letter. (1,5)
**** ERROR: Lock word does not begin with a letter. (1,5)
**** ERROR: Group name does not begin with a letter. (1,5)
**** ERROR: Account name does not begin with a letter. (1,5)

These messages are used to describe errors that occur in defining filenames.

**** ERROR: FORM has more than 26 single character fields. (1,10)

Forms can only have 26 fields that are only 1 character long. The command is aborted and the main prompt returns to the screen.

**** ERROR: Unrecognized command. (1,11)

The command you typed was not a recognized command. The only commands recognized are PROG, DATA, SCHEMA, FORM, DATASET, !, :, Help, and Exit.

**** ERROR: Could not deallocate memory. Program Aborted. (1,12)

The program tried to return memory to the PASCAL Heap and could not. The program is aborted and the MPE prompt returns to the screen.

<< IMAGE/250 to IMAGE/3000 Converted SCHEMA file "<filename>". (1,13)

This is the banner message for schema files.

Converted HP260 form "<filename>". (1,14)

This is the Banner message for form files.

**** ERROR: Data set entry lengths do not match. (1,15)

In this data set, the entry length in the data base does not match the entry length in the back up file. The command is aborted, and the main prompt returns to the screen.

Copyright Hewlett-Packard Co. 1985. (1,16)

This is the COPYRIGHT notice.

**** ERROR: File "<file name>" has records that are too big for the HP3000. (1,17)

The maximum record size on the HP3000 is 32,767 words long. Because of the 1.5 expansion, any file with an HP260 record size over 43,687 bytes long cannot be converted to be used on the HP3000.

**** ERROR: Could not run BBCTDATA.PUB.SYS, CREATEPROCESS error #<error number>. (1,20)

The CREATEPROCESS intrinsic failed. Refer to the *HP3000 Intrinsic Manual* for the error number. The command is aborted and the main prompt returns to the screen.

**** ERROR: MPE COMMAND failed with Error # <error number>. (1,25)

You tried to execute an MPE command (a colon followed by the command name) and an error occurred. This is the Command Interpreter ERRor (CIERR).

**** ERROR: Found unexpected characters at the end of the line. (1,26)

Extra characters were found after the last input character in the line. This message can also result from something in the line not being recognized.

Messages and Their Meanings

**** ERROR: You must specify the FROM file. (1,27)

The file to be converted was not specified.

**** ERROR: You must specify the TO file. (1,28)

The file where the converted output will go was not specified.

**** ERROR: You must specify the REPORT file. (1,29)

The file where conversion warnings will go was not specified.

**** ERROR: Unrecognized MPE command. (1,30)

You tried to execute an MPE command (a colon followed by the command) and the MPE command interpreter did not recognize the command. Neither UDC's nor the RUN command are allowed. Only the following MPE commands are allowed. (Refer to the *MPE Commands Reference Manual* and the *Console Operator's Guide* for descriptions of the commands.)

| | | | |
|-----------------|-------------|--------------|--------------|
| ABORTIO | GETLOG | PTAPE | SHOWALLOW |
| ABORTJOB | GETRIN | PURGE | SHOWCOM |
| ACCEPT | GIVE | PURGEACCT | SHOWDEV |
| ALLOW | HEADOFF | PURGEGROUP | SHOWIN |
| ALTACCT | HEADON | PURGEUSER | SHOWJCW |
| ALTGROUP | HELP | PURGEVSET | SHOWJOB |
| ALTJOB | IMLCONTROL | RECALL | SHOWLOG |
| ALTLOG | JOBFENCE | REFUSE | SHOWME |
| ALTSEC | JOBPRI | RELEASE | SHOWOUT |
| ALTSPoolFILE | JOBSECURITY | RELLOG | SHOWQ |
| ALTUSER | LDISMOUNT | REMOTE | SHOWTIME |
| ALTVSET | LIMIT | REMOTE HELLO | SPEED |
| ASSOCIATE | LISTACCT | RENAME | STARTSPOOL |
| BREAKJOB | LISTF | REPLY | STOPSPool |
| BUILD | LISTGROUP | REPORT | STORE |
| COMMENT | LISTLOG | RESET | STREAM |
| CONSOLE | LISTUSER | RESETACCT | STREAMS |
| DEALLOCATE | LISTVS | RESETDUMP | SUSPENDSPOOL |
| DELETESPOOLFILE | LMOUNT | RESTORE | SWITCHLOG |
| DISALLOW | LOG | RESUMEJOB | TAKE |
| DISASSOCIATE | MPLINE | RESUMELOG | TELL |
| DOWN | MRJCONTROL | RESUMESPOOL | TELLOP |
| DOWNLOAD | NEWACCT | SAVE | TUNE |
| DSCONTROL | NEWGROUP | SECURE | UP |
| DSLINE | NEWUSER | SETDUMP | VMOUNT |
| DSSTAT | NEWVSET | SETJCW | WARN |
| FILE | OUTFENCE | SETMSG | WELCOME |
| FOREIGN | | | |

File "<file name>" already exists. (1,32)

You tried to open a new file, and there is already a permanent file with that name in the system. This normally happens after the conversion, when the file is closed and added to the system. This message is used during a session. During a job, message #36 is used instead.

Do you want to purge old file, and save new file? (N/Y) (1,33)

If error message #32 occurs during a session and not a job, then this question displays. If you type Y or Yes, the old file is purged. If you type N, No, or press the carriage return, the new file is purged and the old file is saved. The main prompt returns to the screen.

**** ERROR: PURGE failed with system error # <error number>, and file error # <error number>. (1,34)

Messages and Their Meanings

If you typed Y in response to message #33, and the old file cannot be purged, this message is the result. In this case the new converted file is purged, the command is aborted, and the main prompt returns to the screen.

Trying to purge "<file name>". (1,35)

This message is printed when a file is being purged.

**** ERROR: File "<file name>" already exists. (1,36)

You tried to open a new file, and there is already a permanent file with that name in the system. This normally happens after the conversion, when the file is closed and added to the system. This message occurs during a job, otherwise message #32 is used. The command is aborted and the main prompt returns to the screen.

**** ERROR: PURGE of "<file name>" failed with system error # <error number>. (1,37)

This message displays if an error occurs during an attempt to purge a file. The command is aborted and the main prompt returns to the screen.

**** ERROR: File "<file name>" is not an HP260 PROG file. (1,39)
**** ERROR: File "<file name>" is not an HP260 SCHEMA file. (1,39)
**** ERROR: File "<file name>" is not an HP260 DATA file. (1,39)
**** ERROR: File "<file name>" is not an HP260 FORM file. (1,39)
**** ERROR: File "<file name>" is not an HP260 DATASET back up file. (1,39)

FORM, BDATA, and DATASET conversions require files transferred using the ARCHIVE TRANSFER softkey. BPROG and SCHEMA conversions require files transferred using the DATAFILE TRANSFER softkey. All files must be in ASCII format. The command is aborted and the main prompt returns to the screen.

Old file "<file name>" purged. (1,40)

The file was successfully purged.

**** ERROR: Purge of "<file name>" not confirmed. (1,41)

If message #33 does not get a Y as a response, this message is printed. In this case the new converted file is purged, the command is aborted, and the main prompt returns to the screen.

**** ERROR: Could not close file "<file name>", File Sys. error # <error number>. (1,42)

This message is printed if an error occurs during an attempt to close a file. The command is aborted and the main prompt returns to the screen.

**** ERROR: Could not open file "<file name>", File Sys. error # <error number>. (1,43)

This message is printed if an error occurs during an attempt to open a file. The command is aborted and the main prompt returns to the screen.

**** WARNING: MPE COMMAND Warning # <warning number>. (1,44)

You entered an MPE command (a colon followed by the command) and a warning occurred. This is the Command Interpreter WARNING (CIWARN).

Records read : <number of lines> (1,47)
 Number of warnings : <number of lines> (1,45)
 lines not converted : <number of lines> (1,46)

These messages are printed out at the end of a conversion. If <number of lines> is zero, "Number of warnings" and "lines not converted" are not printed.

**** WARNING: You need not specify the REPORT file. File ignored. (1,48)

The file where conversion warnings go was specified when it was not necessary. The REPORT file parameter is only used with the PROG and FORM commands. If used with any other command, the REPORT file is ignored.

**** ERROR: Could not open data base "<data base name>", Error # <error number>. (1,49)

This message is printed out if an error occurs during an attempt to execute the DBOPEN command. The command is then aborted and the main prompt returns to the screen. Refer to the specific error number in the *TurboIMAGE Reference Manual* for instructions.

**** ERROR: Could not close data base "<data base name>", Error # <error number>. (1,50)

This message is printed out if an error occurs during the execution of the DBCLOSE command. The command is then aborted and the main prompt returns to the screen. Refer to the specific error number in the *TurboIMAGE Reference Manual* for instructions.

**** ERROR: Could not get information on data base "<data base name>", Error # <error number>. (1,51)

This message is printed out if an error occurs during the execution of the DBINFO command. The command is then aborted and the main prompt returns to the screen. Refer to the specific error number in the *TurboIMAGE Reference Manual* for instructions.

**** ERROR: Could not add to data base "<data base name>", Error # <error number>. (1,52)

Messages and Their Meanings

This message is printed out if an error occurs during the execution of the DBPUT command. The command is then aborted and the main prompt is returned to the screen. Refer to the specific error number in the *TurboIMAGE Reference Manual* for instructions.

**** ERROR: HP260 FORM "<file name>" is too big. Error # <error number>. (1,53)

The HP260 form being converted has more than 12,000 characters in it. BBCT250 cannot convert forms larger than 12,000 characters. The command is aborted and the main prompt returns to the screen.

**** ERROR: Could not RUN FORMSPEC.PUB.SYS, CREATEPROCESS Error # <error number>. (1,54)

The CREATEPROCESS intrinsic failed. The command is aborted and the main prompt is returned to the screen. Refer to the specific error number in the *HP 3000 Intrinsic Manual* for instructions.

**** ERROR: Could not PURGE <file name>, system error #<error number>. (1,55)

**** ERROR: Could not PURGE <file name>, TEMP, system error #<error number>. (1,55)

**** ERROR: Could not RESET <file name>, system error #<error number>. (1,55)

**** ERROR: Could not FILE <file name>, system error #<error number>. (1,55)

The COMMAND intrinsic failed when the above statement was executed. The command is aborted and the main prompt returns to the screen.

SET 2: Program Conversion Warning Messages

These warnings are printed during the conversion of HP260 programs.

!***** WARNING There are only 8 keys in BASIC instead of 24 (as on the HP260) (2,1)

On the HP260 there were 24 softkeys, whereas, BASIC has only 8.

!***** WARNING The <function name> function is not supported in BASIC. (2,3)

The function mentioned is not supported in BASIC. You need to rewrite the code if you use this function. Any statement containing this function will be commented out when it is read into BASIC.

!***** Softkeys are different in BASIC than they were on the HP260. (2,4)

Softkeys can only be pressed during an INPUT statement.

!***** WARNING Forms are different in BASIC than they were on the HP260. (2,4)

This message appears when HP260 forms are converted to VPLUS forms. It is produced for each HP260 forms statement to remind you that more conversion is needed.

!***** WARNING CURSOR FNS are different in BASIC than they were on the HP260.
(2,4)

This message is generated when the cursor statement is converted with any of the following options: IV, BL, UL, HP, or RE. The automatic conversion performed may need further attention.

!***** WARNING The <statement name> statement is not supported in BASIC. (2,6)

The statement mentioned is not supported in BASIC. You need to rewrite the code if you use this statement. This line is commented out when it is read into BASIC.

!***** WARNING The <statement name> statement is not currently supported in BASIC.
(2,14)

The statement mentioned is not currently supported in BASIC. You need to rewrite the code if you use this statement. This line is commented out when it is read into BASIC.

!***** WARNING The <statement name> statement is not in BASIC, use the IMAGE Utilities instead. (2,15)

Messages and Their Meanings

The statement mentioned is not supported in BASIC. There are utilities available to handle these TurboIMAGE statements, and they are located in the group PUB.SYS. See the *IMAGE Data Base Management System Reference Manual* for a description of these utilities. This line will be commented out when it is read into BASIC.

!***** WARNING The <function name> function is not currently supported in BASIC.
(2,16)

The function mentioned is not currently supported in BASIC. You need to rewrite the code if you use this function. This line is commented out when it is read into BASIC.

!***** WARNING The line number is over 1000000. (2,17)

The line number 1000000 was used. The maximum line number allowed in BASIC is 999999. This line is not be added to the program. A line number that is too large may cause the GET command in BASIC to stop reading the program. Use a smaller line number in this line of the ASCII file.

!***** WARNING The line number is missing. (2,18)

There is no line number on the line. The line will not be added to the program. A missing line number may cause the GET command in BASIC to stop reading the program. You need to enter a line number in this line in the ASCII file.

!***** WARNING The format of file name is different on the HP3000. (2,19)

File names on the HP3000 have the following format:
<file__name> <slash> <lockword> <period> <group__name> <period> <account__name>.

!***** WARNING HP260 Volumes have been replaced by HP3000 groups and accounts.
(2,20)

Groups on the HP3000 have the following format:
<group__name> <period> <account__name>.

!***** WARNING Printers are now described by a string instead of a number. (2,21)

All devices on the HP3000 are referred to by file names. Refer to the *MPE Commands Reference Manual* for a description of the file naming conventions.

!***** WARNING The line was not fully converted because it would become too long.
(2,23)

The conversion utility tried to convert the old display control characters into escape sequences, and discovered that the resulting line would be longer than 252 ASCII characters (the size of the lines in the output file). The conversion utility returns the line to the state it was in before any display control characters were converted. The line is commented out even though it has the correct syntax.

SET 3: Help Facility

The HELP facility provides a summary of each command in the conversion package. This summary includes the syntax of each command. The following text is listed on the screen when you enter the HELP command.

The PROG command converts HP260 program files
 PROG <input file>, <output file>, <report file>

The DATA command converts HP260 data files.
 DATA <input file>, <output file>

The DATASET command converts and loads HP260 data bases. You will be asked for the password for adding entries to the data base. Your password will not be seen.
 DATASET <input file>, <data base>

The FORM command converts HP260 forms files.
 FORM <input file>, <output file>, <report file>

The SCHEMA command converts IMAGE/250 schema files.
 SCHEMA <input file>, <output file>

The HELP command provides the description and syntax of all the commands available in the conversion program.

The EXIT command is used to exit from the conversion utility. You can also type E, :: and :EOD.

The ! command precedes a comment (e.g. in a job stream). The ! cannot be typed in the first column. BBCT250 ignores all comments and simply places them in the output file.

Anything typed after a colon (:) will be executed as an MPE command. (The colon should be in the second column when used in a job stream.)

The CONVERT command determines what kind of file you are trying to convert and then converts it.

```
CONVERT <input file>, <output file>, <report file>
```

The OPTION INPUTLOOPS command converts all WAIT, INPUT, ACCEPT, and LINPUT statements into loops that use the RESPONSE built-in to EXIT LOOP.

The OPTION NOINPUTLOOPS command turns off OPTION INPUTLOOPS. OPTION NOINPUTLOOPS is the default.

The OPTION REAL command causes all REALs and SHORTs to be converted into REALs and SHORT REALs in the PROG, DATA, SCHEMA and DATASET commands.

```
OPTION REAL
```


Messages and Their Meanings

The OPTION DECIMAL command causes all REALs and SHORTs to be converted into DECIMALs and SHORT DECIMALs in the PROG, DATA, SCHEMA and DATASET commands. This is the default.

OPTION DECIMAL

The OPTION JOINFORM command will cause the FORM command to convert FORMS into JOINFORM FORMS.

OPTION JOINFORM

The OPTION VPLUS command will cause the FORM command to convert FORMS into VPLUS FORMS. This is the default.

OPTION VPLUS

Appendix C

Differences and Conversions

The information in this appendix is grouped into three categories. The first part describes differences between BASIC/260 and BASIC. The second part lists those BASIC/260 features that are not translatable in BASIC. The last part lists the automatic conversions when BASIC/260 program files are converted to BASIC using the conversion utility.

NOTE

BASIC features that are not part of BASIC/260 are not included in this appendix.

Differences

Many of the features in BASIC/260 are implemented in BASIC. However, in numerous cases, the implementation of a feature in BASIC is different and/or incompatible with the equivalent feature in BASIC/260. Those differences are described here.

Command Statement Only Allowed in Programs

In BASIC, the `COMMAND` statement is not allowed from the keyboard and is restricted to program lines. Nested `COMMAND` statements are not supported.

FOR-NEXT is Statically Matched

The `FOR-NEXT` loop is now a structure, not a pair of independent statements. `FOR`s and `NEXT`s must match before a program can be run. `FOR` and `NEXT` can no longer be included in single-line `IF-THEN` statements (e.g. `IF A>B THEN FOR I=1 TO 5`, or `IF A>B THEN NEXT I`). This was changed for performance considerations, particularly in compiled programs.

CALL Line is Remembered for Return

When a `CALL` is executed, the line remembered is the `CALL` line, not the line after the `CALL` (as on the HP260). This can be observed by adding a statement immediately after the `CALL` while the `CALL` is active. The new statement is executed upon returning from the subprogram.

Common Areas

In BASIC, original common areas define the maximum size allowed for a specified common name during a program run. Otherwise stated, a common area may not grow across "chains" (programmatic GETs). BASIC/260 allows growth.

The MERGE Statement

In a BASIC program, the MERGE statement does not execute a line that replaces it, but executes the following line. This applies only when no explicit execution line is given. BASIC/260 executes the line that replaced the MERGE statement.

Programmatic GETs

In BASIC, the default place to begin executing a new program that results from a programmatic GET statement is always the first line. This is true even if the GET appears in the main program unit and there are lines before it that are not deleted. BASIC/260 executes the line after the GET (or the line that replaced the GET) if the GET is in the main program unit.

Memory Management

Memory management for programs and data is very different between BASIC/260 and BASIC. The HP260 provides a total of 64kb per user for data and the compact representation of the program. LOAD SUB and DEL SUB are used to manage a 64kb partition. In BASIC, the compact representation for non-active program units (as well as all comments and DATA strings) are paged out to extra data segments. This enables large programs to be run if they are divided into small (several hundred line) subunits. The data space available to the user is not paged and will be a maximum of about 30kb in the Interpreter.

Display Control Characters

The display control characters available on the HP260 are not displayable in BASIC, even when a special terminal is used. They are turned into escape sequences.

File Names

File names under MPE are very restricted relative to the HP260. They must consist of a single letter followed by a combination of up to seven letters or digits. The case of the letters is insignificant. No special characters are allowed in file names.

| |
|-------------|
| NOTE |
|-------------|

For easier conversion, change any application that uses special characters in file names or that depends on the case of the letters in file names being significant.

Volume Specifiers

The HP3000 does not generally use the physical disc number or device name as part of the file designator. MPE manages the placement of files on the various disc drives. Instead, the files are organized into groups and accounts. Two files may have the same name but they can be distinguished by residing in different groups and accounts. The BASIC/260 MSI statement should be translated into a FILES ARE IN statement in BASIC. This statement specifies the group and account that should be searched for file references that do not have explicit group and account names.

| |
|-------------|
| NOTE |
|-------------|

For easier conversion, use the MSI statement instead of specifying volume designators as part of the file reference whenever possible. This reduces the number of places that require manual conversion.

Printer Specifications

Printer specifications are quite different between BASIC/260 and BASIC. Two different kinds of changes have been made. One change is the simple substitution of the keywords SEND OUTPUT TO (BASIC) for the keywords PRINTER IS (BASIC/260). This change is handled by the conversion process.

The other change is more significant. The printer destination in BASIC/260 is expressed as a numeric expression. In BASIC the destination is either a keyword, such as DISPLAY, or it is a string expression (e.g. *LP, *PP) that is a file designator. For example, the BASIC/260 statement

PRINTER IS 8

is converted to the following statement in BASIC:

SEND OUTPUT TO DISPLAY

BASIC performs the following automatic conversions of printer destinations:

| <u>BASIC/260 Device #</u> | <u>BASIC Keyword</u> |
|---------------------------|----------------------|
| 0, 1 | PRINTER |
| 8 | DISPLAY |
| 9 | NULL |

Converted BASIC/260 statements containing any other device number are not syntaxed by BASIC.

IMAGE Data Base Management

BASIC does not implement IMAGE/260, but does provide an interface to TurboIMAGE that is functionally similar to the interface on the HP260. The following are differences between TurboIMAGE and IMAGE/260:

1. IMAGE/260 supports read locking. TurboIMAGE does not. A BASIC/260 program is converted to perform a write lock. This may degrade the performance of some applications.
2. DBCLOSE with mode equal to 4 on the HP260 causes buffers to be flushed. There is no corresponding mode for DBCLOSE in TurboIMAGE. The TurboIMAGE intrinsic DBCONTROL may be used to flush IMAGE buffers.
3. IMAGE/260 offers the user the ability to control the placement of data in physical locations. There is no analogous feature in TurboIMAGE. Associated with this, the DBINFO statement of IMAGE/260 supports a 400 class of inquiries about volumes that is not supported by TurboIMAGE. There is also a conflict in that TurboIMAGE uses mode 401 to get information on logging.
4. The capacities in TurboIMAGE are larger than the capacities under IMAGE/260. In some places IMAGE/260 uses a single word integer, while TurboIMAGE uses a double word integer. Therefore, several words in the IMAGE/260 status array are specified to be zero, and may be non-zero under TurboIMAGE. Also, there are places where information is passed in a status word that is always zero in IMAGE/260. In these cases, the TurboIMAGE information is passed through and the word is non-zero.
5. If an IMAGE statement is executed on a line with a line number greater than 32767, then Status(6) contains a negative number. If the line number is greater than 65535, then Status(6) is -1. No converted BASIC/260 program has a line number greater than 9999.
6. IMAGE/260 allows several functions to be invoked programmatically on the HP260 that must be performed through a utility program on the HP3000. Creating a data base (DBCREATE), changing a password (DBPASS) and copying data sets (XCOPY) can all be done with BASIC language statement in BASIC/260. These statements are not supported in BASIC. The equivalent functions must be done interactively through the DBUTIL utility on the HP3000. In addition, the utilities DBERASE, DBSTORE, DBRESTORE, DBPASS, and DBMAINT fall into this category.
7. IMAGE/260 avoids deadlock if more than one data base is used. On the HP3000, multiple RIN capability is required to do locks on multiple data bases and TurboIMAGE does not prevent deadlock. A converted BASIC/260 is therefore subject to the possibility of deadlock if multiple data bases are used.
8. IMAGE/260 allows an arbitrary small integer to be associated with each data item in the root file. This number is called the format number or control number and the meaning of the number is determined by the user's program. This feature is not part of BASIC. There is a DBINFO call in IMAGE/260 that allows a program to examine format numbers. There is no corresponding DBINFO call in TurboIMAGE.
9. There is no direct support in TurboIMAGE for the floating point decimal data types available in BASIC/260 and BASIC. BASIC/260's REAL and SHORT types (which map to DECIMAL and SHORT DECIMAL in BASIC) are represented in the converted schema as K2 and K4. This simply implies that the data is either 2-word binary or 4-word binary. When accessing an TurboIMAGE data base, BASIC recognizes that these data types indicate floating point decimal numbers and processes the numbers accordingly.

10. IMAGE/260 allows data items of type STRING (specifier = x) to be up to 1022 characters long. TurboIMAGE restricts string item length to 254 characters. In both cases, string length must be even. IMAGE/260 allows 512 sub-items, while TurboIMAGE allows only 255.
11. ON KEY Behavior:
- There are only eight softkeys, instead of 24. During conversion constant references to keys above 8 are removed from ON KEY and OFF KEY statements.
 - Pressing a softkey is recognized at any point in a program on the HP260. On the HP3000, pressing a softkey can only be recognized while the terminal is being read (for example, in an input statement).
 - Pressing a softkey on the HP3000 causes a CR-LF to be echoed locally to the terminal screen. This causes the cursor to move. On the HP260, pressing a softkey does not move the cursor.
 - In BASIC, when a softkey is pressed, it results in the termination of the currently executing input statement. On the HP260, this is only done when an ON KEY statement is executed.
 - TFNUM on the HP260 returns the field number that the cursor is in. TFLD (JOINFORM) in BASIC returns the field number that input is to come from; the TAB key has no effect on it.
 - Softkey labels on the HP260 may contain display enhancements. This is not true for BASIC.
 - Softkey labels on the HP260 are centered automatically. The centering may be different in BASIC.
 - Labels on the HP250 are nine characters wide; they are eight characters wide in BASIC.
12. VPLUS Forms
- When a VPLUS form is active, the HALT key is not available.
 - A VPLUS form allows only 128 fields; FORMS/260 allows more.
13. Report Writer
- The PAGE LENGTH default is now 60 instead of 66.
 - The number of blank lines on the top and bottom of a REPORT WRITER page defaults to zero instead of two.
 - On the HP260, the PRINT PAGE statement puts out one blank line. It does not affect the page count, nor does it reset the line count. In BASIC, PRINT PAGE is equivalent to executing the following statement:
- TRIGGER PAGE BREAK,SUPPRESS HEADER,TRAILER
- The page header and page trailer are not printed, but the page count and line count are affected.

The BASIC Compiler

In addition to an interpreter (the type of implementation that exists on the HP260), BASIC provides the ability to compile programs. The BASIC statements are translated into HP3000 machine code instructions that can be run as any other program (e.g. COBOL, Pascal) in the system. Performance improvements for most applications are substantial. Certain I/O-bound applications, however, will show little improvement. Because BASIC/260 was implemented strictly as an interpreter, a number of features were introduced into that language that are not compilable. The execution of these statements requires information that is not available until run-time. Machine code cannot be generated at compile-time to anticipate the run-time conditions. The following statements generate compile-time warnings and run-time errors, and therefore, cannot be compiled. It is recommended that production applications be modified to eliminate these statements.

| | |
|---------|---------|
| COMMAND | RE-SAVE |
| DEFAULT | PAUSE |
| DEL | SAVE |
| LINK | SCRATCH |
| GET | SECURE |
| MERGE | STORE |
| NORMAL | TRACE |

These features are available as interpreter commands. Subprograms can be compiled and later CALLED from interpreted programs. It is therefore possible just to eliminate these statements from low-level routines and compile them, resulting in faster execution of the interpreted main program. Compiled programs also result in a much lower demand on operating system resources (memory, disc, etc.).

Untranslatable Code

These paragraphs list the HP260 statements and functions that are not implemented in BASIC. When the program BBCT250 encounters one of these statements or functions, it does not attempt to modify the program code, but appends the comment: "!! UNTRANSLATABLE" to the program line. This section groups the untranslatable features according to their function on the HP260.

Forms

The CFORM/250, MFORM/250, and PFORM/250 programs are not supported. If you use HP3000 style forms (VPLUS based) use FORMSPEC.PUB.SYS instead. If you use JOINFORMS then the utility that you should use is JFEDIT.PUB.SYS.

Data Base Utilities

The following statements perform utility functions on IMAGE data base files. These functions may not be performed in BASIC. MPE provides the utility programs: DBUTIL.PUB.SYS, DBSTORE.PUB.SYS, and DBRESTORE.PUB.SYS to accomplish the same functions. Refer to the *TurboIMAGE Reference Manual* for more information.

| | |
|----------|------------------|
| DBCREATE | DBRESTORE |
| DBERASE | DBSTORE |
| DBMAINT | READ DBPASSWORD |
| DBPASS | WRITE DBPASSWORD |
| DBPURGE | |

Advanced Data Base Statements

The IN DATASET statement no longer supports the following features:

DIM ALL
 FREE
 IN COM
 USE ALL
 USE REMOTE LIST

The PREDICATE statement with a string relation is not translatable.

TIO statements

The following statements are part of TIO/250. Some of these capabilities are available through MPE operating system services. The BREAK ON/OFF and ECHO ON/OFF statements can be replaced with calls to the MPE Intrinsic FCONTROL. See the MPE Intrinsic manual for more information.

AREAD\$
 BLOCK MODE ON #
 BLOCK MODE OFF #
 BREAK ON / OFF
 ECHO ON / OFF
 ON / OFF BREAK #
 ON / OFF CONNECT #
 ON / OFF DISCONNECT #
 ON / OFF INPUT #
 ON / OFF OUTPUT #
 ON / OFF TRIGGER #
 SEND #
 SEND BREAK #

System dependent

The following statements refer to hardware or software features of the HP260 which are not applicable to the HP3000.

| | |
|--------------------|-------------------|
| ATTACH # | PERFORM |
| BUFFER # | PRINT LABEL |
| CATFILE | READ LABEL |
| CATLINE | RELEASE # |
| CHECK READ OFF # | REQUEST # |
| CHECK READ ON # | RE-STORE BIN |
| DETACH | SET DATE TO |
| DIRECT | SET TIME TO |
| DOOR LOCK / UNLOCK | SPACE DEPENDENT |
| DUPTST | SPACE INDEPENDENT |
| INDIRECT | STORE BIN |
| LOAD BIN | |

Media

The following statements provide facilities for reading and writing IBM diskette media. BASIC does not provide facilities to do this. There is a utility (DISCCOPY/3000) which can transfer data from an IBM floppy to the HP3000. See the Flexible Disccopy/3000 manual for details.

```
ASSIGN ;EBCDIC
ASSIGN ;EBCDIK
CREATE ;CHAR
DELETE #
DUPLICATE
IBMDUMP
IBMWREC
LINPUT #
```

Type SHORT Numbers

On the HP260, numbers of type SHORT took up 2 words (see the *250 BASIC Programming Manual* page 6-47) in a file. In BASIC, numbers of type SHORT will take up 3 words (1 word of a descriptor, and 2 words for data). Because of this, all converted files will have a record size that is 1.5 times that of the 250 file. The excess space will be filled with <end of records>. This allows a 250 file that was full of SHORT numbers to fit in the BASIC file. This also has the side effect that numbers and strings in the file may be moved because there was a SHORT number before them. As a result, you may need to hand convert portions of your program if you use word positioning, the WRD function, or depend on the length of the records in the file. Any DATA files with a record length over 43,687 bytes long cannot be converted to BASIC.

Miscellaneous Statements and Functions

The following miscellaneous statements and functions are also not implemented in BASIC. The EDIT statement should be replaced with INPUT, TINPUT, LINPUT or ACCEPT. The statement LINPUT # should be replaced with READ #.

```
DET (without parameters)
EDIT
EXIT IF (not allowed in a single-line IF... THEN statement)
ON / OFF DELAY
RES
```

The CURSOR statement no longer supports the following items:

```
PALL
UPALL
PL
UP
IF
OF
RIF
ROF
```

LIST KEY and EDIT KEY are no longer supported. Use your terminal's function key editing features instead. KEY files are not converted.

Automatic Conversions

This section contains a table listing the conversions that occur when BBCT250 converts BASIC/260 program files to BASIC.

The left side of the table lists the BASIC/260 statement name or feature, and the right side lists the equivalent BASIC statement or feature.

To successfully use the table below, you should be aware of the following:

- An asterisk (*) before the statement name indicates there are some additional syntax changes resulting from the conversion. These changes primarily involve converting from a positional notation to a keyword notation and/or removing unnecessary parameters from the statement.
- A plus (+) before the statement name indicates that the phrase IF INTERPRETED THEN is added to the beginning of the statement. For example:

| <u>BASIC/260</u> | <u>BASIC</u> |
|------------------|-------------------------------|
| SAVE | IF INTERPRETED THEN SAVE LIST |

When a program is converted, the following line is automatically added to the beginning of the program:

```
1 GLOBAL OPTION BASE 0,DECIMAL  !*** ADDED LINE
```

All line numbers are multiplied by 10 during automatic conversion.

Table C-1. Automatic Conversions

| BASIC/260 | BASIC |
|--|--|
| DISPLAY CONTROL CHARACTERS PROUND() & EXOR BINIOR BINEOR INTEGER SHORT REAL *DEL *FCREATE XCOPY COPY *FIND RE-SAVE RE-STORE SAVE STORE LOAD SCRATCH SCRATCH P SCRATCH V SCRATCH C BIT SKP DEFAULT FCREATE LINK MERGE | ESCAPE SEQUENCES ROUND() + XOR BINEOR BINXOR SHORT INTEGER SHORT DECIMAL DECIMAL +DEL CREATE COPYFILE COPYFILE SEARCH +RESAVE LIST +RESAVE +SAVE LIST +SAVE +GET +SCRATCH PROG +SCRATCH PROG +SCRATCH VARS +SCRATCH COMMON BITRL SKIP +DEFAULT CREATE +LINK +MERGE |

Table C-1. Automatic Conversions (continued)

| BASIC/260 | BASIC |
|---|---|
| *CURSOR OF# CF# IF# UL(n) IV(n) HB(n) BL(n) LOAD KEY STORE KEY RE-STORE KEY# SCRATCH KEY# *ON KEY# OFF KEY# XPOS YPOS NUMPAGE= NUMPAGE FIND DELETE FORM EXIT FORM GET FORM <file name> <line number> WFLN TFNUM LAST BREAK SORT *PAGE LENGTH | CURSOR OFLD() CFLD() IFLD() ("U",n) ("I",n) ("H",n) ("B",n) GET KEY SAVE KEY RESAVE KEY SCRATCH KEY ON KEY OFF KEY CPOS RPOS SET PAGENUM TO PAGENUM SEARCH USING ! CLOSE FORM CLOSE FORM;REMAIN OPEN FORM <file name> +":"+ <file name> <line number>0 NUMREC TFLD() LASTBREAK SORT USING ! (syntax change only) |

Table C-1. Automatic Conversions (continued)

| BASIC/260 | BASIC |
|--|---|
| SCRATCH A RUN-ONLY MSI MASS STORAGE IS *SYSTEM PRINTER IS *PRINTER IS *PRINT ALL IS NORMAL *TRACE *TRACE ALL VARIABLES TRACE VARIABLES A=B=C=1 milliseconds *DBCLOSE *DBDELETE *COM *ASSIGN *CREATE *CAT *DBFIND *DBGET *DBINFO *DBLOCK *DBOPEN *DBPUT *DBUNLOCK *DBUPDATE *DATA *IMAGE AVAIL HOLE Array (*) AND OR TYP(N) WRD(N) REC(N) SLEN(N) SECURE COMMAND *PREDICATE | +SCRATCH ALL RUNONLY FILES ARE IN FILES ARE IN SEND SYSTEM OUTPUT TO SEND OUTPUT TO COPY ALL OUTPUT TO + TRACE OFF + TRACE EXEC + TRACE EXEC VARS or TRACE VARS IN ALL + TRACE VARS A,B,C=1 ((seconds)/1000) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) (syntax change only) trailing comments are removed trailing comments are removed 32767 32767 Array () LAND LOR TYP(ABS(N)) WRD(ABS(N)) REC(ABS(N)) SLEN(ABS(N)) +SECURE +COMMAND (syntax change only) |

Table C-1. Automatic Conversions (continued)

| BASIC/260 | BASIC |
|--|--|
| 10 WORKFILE #n;THREAD IS... | 100 WORKFILE IS #n 101 POSITION #n;RESET ***ADDED LINE 102 THREAD IS !... ***ADDED LINE |
| 10 WAIT | 100 LOOP *** ADDED LINE 101 ACCEPT *** SYNTAX CHANGE 102 EXIT IF RESPONSE=1 *** ADDED LINE 103 ENDLOOP *** ADDED LINE |
| Under option NOINPUTLOOPS: 10 INPUT 10 ACCEPT 10 LINPUT | 100 INPUT 100 ACCEPT 100 LINPUT |
| Under option INPUTLOOPS: 10 INPUT | 100 LOOP 101 INPUT ***ADDED LINE 102 EXIT IF RESPONSE>0 ***ADDED LINE 103 ENDLOOP ***ADDED LINE |
| 10 ACCEPT | 100 LOOP ***ADDED LINE 101 ACCEPT 102 EXIT IF RESPONSE>0 ***ADDED LINE 103 ENDLOOP ***ADDED LINE |
| 10 LINPUT | 100 LOOP ***ADDED LINE 101 LINPUT 102 EXIT IF RESPONSE>0 ***ADDED LINE 103 ENDLOOP ***ADDED LINE |



Appendix D

Using PERFORM and DSN/DS

The DSN/DS product is not supplied with BASIC. DSN/DS is purchased separately. However, DSN/DS provides the fastest means available for transferring files from the HP260 to the HP3000. Refer to the *DSN/DS 260 Manual* for complete information on how to configure the HP260 and the HP3000 for use with this program, and for details about transferring files with DSN/DS.

In order to automate file transfers between the HP260 and the HP3000, it is also necessary to have the PERFORM DROM on the HP260. Refer to Chapter 12 in the *BASIC/260 Programming Manual* for complete information on PERFORM. This appendix provides an example of how to use PERFORM with DSN/DS in order to transfer files automatically from the HP260 to the HP3000.

The PERFORM File

Before issuing the PERFORM statement on the HP260, you must create a "perform file" containing the commands and user input that will be processed in Perform Mode. The perform file must be a type "DATA" file containing ASCII string data with records no longer than 160 characters. When in Perform Mode, the statements in the perform file will be executed one at a time, sequentially. When used with DSN/DS the perform file can contain HP260 commands, MPE commands, BASIC/260 statements, user input or prompts from programs, and special perform commands. The perform commands are preceded by a colon. (MPE commands should NOT be preceded by colons.) To run in completely automated mode, the "UNATTENDED" statement must be specified at the beginning of the perform file.

Sample PERFORM File

The next two pages contain an example of a perform file set up to transfer a schema file and an IMAGE database backup file from the HP260 to the HP3000. The perform file also includes the MPE commands needed to convert the files using the automatic conversion utility (BBCT250.PUB.SYS), and the execute the utilities required to build the database on the HP3000. Repeat lines 16 through 32 in the perform file (changing the file names) to send another binary file (a forms file, data set backup file or a BASIC/260-formatted data file) to HP3000. Repeat lines 34 to 62 (changing file names) to send another ASCII file (a schema file, program source code, or documentation file). If the perform file is named "perfcom", it would be executed with the command: PERFORM perfcom.

```

1  :!   PERFORM routine for DS/260 file transfer to HP3000
2  :!
3  :!   Note: to get a hardcopy output from BBCT250, set
4         up a file equation before invoking the program, e.g.
5         FILE LP;DEV=LP
6         and back reference LP in the BBCT250 commands
7  :!
8  UNATTENDED
9  :!
10 RUN "DS:S"
11   "DSFIG:S "
12 :!
13 :!   Logon to the HP3000
14 HELLO WESLEY.SAWYER
15 :!
16 :!   Hit "TRANSFER TO HP3000" key
17 :KEY #6
18 :!
19 :!   Hit "ARCHIVE TRANSFER" key
20 :KEY #2
21 :!
22 :!   Enter HP260 source file name
23 EMPBCK
24 :!
25 :!   Enter HP3000 destination file name
26 EMPBCKA
27 :!
28 :!   Hit "CREATE PERM FILE" key
29 :KEY #1

```

```
30 :!  
31 :! Hit "BEGIN TRANSFER" key (File will be transferred here)  
32 :KEY #1  
33 :!  
34 :! Hit "TRANSFER TO HP3000" key (Set up another file transfer)  
35 :KEY #6  
36 :!  
37 :! Hit "DATA FILE TRANSFER" key  
38 :KEY #4  
39 :!  
40 :! Enter HP260 source file name  
41 EMPSCH  
42 :!  
43 :! Enter record type: Variable Length (always specify variable!)  
44 V  
45 :!  
46 :! Enter HP3000 destination file name  
47 EMPSCHD  
48 :!  
49 :! Enter record type: Fixed Length (faster than variable)  
50 F  
51 :!  
52 :! Enter number of bytes/record: 256  
53 256  
54 :!  
55 :! Enter maximum number of records: numeric value  
56 1000  
57 :!  
58 :! Hit "CREATE PERM FILE" key  
59 :KEY #1  
60 :!  
61 :! Hit "BEGIN TRANSFER" key (File will be transferred here)  
62 :KEY #1  
63 :!  
64 :! Run conversion program on the HP3000  
65 RUN BBCT250.PUB.SYS  
66 :!  
67 :! Convert schema file  
68 SCHEMA EMPSCHD,EMPSCHEM  
69 :!  
70 :! Exit from BBCT250  
71 EXIT  
72 :!  
73 :! Create root file from schema on HP3000  
74 FILE DBSTEXT=EMPSCHEM  
75 RUN DBSCHEMA.PUB.SYS;PARAM=1  
76 :!  
77 :! Create data set files on HP3000  
78 RUN DBUTIL.PUB.SYS  
79 CREATE EMPLOY  
80 EXIT  
81 :!  
82 :! Run conversion program on the HP3000  
83 RUN BBCT250.PUB.SYS
```

Using PERFORM and DSN/DS

```
84 :!  
85 :! Convert and load data into new database  
86 DATASET EMPBCKA,EMPLOY  
87 KING  
88 :!  
89 :! Exit from BBCT250  
90 EXIT  
91 :!  
92 :! Logoff the HP3000  
93 BYE  
94 :!  
95 :! Exit DS/250  
96 :KEY #8  
97 :!  
98 :! End of PERFORM procedure  
99:END
```

A

- additional commands, 2-5
 - EXIT, 2-5
 - HELP, 2-5
 - OPTION DECIMAL, 2-6
 - OPTION INPUTLOOPS, 2-5
 - OPTION JOINFORM, 2-6
 - OPTION NOINPUTLOOPS, 2-6
 - OPTION REAL, 2-6
 - OPTION VPLUS, 2-6
- !(comment), 2-5
- automatic conversions, C-9
- ACCEPT statement, 3-4

C

- commands to convert files, 2-3
 - CONVERT, 2-5
 - DATA, 2-4
 - PROG, 2-4
 - SCHEMA, 2-4
- conversion process description, 2-1
- conversion process overview, 1-2
- converting files to BASIC, 3-1
 - ARCHIVE TRANSFER softkey, 3-1
 - data bases, 3-7
 - data files, 3-5
 - DATA FILE TRANSFER softkey, 3-1
 - forms, 3-10
 - programs, 3-1
 - PROG command, 3-1
 - softkeys, 3-4
- CONVERT command, 3-11

D

- data bases, 3-7
- data files, 3-5
- differences and conversions in BASIC/260 & BASIC, C-1
- differences between BASIC/260 & BASIC
 - common areas, C-2
 - CALL line, C-1
 - COMMAND statement, C-1
 - file names, C-2
 - IMAGE, C-4
 - memory management, C-2
 - MERGE statement, C-2

printer specifications, C-3
programmatic GETs, C-2
the compiler, C-6
untranslatable code, C-6
volume specifiers, C-3

F

forms conversion, 3-10
FORMS/260 to JOINFORM, 4-1
FORMS/260 vs. VPLUS/3000, 3-10
general information, 3-10
procedure, 3-10

G

general messages, B-2

H

HELP facility, B-11
CONVERT command, B-11
DATA command, B-11
DATASET command, B-11
EXIT command, B-11
FORM command, B-11
HELP command, B-11
OPTION DECIMAL command, B-12
OPTION INPUTLOOPS command, B-11
OPTION JOINFORM command, B-12
OPTION NOINPUTLOOPS command, B-11
OPTION REAL command, B-11
OPTION VPLUS command, B-12
PROG command, B-11
SCHEMA command, B-11
! command, B-11

I

INPUT statement, 3-4

J

JOINFORM conversions table, 4-1
JOINFORM forms, 3-10

L

LINPUT statement, 3-4

M

messages and their meanings, B-1

O

OPTION INPUTLOOPS, 3-4

OPTION NO INPUT LOOPS, 3-4

P

program conversion warning messages, B-9

**R**

READ FORM statement, 3-4

RESPONSE function, 3-4

T

transferring files, 2-1

DSN/DS, 2-1

TRNSFR, 2-1

TINPUT statement, 3-4

TRNSFR, A-1

considerations when using, A-1

U

using a STREAM file, 2-3

using PERFORM and DSN/DS, D-1

using the conversion program, 2-3

using TRNSFR, A-1

archive transfer, A-4

batch file transfer, A-7

data file transfer, A-6

getting started, A-2

V

VPLUS forms, 3-10

W

WAIT statement, 3-5

READER COMMENT SHEET

HP Computers

**BASIC/260 to HP Business BASIC
Conversion Guide**

32115-90005 August 1986

We welcome your evaluation of this manual. Your comments and suggestions help us to improve our publications. Please explain your answers under Comments, below, and use additional pages if necessary.

Is this manual technically accurate? Yes No

Are the concepts and wording easy to understand? Yes No

Is the format of this manual convenient in size, arrangement, and readability? Yes No

Comments:

This form requires no postage stamp if mailed in the U.S. For locations outside the U.S., your local HP representative will ensure that your comments are forwarded.

FROM:

Date _____

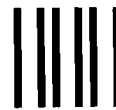
Name

Company

Address

FOLD

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



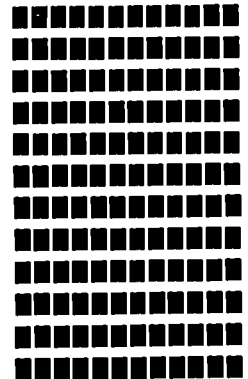
BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1070 CUPERTINO, CALIFORNIA



POSTAGE WILL BE PAID BY ADDRESSEE

Publications Manager
Hewlett-Packard Company
ITG Languages-Marketing
19111 Pruneridge Avenue
Cupertino, California 95014



FOLD

FOLD