HP 260 to Eloquence Migration Guide

B.06.32

Edition E1202

Legal Notices

Restricted Rights Legend

Acknowledgments

Marxmeier Software AG
Besenbruchstrasse 9
42285 Wuppertal
Germany

Printing History

The manual printing date indicates its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. New editions are complete revisions of the manual.The dates on the title page change only when a new edition or a new update is published.

Manual updates may be issued between editions to correct errors or document product changes. Manuals that are published on the Eloquence website (www.hp-eloquence.com/doc) may be updated more often, please visit this website periodically for the most recent versions. To ensure that you receive the updated or new editions, you should also subscribe to the appropriate product support service.

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	Mar 1990	A.01.00
Second Edition	July 1991	A.03.00
Third Edition	January 1997	A.06.00
Fourth Edition	October 1997	A.06.00
Fifth Edition (E1202)	December 2002	B.06.32

Printed in the Federal Republic of Germany.

Printing History

Contents

1

Introduction

This guide describes how to move HP 260 applications from an HP 260 computer to an UNIX system.

In order to successfully use this guide, you should be familiar with how to use the HP 260. This includes programming and system management of the HP 260. Additionally, you should know how to use the UNIX system and the operating system. This includes system administration knowledge. You should also have had some experience with Eloquence.

In this manual we explain the migration as it works on the HP9000 with HP-UX.

Useful References

To help you with your task we recommend the following manuals:

- HP 260 Installation Manuals.
- Eloquence Manuals.
- HP-UX Reference Volumes 1 through 3.
- *HP 9000: System Administrator Manual*.
- *HP 9000: Concepts and Tutorials: UUCP*.

Further, the use of the HP-UX man command as online-help is recommended.

Example:

```
man cp
```

This causes an explanation of the cp (copy) command to appear on the screen.

Conventions

The following conventions are used throughout this manual:

- **Bold type** is used when a new term is introduced.
- `Computer font` indicates text to be input exactly as shown or text that is output from the system.
- *Italic type* is used for emphasis and titles of publications. It is also used to indicate parameters that are user defined.
- KEYCAP represents a key on the keyboard.
- Shading represents the softkeys displayed on the computer screen.
- …indicates that the previous variable can be repeated.
- [ ] indicates that information inside the brackets is optional. If there are brackets within brackets, the information within the inner bracket may only be specified if the information in the outer bracket is specified. Information may also be stacked in brackets. For example, A or B or neither may be selected when the following is shown:

$$\begin{bmatrix} A \\ B \end{bmatrix}$$

- { } indicates that one of the choices stacked within the braces must be selected. For example, A or B or C must be selected when the following is shown:

$$\begin{Bmatrix} A \\ B \\ C \end{Bmatrix}$$

NOTE: Notes contain important information that is set off from the text.

Prerequisites

An HP 260 with at least the following:

- 1 Terminal (for example, HP 700/92).
- Optional: 1 Printer (for example, HP 2235B).
- 1 available RS-232C interface which can be configured as a workstation.
- 1 tape drive (for example, HP 9144A).
- One or more fixed disks (moveable) containing the application(s) to be migrated, or one or more cartridge tapes containing the application(s) to be migrated.
- HP260 Operating system B.09.02.

An HP 9000 with at least the following:

- 1 Terminal (for example, HP 700/92).
- Optional: 1 Printer (for example, HP 2235B).
- 1 available RS-232C interface.
- 1 tape drive (for example, HP 9144A).
- One or more fixed disks with space for your application(s). As a rule of thumb, during migration the space allowed for your application should be at least double what the application takes up on the HP 260. This is because at one point during the migration process, you will have the HP 260 and Eloquence versions of the application along with the export files on the hard disk.
- Operating System HP-UX 9.x or more recent version.
- Eloquence Runtime or Development version B.06.32 or more recent version, installed on the HP 9000.

In order to link the two systems via the RS-232C interfaces, you should use the following pin arrangement:

HP 260	HP 9000 Series 800
2	3
3	2
7	7
20	20

Note that pins 2 and 3 *must* be crossed.

The Migration Tools

The migration tools are part of the Development Version of Eloquence. They consist of four files, which are located in directory `/opt/eloquence6/tools`.

The files are:

ctd260	file transfer.
d.260stmt	default specification file for the stmtcheck program.
hp260exp	database conversion.
stmtcheck	application analysis.

2

Migrating HP 260 Applications

After all the preparations have been made you can begin with the actual migration. The migration basically consists of five procedures, as shown below:

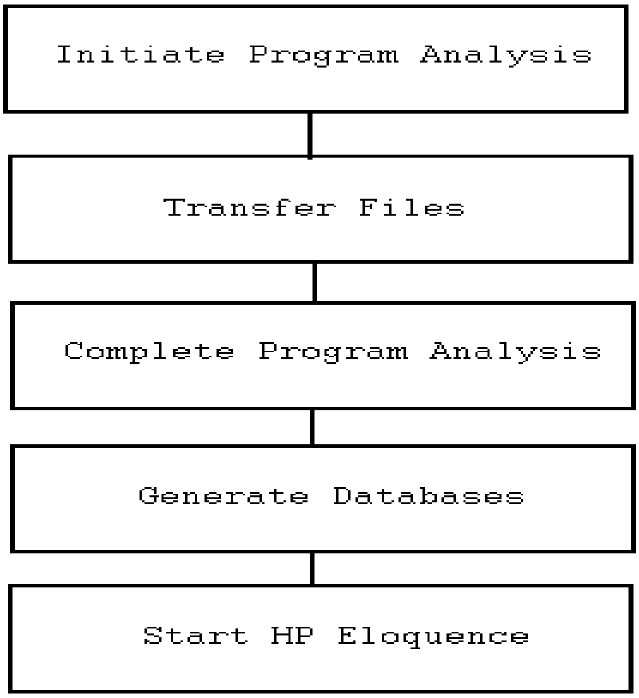

Figure 1 **Migration Schema**

Initiating HP 260 Program Analysis

The process of program analysis refers to checking HP 260 program (.PROG) files for the following:

- Binary code.
- Statements not implemented in Eloquence.
- Statements that work differently in the Eloquence environment.

Checking HP 260 programs for binary code is done on the HP 260 using the BINCHK program, while the remaining part of the analysis process is performed on the HP 9000 Series 800. For this reason, the discussion pertaining to program analysis is divided into two sections within this chapter.

This section discusses what must be done on the HP 260 side. page 21 discusses what must be done on the HP 9000 Series 800 after the HP 260 files have been transferred.

NOTE: The process of checking for statements not implemented or that work differently in Eloquence can be done on the HP 260 instead of the HP 9000 Series 800. This is done by using the HP 260 KWDS utility. This utility is normally used by those considering migrating to Eloquence. The information produced can be useful in determining the amount of work necessary to convert HP 260 program files to Eloquence program files. Documentation accompanies the KWDS utility software.

BINCHK

This procedure is only for HP 260 program files (files of type .PROG).

Eloquence is not able to process HP 260 binary programs. It can only execute HP 260 BASIC intermediate code. For this reason a check must be made to find out what application programs contain binary code. To do this, start the binary test program on the HP 260 using the following command:

RUN "BINCHK [,*volume name*] "

When the program starts, the screen displays all the devices linked to the HP 260, together with their volume names and device addresses. The message **`Please select a device`** is displayed. Press the softkey that indicates where the application software is stored. Check that the word **`SELECTED`** appears on screen next to the required volume name. If it does not, press softkey EXIT then softkey RESTART and select again.

At this point, the BINCHK program checks the selected device for programs containing binary code. Once this is done, the message **`Please enter a printer number`** appears. Type in a valid printer number. The programs found containing binary code are then printed.

The checking procedure is completed properly when the message **`BINCHK completed`** appears. You can interrupt the BINCHK program at any time by pressing END PROGRAM.

The next step is to use a series of commands that will automatically remove the binary code from a program. Here is an example using a program named DBMAP1:

```
LOAD”DBMAP1”
```

Erases any program and data in memory and loads DBMAP1 and any accompanying binary programs.

```
SAVE”dbmap1”
```

Creates a data file (dbmap1) and saves the program currently in memory (DBMAP1) into the file dbmap1 without the binaries. Note that this information is saved as string data.

```
SCRATCH A
```

Clears DBMAP1 and associated files from memory.

```
GET”dbmap1”
```

Reads into memory the dbmap1 file *minus* any binary code; thus, the correction needed is made.

```
RE-STORE”DBMAP1”
```

Stores the file currently in memory into the program file DBMAP1 (now without binaries).

```
PURGE”dbmap1”
```

Deletes the file dbmap1.

If a GET statement causes an error message in a program line of one of your applications, you have probably either not loaded the TOOLS DROM or you are using non-supported binaries. Example:

```
GET ”dbmap8”
```

Display:

```
1120  ACCEPT ID$

IMPROPER EXPRESSION
```

In this case the TOOLS DROM has not been loaded.

NOTE: The file names must not contain a slash (”/”). The following characters could also lead to conflict if used in file names: $ * : [ ] ( ).

Transferring HP 260 Files

This discussion assumes that the HP 260 application to be transferred is on a fixed disk or cartridge tape. The following steps explain the transfer process:

1 Login at the HP-UX prompt.

2 Use the HP-UX mkdir command to create a directory for the application.

For example, enter:

```
mkdir /extra/migrate/vol1
```

3 Change to this directory by entering the following:

```
cd /extra/migrate/vol1
```

4 If transferring your HP 260 application from a fixed disk, connect the disk to the HP 9000 Series 800.

If transferring your HP 260 application from a cartridge tape, insert the tape into the appropriate drive. If you are using a cartridge tape autochanger (HP 35401), insert the cartridge tape in the first drive.

5 Find out the device file name (character device file) of your fixed disk or cartridge tape drive. **`/dev/rdsk/c1001d0s2`** is an example of a device file name for a disk. **`/dev/rct/c1002d0s2`** is an example of a device file name for a cartridge tape drive.

6 Transfer the files on the fixed disk or cartridge tape into the current directory of the HP 9000 Series 800 using the following command:

```
ctd260 -xv device file name
```

It is possible to transfer only selected files. Refer to page 25 for more information.

7 Verify that the ctd260 command executed successfully by entering the following command:

```
ll
```

This command displays all files with the extension .DATA, .PROG, .FORM, .R260, and .DSET.

8 If you have an application that supports several volumes, you can create several directories (for example, **`/extra/migrate/vol2`** and so on) and you can then put the files into the appropriate directory using the HP-UX mv command. These volumes must be defined in your user, group, or global Eloquence configuration file.

Completing HP 260 Program Analysis

Before continuing with this section, make sure you have done the following:

1 Used the BINCHK program to check the HP 260 programs you intend to transfer to the HP 9000 Series 800.
2 Removed binary code from the HP 260 programs to be transferred.
3 Transferred your HP 260 applications from the HP 260 to the HP 9000 Series 800.

If you have completed the above steps, you are ready to complete the program analysis process.

list

The list command is used after the program (.PROG) files are transferred from the HP 260 to the HP 9000 Series 800.

The purpose of the list command is to make ASCII text files from the program files. This is a preliminary step to using the Eloquence stmtcheck command. Use the list command in the following manner:

list *programname* > *textfilename*

Refer to the Eloquence Manual for information on the options available with the list command.

By diverting the output to *textfilename* you now have available a source listing of the program as an ASCII text file.

To speed up this process, it is possible to write a script file that will convert all program (.PROG) files in the specified directory to ASCII text files named ***filename*.txt**. You might call this script file list_all. This is sure to be of assistance in the first phase of migration. Here is an example list_all script file:

```
for i in *.PROG
do
list $i > 'basename $i .PROG'.txt
done
```

stmtcheck

The next step is to check the text files for critical statements. Critical statements are those HP 260 statements that are not implemented or work differently in Eloquence. This check is done using the Eloquence stmtcheck command. Use the stmtcheck command in the following manner:

stmtcheck *textfilename*

This checks the named text file for critical statements. If any are found, the program and the program lines containing the critical statements are indicated. If you need a hard copy of this, use the following command sequence:

stmtcheck *textfilename* | lp

The added **`| lp`** sends the result of the stmtcheck to the default printer.

The stmtcheck program also works with wildcard characters. This simplifies the stmtcheck process. For example, if all the text files you created with list have the extension **`.txt`**, the command **`stmtcheck *.txt`** can be executed. This will cause all files in the current directory with the extension **`.txt`** to be checked, and therefore prevent executing the stmtcheck command for each text file.

NOTE: Before using the stmtcheck program, verify that there is a file named .260stmt in your HOME directory or in the directory **`/opt/eloquence6/tools`**. If not, change to the directory **`/opt/eloquence6/tools`** and copy the file d.260stmt to .260stmt. The file .260stmt contains the critical statements that the stmtcheck program checks for.

Refer to page 25 for more information on the Eloquence stmtcheck command.

store

Once you know which statements, if any, need to be changed, you can either change the program (.PROG) within Eloquence or change the text file using any HP-UX editor. If you chose to make the changes within Eloquence, delete the text file to save disk space. If you chose to change the text file, you must use the store command after making your changes. The Eloquence store command converts the ASCII text file containing program source code to an Eloquence program (.PROG) file. For instructions on how to use this command, refer to the Eloquence manual.

Creating an Eloquence Database

HP 260 databases cannot be used in their original form. They must be converted to Eloquence databases. This conversion process consists of four main steps, as shown below:

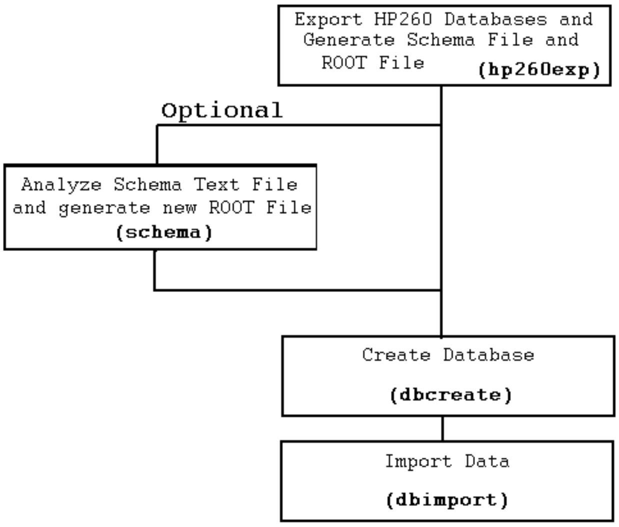
Export HP260 Databases and Generate Schema File and ROOT File (hp260exp)
Optional
Analyze Schema Text File and generate new ROOT File (schema)
Create Database (dbcreate)
Import Data (dbimport)

Figure 2 **Creation of Eloquence Database**

Carry out the following steps to convert an HP 260 database to an Eloquence database:

1 Use the hp260exp command to extract the contents of manual and detail data sets and convert them to ASCII text files. These text files are named in the format *databasename.datasetnumber*.exp. They are referred to as export files. The hp260exp command can also be used to create a schema text file named *databasename*.txt and a ROOT file named *databasename*.ROOT.

To explain this process, suppose you have an HP 260 database named SAD that has been transferred into the directory **`/extra/migrate/vol1`**. Also suppose you want to generate a schema text file and ROOT file in addition to the export files. The first step is to change to the appropriate directory and execute the hp260exp command as shown here:

```
cd /extra/migrate/vol1
```

```
hp260exp -v SAD
```

This hp260exp command generates the schema text file SAD.txt and the export files SAD.03.exp, SAD.04.exp, SAD.05.exp, SAD.06.exp. The reason the export files start with data set 03 is because the first two data sets are automatic data sets, and automatic data sets are filled automatically when the Eloquence dbimport command is executed.

The -v option specifies that a detailed listing of procedures be displayed. For a description of other options available, refer to page 25 .

2 Create a new database from the schema text file with the schema command. The schema command checks the syntax of the schema text file and, if the syntax check reveals no errors, creates a database catalog or optionally a ROOT file (for A.05.xx compatibility).

Using SAD.txt as an example, the command sequence is as follows:

```
schema SAD.txt
```

Refer to the Eloquence manual for more information on the Eloquence schema command.

NOTE: The export files are ASCII text files; therefore, numbers are stored as numbers and text in quotation marks. The division between individual data fields is marked by a comma.

3 Use the Eloquence dbcreate command to generate the Eloquence database. For example:

```
dbcreate -v SAD
```

Refer to the Eloquence manual for more information on the Eloquence dbcreate command.

4 The final step to convert an HP 260 database to an Eloquence database is to use the Eloquence dbimport command to load the data from the export (.exp) files into the data sets created using dbcreate. Here is an example:

```
dbimport -v SAD
```

This command sequence loads the data of the SAD export (.exp) files into the database.

Refer to the Eloquence manual for more information on the Eloquence dbimport command.

3

Eloquence Migration Tools

This chapter contains a description of the Eloquence programs used for the migration process.

ctd260

The ctd260 program reads HP 260 files of type PROG, DATA, FORM, ROOT, and DSET and copies them to the HP 9000.

Syntax

ctd260 [options] *device file*

Options:

-help	Show usage (this list)
-c	Catalog device.
-x	Extract files from device.
-o *path*	Output directory.
-n *arg*	Name mask (up to ten, default is all files).
-t *arg*	File types (up to ten, default is all types).
-v	Verbose.
-k	Keep files already existing (don’t overwrite).
-S	Select size of read buffer. See tcio(1) and ct(7)
devicefile	Enter device file from which to read. Specifying a hyphen (-) in place of a *devicefile* will force reading from the stdin (standard input device).

Examples:

Example One:

```
ctd260  -xv -ndbm* /dev/rdsk/c1001d0s2
```

Extracts files from disk.

```
ctd260  -xv -ndbm* /dev/rct/c1002d0s2
```

Extracts files from cartridge tape.

While one command sequence shows the device file name for a disk and the other a cartridge tape, both do the following:

- Extract all files whose names begin with **dbm** from the corresponding device file and display everything on the screen.

Display:

```
B1368B CTD260 (C) COPYRIGHT HEWLETT-PACKARD GmbH 1991 (A.03.00)

*** volume header ***

System id code: 600
Volume Size    : 16352 kbytes
Directory size:  4080 entries
User area      : 16160 kbytes
Volume label   :

NAME   PR TYPE RECS   RECL   START
----- -- ---- ------ ------ --------
dbmap2   DATA     11    256      768
dbmap1   DATA      7    256      779
dbmap3   DATA     16    256      786
dbmap4   DATA     11    256      802
dbmap5   DATA     13    256      813
```

Example Two:

```
ctd260 -xv -tFORM -tR260 /dev/rdsk/c1001d0s2
```

Files of type FORM and ROOT are extracted into the current directory.

As there are no file types under HP-UX, the file names have an extension which indicates the file type the file was on the HP 260:

***name*.PROG**	program file
***name*.FORM**	form file
***name*.DATA**	data file
***name*.R260**	ROOT file for named database
***name*.*nn*.DSET**	data set file for set *nn* of named database

Eloquence commands, statements, utilities work with *name*. Pure HP-UX commands or utilities work with complete file name.

NOTE: When transferring ROOT files the ctd260 program generate files with the extension .R260. These files denote an HP 260 root file.

Converting FVBACK files

FVBACK is a HP260 utility program to backup a whole disc into a set of backup files. The FVBACK file format is closely related to the HP260 media format.

The ctd260 utility can be used to extract files from archives created by the HP260 FVBACK utilitiy. The undocumented -f commandline flag is used to unpack HP260 FVBACK files.

To unpack an FVBACK archive, the following steps must be executed:

1. Transfer all FVBACK files from HP260 media onto local disk.

   FVBACK files are of type BKUP, which are normally ignored by ctd260. In order to extract them, you must specify the -t BKUP argument to ctd260.

2. Unpack the FVBACK files

   To unpack the FVBACK files, specify the -f commandline flag to ctd260 and the file name of the first FVBACK file. The file name of subsequent backup files are derived from the initial backup file name.

For example, if you have three FVBACK files named FVBFL.BKUP, FVBFL2.BKUP and FVBFL3.BKUP:

```
ctd260 -f -x FVBACK.BKUP
```

This will unpack all files, contained in the three backup files.

NOTE: This feature is considered unsupported because some revisions of FVBACK are known to create inconsistent FVBACK files so ctd260 may not be able to unpack them.

hp260exp

The Eloquence hp260exp command performs a serial or chained read on all data set files or the selected data sets of the given HP 260 database, extracts the data, and generates export text files labelled *databasename.datasetnumber*.exp. Automatic master data sets are excluded. In addition, it generates a schema text (.txt) file.

NOTE: Any existing maintenance password in the HP 260 database will *not* be taken over and must be re-installed in the new database with the dbmods program.

Syntax

hp260exp [options] *root file* [*set* ...]

Options:

-o *path*	Set output path.
-c	Chained export.
-s	No schema text file (only sets).
-v	Verbose.

NOTE: If *no* data set name or number is given all data sets belonging to the named database will be exported.

Examples:

To export all data sets from the SAD database, enter the following command:

```
hp260exp -v -o./export SAD
```

The output from this command is written to the subdirectory export. This is specified by the **`-o./export`** option. The period (.) before **`/export`** represents the current directory.

In addition to export files, this command generates a schema text file and root file for the SAD database.

Display:

```
B1368A HP260EXP (C) COPYRIGHT HEWLETT-PACKARD GmbH 1991 (A.03.00)

Processing data base SAD
Root file path:
-o path = set output path

DATA SET              CAPACITY #ENTRIES
---------------- -- - -------- --------
DATE             01 A       51        1
ORDER            02 A      101      100
PRODUCT          03 M       11        8
LOCATION         04 M       17       10
OPTION           05 D      300       16
CUSTOMER         06 D      100       11
  |              |  |        |        |
Data set name    |  |        |    Number of entries
                 |  |        |
                 |  |        |
                 |  |     Possible number of entries
                 |  |
                 |  |
                 | Data set type
                 | A = automatic
                 | M = manual master
                 | D = detail
                 |
               Data set number
```

To verify that the export files, schema text file, and root file have been created, use the HP-UX ll (long list) command. For example, the following command displays the export files:

```
  ls -l *exp

-rw-rw-rw-   1 guest   migctr     105 Aug 14 21:12 SAD.03.exp
-rw-rw-rw-   1 guest   migctr       9 Aug 14 21:12 SAD.04.exp
-rw-rw-rw-   1 guest   migctr     324 Aug 14 21:12 SAD.05.exp
-rw-rw-rw-   1 guest   migctr     698 Aug 14 21:12 SAD.06.exp
```

stmtcheck

The stmtcheck program reads HP 260 program files that have been converted to ASCII files and checks for statements that are not implemented or work differently in Eloquence. The file names and the critical program lines are listed on the standard output device (stdout).

Before using the stmtcheck program, verify that there is a file named .260stmt in your HOME directory or in the directory **`/opt/eloquence6/tools`**. If not, change to the directory **`/opt/eloquence6/tools`** and copy the file d.260stmt to .260stmt. The file .260stmt contains the critical statements that the stmtcheck program checks for.

Syntax:

stmtcheck *file* [*file* ...]

Example:

This example shows how to check text files dbmap1.txt and dbmap2.txt for statements contained in the file .260stmt. For a listing of these two files, refer to the Eloquence DBMS manual.

Execute the stmtcheck command:

```
stmtcheck dbmap1.txt dbmap2.txt
```

The output generated by the stmtcheck program shows the program name and the line number containing any statement defined in the .260stmt file.