Compaq BASIC for OpenVMS VAX Systems V3.9 Release Notes

Summary of VAX BASIC V3.9

November 2000

© 2000 Compaq Computer Corporation

COMPAQ, VAX, VMS, and the Compaq logo Registered in U.S. Patent and Trademark Office.

OpenVMS is a trademark of Compaq Information Technologies Group, L.P. in the United States and other countries.

Motif is a trademark of The Open Group in the United States and other countries.

All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from Compaq required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

Compaq Computer Corporation
Houston, Texas

Contents

1	**RELEASE SUMMARY**	**1**
2	**GENERAL NOTES**	**1**
3	**NEW FEATURES IN THIS RELEASE**	**1**
4	**PROBLEMS FIXED IN THIS RELEASE**	**1**
5	**KNOWN PROBLEMS AND RESTRICTIONS IN VAX BASIC V3.9**	**1**
	5.1 Corrections to Documentation	**3**
	5.1.1 Reference Manual • 3	
	5.1.2 User's Guide • 3	
6	**ADDITIONAL INSTALLATION CONSIDERATIONS**	**3**
7	**REPORTING SOFTWARE PROBLEMS**	**4**

1 Release Summary

Compaq BASIC for OpenVMS VAX Systems is a native compiler that runs on and generates code for OpenVMS for VAX Systems. Version V3.9 requires OpenVMS for VAX Systems, Version 5.5-2 or later to run. The primary purpose of this release is rebranding, otherwise it contains only minor bug fixes.

2 General Notes

- Throughout these notes, every use of the name Compaq BASIC or the name BASIC used alone, applies to both Compaq BASIC for OpenVMS VAX Systems, and Compaq BASIC for OpenVMS Alpha Systems. VAX BASIC refers to information that applies specifically to Compaq BASIC for OpenVMS VAX Systems. Alpha BASIC refers to information that applies specifically to Compaq BASIC for OpenVMS Alpha Systems.
- The DATE4$ function is available to ease support of dates in the 21st century and beyond. It functions the same as the DATE$ function except that the year portion of the result string contains two more digits indicating the century. For example:

```
PRINT 32150, DATE$ (32150), DATE4$ (32150)
```

Produces the following output:

```
32150   30-May-02   30-May-2002
```

3 New Features In This Release

No new functionality has been included in VAX BASIC V3.9.

4 Problems Fixed In This Release

The following problems in previous versions of VAX BASIC have been corrected in VAX BASIC V3.9:

- BASIC$STARLET.TLB is missing or contains different definitions than expected. BASIC$STARLET.TLB is now generated from directly from SYS$LIBRARY:STARLETSD.TLB during installation of VAX BASIC so that the definitions will match.

5 Known Problems and Restrictions in VAX BASIC V3.9

The following summarizes problems and restrictions in VAX BASIC V3.9.

- The most negative integer constants cannot be represented. The compiler incorrectly issues an error on references to constants representing the most negative integer of each of the integer data types.

The following table gives appropriate workarounds.

problem integer	workaround
BYTE –128%	BYTE –127% – 1%
WORD –32768%	WORD –32767% – 1%
LONG –2147483648%	LONG –2147483647% – 1%

- Definitions for RAB$V..., FAB$V..., etc. are not included in BASIC$STARLET.TLB since there is no support for bit fields in the BASIC language.
- If a VAX BASIC program contains more than 32767 lines of code, the compiler will feed the debugger incorrect source line information for the lines above 32767.
- Programs using records 32768 to 65535 bytes in size as array elements generate the RECTOOLAR error message when compiled /DEBUG or compiled using the RUN command within the BASIC environment, and no object file is generated. These programs do not generate the error otherwise and can be compiled and linked to produce runnable programs.
- You cannot concatenate programs at DCL level with a plus sign (+) if one or more of the programs contain no BASIC line numbers. For example, the following command will generate an error if either A.BAS or B.BAS contains no line numbers:

```
$ BASIC A.BAS+B.BAS
```

- VAX BASIC allows a maximum of 65535 dynamic string variables, dynamic string array elements, and string temporaries in a single compilation unit. An attempt to allocate more than 65535 dynamic strings may result in a Stack Overflow error at run time.
- In a VAX BASIC subprogram, a parameter passed to the subprogram that is an RFA array cannot be the target parameter of a call to an external RFA function. The VAX BASIC compiler reports an internal coding error during compilation of such a subprogram. Workarounds for this problem include passing the RFA array to the subprogram by reference or assigning the result via an intermediate variable.
- Sharing a MAP between a BASIC module and a shareable image does not work if the non-shared image is run in the BASIC environment.
- When using the CTRLC function through a SET HOST session, the CTRLC function is not always executed. If the same program is run from a local session, it behaves correctly.
- Failures on accesses to virtual arrays when made from within a WHEN block protected region will cause the application to transfer control to the default error handling mechanism (ON ERROR GOTO) if it exists. All upper level protected region handlers are ignored.

  Please note that this problem has existed ever since WHEN blocks were first introduced. It may, however, be fixed in a future release of the OpenVMS for VAX Systems.
- A varying string data type that is part of a record structure pulled from the CDD into a BASIC program may be assigned a length different than what is specified in CDD. This will be fixed in a future release of the CDD/Repository.
- Under some circumstances, VAX BASIC may produce an SCA analysis file with the wrong format for a program using EXTERNAL functions.
- Using the RFA data type as the data type of a FUNCTION may cause a reserved address fault at run-time.

- Cross-reference listings incorrectly list MOVE TO statements as destructive references of the variables whose values are being moved into the I/O buffer. MOVE TO statements are not destructive references.
- If your program contains a comment line beginning with a BASIC line number, the VAX BASIC compiler generates incorrect OpenVMS DEBUGGER line number information. If the first line after the comment also has a BASIC line number and that line is the target of a GOTO or GOSUB, then any breakpoint that is set on that GOTO or GOSUB target is not triggered.
- If an array element specification subscripted with a conditional expression is used as the conditional expression in a WHILE or UNTIL statement, an internal coding error results.
- The /NOWARN qualifier does not suppress the warning 'BASIC-I-DUPLNFND, Duplicate line number found'.

5.1 Corrections to Documentation

The following corrections should be made to the documentation.

5.1.1 Reference Manual

Corrections to the *Compaq BASIC for OpenVMS Alpha and VAX Systems Reference Manual*.

- In Chapter 4, under the COMMON statement (page 4-19), item 6 of the Syntax Rules should read: "*Data-type* can be any BASIC data type keyword or a data type defined by a RECORD statement. Data type keywords, size, range, and precision and listed in Table 1-2."

5.1.2 User's Guide

Corrections to the *Compaq BASIC for for OpenVMS Alpha and VAX Systems User's Manual*.

- The following information should added to Section 4.6 Stepping Into BASIC Routines.

  "Stepping INTO or OVER a GOSUB statement behaves in what may appear to be an unpredictable manner. This can be explained by knowing that a GOSUB is really just a special kind of GOTO statement which is handled by a RTL routine. STEP/INTO a GOSUB steps into the RTL routine as shown in Table 4-1 Resultant Behavior of the STEP /INTO Command. STEP or STEP/OVER a GOSUB steps over the RTL call to the following point in the control stream which is the actual code of the GOSUB routine."

6 Additional Installation Considerations

To install VAX BASIC requires approximately 958 global pages and 4 global sections.

A WORKING SET quota of 400 is recommended to optimize performance for accounts primarily accessing VAX BASIC.

In order to use the BASRTL included on this kit, extract the image from saveset BASIC039.E using, for example:

```
$ BACKUP/LOG BASIC039.E/SAVE/SELECT=BASRTL.EXE *
```

Then copy the BASRTL image to the system library and install it in IMAGELIB. The following commands show how to do this.

```
$ COPY BASRTL.EXE SYS$COMMON:[SYSLIB]
$ LIBRARY SYS$SHARE:IMAGELIB SYS$SHARE:BASRTL.EXE
```

To update the installed image on a standalone system use:

```
$ INSTALL REPLACE SYS$SHARE:BASRTL
```

To update the installed image in a cluster environment use:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> DO INSTALL REPLACE SYS$SHARE:BASRTL
SYSMAN> EXIT
```

7 Reporting Software Problems

Complete and concise information will help to receive accurate and timely service to software problems. Experience shows that many problem reports do not contain sufficient information to duplicate or identify the problem.

Review the Software Product Description (SPD) and Warranty Addendum for an explanation of warranty. If you encounter a problem during the warranty period, report the problem as indicated or follow alternate instructions provided for reporting SPD nonconformance problems.

When you prepare to report a problem, please take the following steps:

- Describe as accurately as possible the circumstances and state of the system when the problem occurred. Include in the description the version and product at fault, and the version and operating system being used. (You can find the version number by typing BASIC and pressing RETURN.) Illustrate the problem with specific examples.
- Reduce the problem to as small a size as possible, preferably to a program unit of 30 source lines or fewer.
- Remember to include listings of any command files, relevant data files, CDD record definitions, and so on.
- Provide a listing of the complete source program and any INCLUDE files.
- Submit sources electronicly or on machine-readable media if possible. No media will be returned to you when the report is answered unless you specifically request so.

- Report only one problem per report. This will facilitate a faster response.