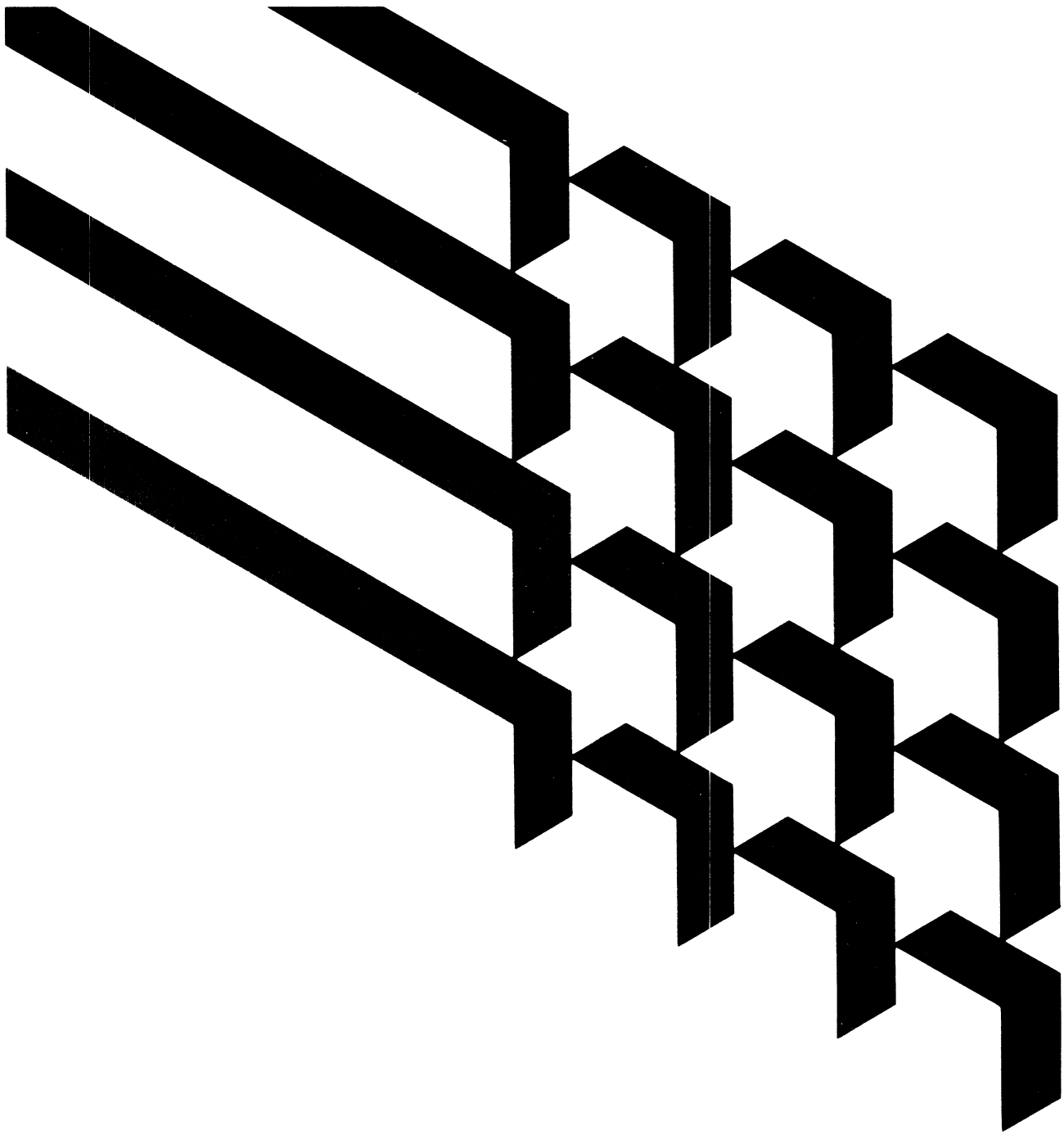
HEWLETT PACKARD

BASIC/1000C

Installation and Configuration Guide



BASIC/1000C

Installation and Configuration Guide

HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014

MANUAL PART NO. 92857-90002
Printed in U.S.A. March 1983
E0383

PRINTING HISTORY

The Printing History below identifies the Edition of this Manual and any Updates that are included. Periodically, Update packages are distributed which contain replacement pages to be merged into the manual, including an updated copy of this Printing History page. Also, the update may contain write-in instructions.

Each reprinting of this manual will incorporate all past Updates, however, no new information will be added. Thus, the reprinted copy will be identical in content to prior printings of the same edition with its user-inserted update information. New editions of this manual will contain new information, as well as all Updates.

To determine what software manual edition and update is compatible with your current software revision code, refer to the appropriate Software Numbering Catalog, Software Product Catalog, or Diagnostic Configurator Manual.

Preliminary Edition ..... Jan 1983
First Edition ..... Mar 1983

CONTENTS

ILLUSTRATIONS

TABLES

Chapter 1
Introduction

This manual explains the installation and configuration of the HP 92857A BASIC/1000C language software, which includes an interpreter program and a compiler program. In this manual, the term "BASIC/1000C" refers to both the interpreter and the compiler.

Chapter 2, Components of BASIC/1000C, lists the manuals that are shipped with the BASIC/1000C product, and describes the programs that comprise it and the files included with it.

Chapter 3, System Requirements, describes the operating system, disc, memory, and system library requirements for BASIC/1000C, and special requirements for compiled programs. Read Chapter 3 and be sure that system requirements are met before installing the product.

Chapter 4, Installation of the BASIC/1000C Product, describes the installation procedure for BASIC/1000C and explains how to adjust program parameters to improve performance (configuration).

Chapter 2
Components of BASIC/1000C

The BASIC/1000C product consists of manuals, programs, interpreter files, and compiler files.

Manuals

The following manuals are shipped with BASIC/1000C product:

BASIC/1000C Reference Manual (Part No. 92857-90001)
BASIC/1000C Configuration Guide (Part No. 92857-90002)
BASIC/1000C Quick Reference Guide (Part No. 92857-90003)
Link Support Document (Part No. 92857-90004)

Programs

BASIC/1000C is comprised of the programs BASIC, RBEX, CBASC, BDAT, RINTR, and BBMG.

BASIC

BASIC is the program that performs the preparation phase of a BASIC interpreter session. This program is the part of the interpreter that checks syntax and builds intermediate code for the program RBEX to execute. It contains a simple line editor and does most of the command processing (e.g., SAVE).

RBEX

RBEX is the program that performs the execution phase of a BASIC interpreter session. It is scheduled when the user enters RUN from the BASIC interpreter. RBEX contains the debug environment, and is a Virtual Memory Area (VMA) program.

CBASC

CBASC is the BASIC compiler. A compiled BASIC program executes faster than a program run in the interpreter.

BDAT

BDAT is a utility program that creates block data entry points for common blocks. BDAT is run before segmenting a compiled program with SGMTR, the automatic segmenter, and/or before linking with MLLDR, the multilevel segmentation loader (used with the RTE-6/VM operating system only).

RINTR

RINTR logs interrupts and alerts BASIC that interrupts have occurred. RINTR must be RP'd if the ON INTR statement (used for interrupt processing) is used in BASIC/1000C. Only one copy of RINTR is needed, but multiple BASIC programs can handle interrupts.

BBMG

BBMG builds information necessary for interpreted BASIC programs to communicate with compiled programs. BBMG builds two files: An information file for BASIC and a dummy main for the overlay where the compiled routines reside.

Interpreter Files

The following files are supplied with the BASIC interpreter:

*BAS.6	—	FMP Transfer file and info to link BASIC for RTE-6/VM.
*BAS.A	—	FMP Transfer file and info to link BASIC for RTE-A.
#BAS.6	—	LINK command file for BASIC on RTE-6/VM.
#BAS.A	—	LINK command file for BASIC on RTE-A.
#RBX.6	—	LINK command file for RBEX on RTE-6/VM.
#RBX.A	—	LINK command file for RBEX on RTE-A.
#BBMG	—	LINK command file for BBMG.
#RINTR	—	LINK command file for RINTR.
%FOX.6	—	Force into executor relocatable for RTE-6/VM.
%FOX.A	—	Force into executor relocatable for RTE-A.
%SAM.6	—	Samurai relocatable for RTE-6/VM.
%SAM.A	—	Samurai relocatable for RTE-A.
%B$T12	—	Relocatable used by RINTR.
%RINTR	—	Relocatable for RINTR.
%READA	--	Relocatable to replace the A.1 version of FMP routine READF.
$BCALL	—	Relocatable library for linking to binary modules.
%BEXEC	—	Relocatable BASIC-EXEC Interface.
"BERRS	—	Type 1 error file for BASIC/1000C.
"B.ERR	—	Type 4 error file for users.
#MRBAS	—	MERGE command file for BASIC library
#MRRBX	—	MERGE command file for RBEX library.
%BSSKL	—	BASIC MERGEd relocatable skeleton.
$BLIB1	—	BASIC MERGEd relocatable library part 1.
$BLIB2	—	BASIC MERGEd relocatable library part 2.
%RXSKL	—	RBEX MERGEd relocatable skeleton.
$RLIB1	—	RBEX MERGEd relocatable library part 1.
$RLIB2	—	RBEX MERGEd relocatable library part 2.
$RLIB3	—	RBEX MERGEd relocatable library part 3.
%BMSKL	—	BBMG MERGEd relocatable skeleton.
$LBMGL	—	MERGEd and LINDXed BBMG library.

Compiler Files

Files for loading the compiler:

*CBA.6	—	FMP transfer file and info to link CBASC for RTE-6/VM.
*CBA.A	—	FMP transfer file and info to link CBASC for RTE-A.
#MERGC	—	Command file for MERGE to merge the relocatables of CBASC.
#CBA.6	—	LINK command file for CBASC on RTE-6/VM.
#CBA.A	—	LINK command file for CBASC on RTE-A.
%CBA.1	—	Relocatable for CBASC, part 1.
%CBA.2	—	Relocatable for CBASC, part 2.
%SAMAC	—	SAMURAI segmentation for CBASC for RTE-A.
%SAM6C	—	SAMURAI segmentation for CBASC for RTE-6/VM.
$ALIB	—	LINDXed Library for CBASC.
#BDAT	—	LINK command file for BDAT utility.
%BDAT	—	Relocatable for COMMON utility program for MLLDR loader.

Files for loading a compiled program:

#LNK.L	—	LINK command file for loading with local memory management.
$BASCL	—	LINDXed run-time Library.
%IB.XX	—	Relocatable for dummy HP-IB library.
&IB.XX	—	Source for dummy HP-IB library.
#LNK.E	—	LINK command file for loading with EMA memory management.
#LNK.V	—	LINK command file for loading with VMA memory management.
%MMGT2	—	EMA/VMA memory management routines.
%L.EMA	—	EMA/VMA memory management used with LINK.
%RT.6M	—	EMA/VMA mapping routines for RTE-6/VM.
%RT.AM	—	EMA/VMA mapping routines for RTE-A.
%B.EMA	—	EMA buffer routines.
%B.EIO	—	EMA I/O routines.
%B.VMA	—	VMA buffer and I/O routines.
%F.EMA	—	EMA/VMA memory management used with MLLDR.
%S.EMA	—	Relocatable for memory management workspace for MLLDR.
&S.EMA	—	Source for memory management workspace for MLLDR.
*B.MLE	—	Edit file for MLS programs using EMA memory management.
*B.MLV	—	Edit file for MLS programs using VMA memory management.

Files for the LINK loader revision 2302:

%LINKA	—	LINK segment A.
%LINKB	—	LINK segment B.
%LINKC	—	LINK segment C.
%LINKD	—	LINK segment D.
%LNKRA	—	LINK relocatable for RTE-A.
%LNKR6	—	LINK relocatable for RTE-6/VM.
%LNKDA	—	LINK default for RTE-A.
%LNKD6	—	LINK default for RTE-6/VM.

Chapter 3
System Requirements

This section explains operating system, disc, memory, and system library requirements for BASIC-/1000C, and special requirements for compiled programs.

Operating System Requirements

BASIC/1000C is supported on the following system configurations:

Operating System	Processor
RTE-6/VM REV 2226	E, F
RTE-A REV 2301	A600, A700, A900

VMA microcode must be installed on E- and F-series computers. Refer to the *RTE-6/VM Software Installation Manual* (Part No. 92084-90011), Chapter 5.

BASIC/1000C Disc Requirements

Table 3-1 describes the files that remain on the system after installation. Temporary files that are created during interpreter operation are described in the second section below. The disc space requirements for files used only during the installation of the BASIC/1000C product are listed in the section Disc Space Requirements for Installation.

Table 3-1. Files after Installation of BASIC/1000C

FILE	APPROXIMATE NUMBER OF 128-WORD DISC BLOCKS REQUIRED
BASIC	1210
RBEX	1800
BBMG	370
RINTR	50
"BERRS	350
"B.ERR	190
$BCALL	11
%READA	17
%BEXEC	5
CBASC	1020
BDAT	200
$BASCL	780
#LNK.L	2
%IB.XX	2
&IB.XX	8
#LNK.E	4
#LNK.V	4
%MMGT2	6
%L.EMA	2
%RT.6M	2
%RT.AM	2
%B.EMA	6
%B.EIO	3
%B.VMA	9
%F.EMA	1
%S.EMA	1
&S.EMA	5
*B.MLE	5
*B.BLV	5

The files above fall into the following categories:

1. Type-6 files for program storage.

   Each user uses a separate copy of BASIC, RBEX, CBASC, BDAT, and BBMG. Only one copy of RINTR is needed per system.

2. Files that exist only while the interpreter is executing.

File	Minimum Number of Disc Blocks Required	Type
0XXY00	24	1
0XXY01	96	3
0XXY02	96	1
0XXY03	96	1
0XXY04	96	2
0XXY05	96	2

NOTE

The digit sequence XXY is assigned by FMGR. Each file above, except 0XXY00, can be larger if extents are required.

It is recommended that a system cartridge be dedicated to scratch files. Many programs other than BASIC use scratch files (e.g., LINK). Refer to the system installation instructions to set up this cartridge.

3. Error scratch file for CBASC.

CBASC creates a type-2 extendable 24-block file, used for error handling during compilation. Its size is determined by the number of errors encountered during compilation.

4. Virtual Memory Area files used by RBEX and CBASC.

RBEX and CBASC are VMA programs. VMA programs maintain an image of the VMA area in a disc file named 0XXXVM, where XXX is assigned by the system. The size of this file is set by the VS command. In the LINK command files for RBEX and CBASC, the size is set to 500 pages, which is usually adequate. The section BASIC and VMA Usage explains how the size of this file affects performance.

5. Error files "BERRS and "B.ERR.

Two error files are supplied with BASIC/1000C, "BERRS and "B.ERR. Both contain the same BASIC error messages, but in different forms. "BERRS is a type-1 file, used by the interpreter, compiler, and compiled programs to report errors. "B.ERR is a type-4 file, used by user programs that print their own error messages. In "B.ERR, the error number corresponds to the record number in the file. The first record of "B.ERR also contains the part number of the file, which can be removed with EDIT/1000.

6. Library files and other files used to load compiled programs.

Most of these files are relocatable libraries and LINK and EDIT/1000 command files. For descriptions and uses of these files, refer to the section Compiler Files in this manual, and to Chapter 3 of the BASIC/1000C Reference Manual (Part #92857-90001).

7. User-created files.

   These files are of two types:

   a. Files created in the interpreter with the STORE or SAVE command, which save BASIC programs.
   b. Files created by BASIC programs for data storage.

Memory Requirements

The approximate memory requirements for the BASIC/1000C subsystem are:

Program	Approximate Partition Size (in Words)
BASIC	32K
RBEX	176K on E- and F-Series, 198K on A-Series
CBASC	98K on E- and F-Series, 120K on A-Series
BBMG	32K
RINTR	7K
BDAT	27K

System Library Requirements

The system libraries required to run BASIC on RTE-6/VM are:

$PLIB
$6SYLB
$MLIB1
$MLIB2
$IB6A

On RTE-A, they are:

$PLIB
$BIGLB

Special Requirements for Compiled Programs

Programs compiled with CBASC must be linked with LINK, revision 2302 or later. This version of LINK is provided with the BASIC/1000C product.

The MERGE utility program must be used when loading a compiled program with MLLDR, the multilevel segmentation loader. MERGE must be revision 2226 or later.

Chapter 4
Installation of the BASIC/1000C Product

BASIC/1000C can be installed on a system that:

- Has the utility programs required for installation
- Meets the disc space requirements

Installing the BASIC/1000C product includes:

- Copying files from shipped media to cartridge 32727
- Loading the version of LINK provided with the product
- Installing the interpreter (BASIC, RBEX, RINTR, BBMG)
- Installing the compiler (CBASC, BDAT (RTE-6/VM only))
- Adjusting program parameters

Utility Programs Required for Installation

The transfer files used to install the BASIC/1000C product require the following utility programs:

Program	Minimum Revision Date
LINK	2302
LINDX	2226
MERGE	2226

NOTE

LINK revision 2302 is provided with the BASIC/1000C product. Loading instructions are in the section Loading LINK.

Disc Space Requirements for Installation

BASIC/1000C requires the following disc resources for installation:

- a 200-track cartridge with cartridge reference number 32727, to be used for media storage.
- 170 contiguous tracks, to be used for scratch space during the installation process. If the system scratch cartridge is used, the 170 tracks must be contiguous during the installation process.
- Space for the resulting type-6 program files (73 tracks with 128 sectors per track, and 97 tracks with 96 sectors per track).
- Space for general user files for BASIC program development (23 tracks with 128 sectors per track, and 30 tracks with 96 sectors per track).

Copying Files from Shipped Media to Cartridge 32727

Using FC (File Copy Utility), copy the BASIC/1000C files from the shipped media to disc cartridge 32727. Thirty-four tracks with 128 sectors per track or 45 tracks with 96 sectors per track must remain free for use during installation.

Loading LINK

Load the LINK loader provided with BASIC/1000C according to the procedure in the LINK Relocating Loader Manual (Part No. 92077-90009).

Interpreter Installation (BASIC, RBEX, RINTR, BBMG)

Install the interpreter by transferring to the file *BAS.6 (on RTE-6/VM) or *BAS.A (on RTE-A), issuing the parameters described in Figure 4-1. Note that file names ending with ".6" are to be replaced with file names ending with ".A" for installation on an RTE-A system.

```
:*To LINK the interpreter, enter:
:*
:*        TR, BAS.6,<LNKCRN>,<SPCRN>,<USERCRN>,<SCRCRN>
:*  For safety purposes, all the command files below specify
:*  cartridge '32727' for the location of all relocatable
:*  files.  If you already have a cartridge 32727, or wish to
:*  name <LNKCRN> (see below) something else, all command files
:*  should be edited to reflect the change.  These files are:
:*
:*        #BAS.6      #RINTR
:*        #RBX.6      #MRBAS
:*        #BBMG       #MRRBX
:*
:*  NOTE: All Track sizes given below have 96 Sectors/Track.
:*
:*  <LNKCRN> is the name of the cartridge where all the files to
:*  LINK the interpreter are located.  Two large files ($BASL and
:*  and $RBXL) will be created on this cartridge during the MERGEing
:*  process. The LINK output files will also be placed on this
:*  cartridge.  If something goes wrong during the MERGEing, these
:*  two files should be purged (this transfer file will try to purge
:*  them).  To have enough room, this cartridge should have
:*  45 free Tracks.
:*
:*  <SPCRN> is the name of the cartridge where all the programs
:*  (type-6 files) are to be stored. About 75 Tracks of space are
:*  needed for the interpreter's type-6 files.
:*
:*  <USERCRN> is the name of the cartridge where files
:*  that must be available to all users will be stored.
:*  This includes the files: $BCALL, %BEXEC, "BERRS, "B.ERR,
:*  and %READA (used by compiled programs).  13 Tracks of space
:*  are needed for these files.  If any of these files are not
:*  transferred correctly, a message will be issued; however, the
:*  transfer file will continue.  When it finishes, please fix
:*  the problem and move the files manually.
:*
:*  <SCRCRN> is the name of the cartridge where the scratch
:*  files used by LINK will temporarily reside.  Two large
:*  LINDX files ($LBASL and $LRBXL) are created on this cartridge
:*  and then purged after the LINKing process.  NOTE: Little
:*  should be happening on <SCRCRN> when the interpreter is being
:*  LINKed or you may run out of room.  If for some reason these
:*  files are left lying around, purge them.  <SCRCRN> should
:*  have 170 free Tracks.
```

Figure 4-1. Listing of Beginning of *BAS.6 File

The *BAS.6 and *BAS.A files perform the following tasks:

1. Purge old LINDXed libraries ($LBASL and $LRBXL).
2. OF BASIC, RBEX, RINTR, and BBMG, in case they are RP'd.
3. MERGE and LINDX RBEX (interpreter executor) library.
4. LINK RBEX.
5. MERGE and LINDX BASIC (interpreter editor) library.
6. LINK BASIC.

7. LINK BBMG and RINTR.

8. Copy the files "BERRS, "B.ERR, %BEXEC, and $BCALL to the user cartridge.

9. Purge LINDXed libraries.

Compiler Installation (CBASC, BDAT)

Install the compiler by transferring to the file *CBA.6 (on RTE-6/VM) or *CBA.A (on RTE-A), issuing the parameters described in Figure 4-2. Note that file names ending with ".6" are to be replaced with file names ending with ".A" for installation on an RTE-A system.

```
:*  To link CBASC, enter:
:*
:*       TR, *CBA.6,<LNKCRN>,<SPCRN>,<USERCRN>,<SCRCRN>
:*
:*       For safety purposes, the LINK command file for CBASC and
:*       the MERGE command file below reference cartridge 32727.
:*       These files should be edited to the <LNKCRN> if it is
:*       other then 32727.  The files to edit are:
:*
:*            #CBA.6
:*            *MERGC
:*
:*       <LNKCRN> is the name of the cartridge where all
:*       the files to link BASIC are located.
:*
:*       <SPCRN> is the name of the cartridge where CBASC and BDAT
:*       (type-6 files) is to be stored.
:*       About 26 Tracks (96 Sectors/Track) of space if needed
:*       for the CBASC and BDAT type-6 files.
:*
:*       <USERCRN> is the name of the cartridge where files
:*       that must be available to all users are to be stored.
:*
:*       <SCRCRN> is the name of the cartridge where the
:*       scratch files used by link will temporarily reside.
:*
```

Figure 4-2. Listing of Beginning of *CBA.6 File

The *CBA.6 and *CBA.A files perform the following tasks:

1. Use MERGE to merge the two relocatable pieces of the compiler (%CBA.1 and %CBA.2) into %CBASC.

2. OF BDAT (on RTE-6/VM) and CBASC, in case they were RP'd.

3. Use LINK to link BDAT (on RTE-6/VM) and CBASC. (LOADR and MLLDR cannot be used instead of LINK.)

4. Copy the files necessary for loading a compiled program to a user cartridge (See Section 1.2.4 for the list of these files).

5. Purge the %CBASC file if step 3 was successful.

Cartridge 32727 is no longer needed.

Adjusting Program Parameters

Before running the BASIC/1000C product, adjust the program parameters as described below. This ensures that the appropriate system resources are allocated to BASIC/1000C. These steps affect the execution speed of the product.

BASIC and VMA Usage

BASIC/1000C uses system resources during execution. To obtain the best results from BASIC, it is important to understand how these resources are used, and how their use can be adjusted.

Both RBEX and CBASC use the virtual memory capabilities of RTE-6/VM or RTE-A extensively. Unlike normal VMA programs, however, they use the virtual area for both code segments and data. They also use the working set (the memory-resident part of the virtual area) for both code and data.

Figure 4-3 shows the virtual area in increasing level of detail (that is, the Virtual Memory Area consists of the Working Set and the Backing Store; the Working Set consists of Code and Data parts).

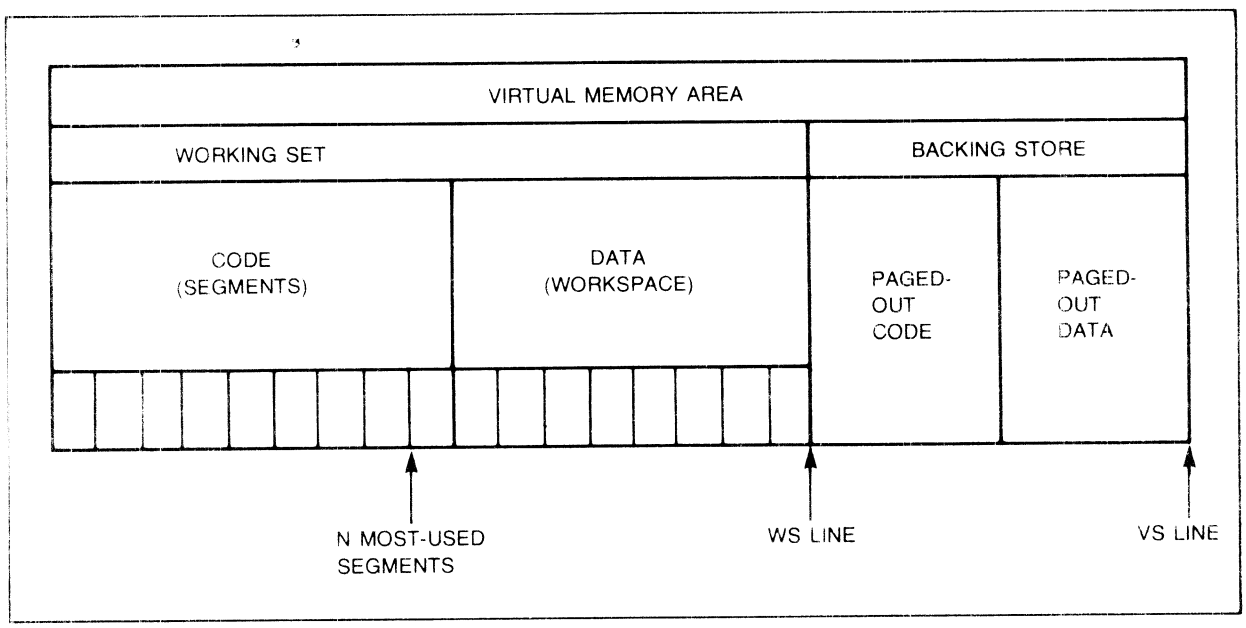


Figure 4-3. Virtual Memory Area

The WS and VS lines can be moved left or right with commands and options, to tune the performance of BASIC. The division of the working set into code segments and data workspace is beyond user control; BASIC makes the optimum separation.

The VS disc file resides either on the top user cartridge, or on the system scratch-file cartridge. A system scratch-file cartridge is recommended; it is designated with the VL command on RTE-6/VM or with an SC command in the boot file on RTE-A. The scratch cartridge is used by all copies of BASIC/1000C, EDIT/1000, and other programs that use scratch files.

A VM33 ERROR is reported if there is not enough space on the cartridge to allocate the VS disc file. Either reclaim space on that cartridge, or designate a larger scratch cartridge.

Interpreter Virtual Size and Working Set Size

Virtual size is adjusted with the VS operator command to the operating system. For example, the command `SYVS,RBEX,510` to RTE-6/VM (or `VS,RBEX,510` to RTE-A) sets the virtual size to 510 pages. Virtual Size does not affect execution speed.

The size of the interpreter must be large enough to execute the largest user program, but not so large that it wastes disc space. Set it to a value of 280 to 65K pages for RTE-6/VM systems; 280 and 6144 pages for RTE-A systems. For most purposes, the default value of 500 pages is sufficient. If it is not, an error 109 (out-of-room) occurs.

Working set size directly affects interpreter speed. It is adjusted with the WS system command to the operating system; for example, `SYWS,RBEX,150` to RTE-6/VM (or `WS,RBEX,150` to RTE-A). The working set size must be less than or equal to the size of the largest EMA partition on the system, minus the interpreter program size (29 pages).

On RTE-6/VM, the LINK command file sets WS to 147, which is adequate for most applications. A WS value of 147 requires a 176K partition (147 + 29).

On RTE-A, the LINK command file sets WS to 169, which is adequate for most applications. A WS value of 169 requires a 198-page partition (169 + 29).

Decreasing the working set size can increase the number of copies of BASIC that can run concurrently, although each user program will execute more slowly. The following section explains how virtual size and working set size affect performance.

Program priority, system load, and disc speed also affect performance.

After BASIC has been installed with the default virtual size and working set size, the virtual size and working set size can be adjusted to increase execution speed, decrease memory use, or decrease the amount of disc space required for the VMA file.

The virtual size and working set size are interdependent. Increasing the working set size and decreasing the virtual size can cause VM82 errors. Follow the guidelines below to avoid these problems.

The working set size (WS) determines the amount of memory and time that the interpreter requires to execute a BASIC program. Execution time is affected by the amount of memory available. If the working set is too large, more memory is required to execute a program, but performance does not improve. If the working set is too small, more time is spent moving interpreter segments or program pages between memory and the VMA file, and the program executes more slowly. Generally, moving interpreter segments degrades performance more than moving program pages.

The default working set size is just above what is required for all seven interpreter execution segments and enough user memory to run reasonably large programs without accessing the disc.

Decreasing the working set size reduces the amount of memory required to run a BASIC program, but can degrade performance.

Increasing the working set size can improve the performance of very large programs; however, a large increase in working set size can require an increase virtual size (as explained later in this section).

Tables 4-1 through 4-5 summarize the results of a benchmark test that demonstrated the effect of segment swapping with several computer/disc combinations.

These tables show:

1. The dramatic advantage of a large working set. Although programs that do not use all of the features of BASIC/1000C (or use them infrequently) can achieve maximum performance with working sets smaller than those shown in the tables, using even one disc-resident segment degrades performance significantly.

2. The relationship between working set size and performance on quiescent systems. The benchmark test was run on systems where no other programs (including other copies of BASIC) were performing significant activities.

3. The relative positions of the performance breakpoints for the BASIC interpreter on the E-series, F-series, A600, A700, and A900 computers. Program size can shift the breakpoint; for example, a small program may run faster on an E-series computer when the working set size changes from 95 to 96, but a large program might not show improvement until the working set size is 99 (see Table 4-1).

These tables do not show:

1. The performance of any given program at any given working set size. This benchmark, and the small amount of data that it requires, fits in a small amount of memory. For a large program, or a program that uses large data structures, the movement of user pages to and from the VMA file can impact performance, increasing the number of pages required to benefit from the performance breaks shown in the tables.

2. The relative performance of the computers involved. Since the computers did not use the same discs, disc speed varied. Even when all interpreter segments reside in memory, the process of transferring to the BASIC Executor and loading the execution segments into memory requires much disc access, so disc speed affects performance.

Tables 4-1 through 4-5 show execution times for small working set sizes that only allow one or two segments in memory. These are shown for educational reasons only. Some small programs can run quickly with three or four segments in memory, but most programs require five or more segments for reasonable performance. The greatest performance increase occurs when the last disc-resident segment is included in the working set.

Table 4-1. Working Set Size and Execution Time on E-Series

System Type : E-Series
Disc Type : 7905

RANGE OF WORKING SET	RANGE OF EXECUTION TIME	COMMENTS
1 – 5	not applicable	VM83 error — Insufficient Working Set
6 – 10	655 – 374	One segment in memory. Performance improves rapidly as the amount of working set available for the user program and data increases from several hundred to several thousand words. Performance does not improve as dramatically as additional segments are added, because a larger amount of space is reserved for the user before more segments are added.
11 – 30	373 – 372	Still one segment in memory.
31 – 51	279 – 277	Two segments in memory.
52 – 72	200 – 197	Three segments in memory.
73 – 95	143 – 141	Four segments in memory.
96 – 119	87 – 85	Five segments in memory.
120 – 143	39 – 39	Six segments in memory.
> 143	< 3	All seven execution segments in memory. Additional increases in the working set are used for the user program and data.

Table 4-2. Working Set Size and Execution Time on F-Series

System Type : F-Series
Disc Type : 7925

RANGE OF WORKING SET	RANGE OF EXECUTION TIME	COMMENTS
1 – 5	not applicable	VM83 error — Insufficient Working Set
6 – 10	492 – 223	One segment in memory. Performance improves rapidly as the amount of working set available for the user program and data increases from several hundred to several thousand words. Performance does not change as dramatically as additional segments are added, because a larger amount of space is reserved for the user before more segments are added.
11 – 30	222 – 220	Still one segment in memory.
31 – 51	164 – 162	Two segments in memory.
52 – 72	117 – 115	Three segments in memory.
73 – 95	83 – 80	Four segments in memory.
96 – 119	49 – 49	Five segments in memory.
120 – 143	23 – 23	Six segments in memory.
> 143	< 3	All seven execution segments in memory. Additional increases in the working set are used for the user program and data.

Table 4-3. Working Set Size and Execution Time on A600

System Type : A600
Disc Type : 7908

RANGE OF WORKING SET	RANGE OF EXECUTION TIME	COMMENTS
1 – 25	not applicable	VM83 error — Insufficient Working Set
26 – 30	887 – 511	One segment in memory. Performance improves rapidly as the amount of working set available for the user program and data increases from several hundred to several thousand words. Performance does not improve as dramatically as additional segments are added, because a larger amount of space is reserved for the user before more segments are added.
31 – 50	510 – 508	Still one segment in memory.
51 – 71	376 – 373	Two segments in memory
72 – 92	265 – 260	Three segments in memory.
93 – 115	188 – 183	Four segments in memory.
116 – 139	112 – 110	Five segments in memory.
140 – 163	51 – 50	Six segments in memory
> 163	< 3	All seven execution segments in memory. Additional increases in the working set are used for the user program and data.

Table 4-4. Working Set Size and Execution Time on A700

System Type : A700
Disc Type : 7906

RANGE OF WORKING SET	RANGE OF EXECUTION TIME	COMMENTS
1 – 25	not applicable	VM83 error — Insufficient Working Set
26 – 30	620 – 348	One segment in memory. Performance improves rapidly as the amount of working set available for the user program and data increases from several hundred to several thousand words. Performance does not improve as dramatically as additional segments are added, because a larger amount of space is reserved for the user before more segments are added.
31 – 50	348 – 347	Still one segment in memory.
51 – 71	261 – 259	Two segments in memory.
72 – 92	186 – 184	Three segments in memory.
93 – 115	133 – 130	Four segments in memory.
116 – 139	79 – 78	Five segments in memory.
140 – 163	36 – 36	Six segments in memory.
> 163	< 3	All seven execution segments in memory. Additional increases in the working set are used for the user program and data.

Table 4-5. Working Set Size and Execution Time on A900

System Type : A900
Disc Type : 7912

RANGE OF WORKING SET	RANGE OF EXECUTION TIME	COMMENTS
1 – 25	not applicable	VM83 error — Insufficiant Working Set.
26 – 30	412 – 234	One segment in memory. Performance improves rapidly as the amount of working set available for the user program and data increases from several hundred to several thousand words. Performance does not improve as dramatically as additional segments are added, because a larger amount of space is reserved for the user before more segments are added.
31 – 50	233 – 232	Still one segment in memory.
51 – 71	173 – 172	Two segments in memory.
72 – 91	123 – 122	Three segments in memory.
92 – 114	87 – 86	Four segments in memory.
115 – 138	53 – 52	Five segments in memory.
139 – 162	23 – 23	Six segments in memory.
> 162	< 1	All seven execution segments in memory. Additional increases in the working set are used for the user program and data.

The default virtual size of 500 for the BASIC Executor is large enough to handle many large programs. Programs that use very large arrays need a larger virtual size, and for small programs, the virtual size can be reduced to conserve disc space.

For a system of type:	When the WS has been set to the absolute minimum, which is:	The absolute minimum VS which will not result in errors is:
E	6	257
F	6	257
A600	26	259
A700	26	259
A900	26	259

The minimum VS values shown above are provided for reference only. In actual use, never set the virtual size to fewer than 280 pages. The minimum configurations shown above only allow for sufficient memory (about 400 bytes) to run the user program:

```
99 END
```

with the minimum standard file configuration (all standard files default to the terminal).

When transporting a BASIC program from a 98xx Desktop Computer to the HP 1000, add the amount of user RAM required to execute on the Desktop Computer (2K bytes equals one page) to the minimum virtual size shown above (but not less than 280). Set the virtual size of the BASIC Executor to a number greater than this sum.

When developing new BASIC programs on the HP 1000, multiply the expected size of the largest program (in lines) by 22, divide by 1000, and add the result to the minimum virtual size shown above. Set the virtual size of the BASIC Executor to this number or higher (but not less than 280).

For example, to support a 3000-line program on an A900:

```
3000 x 22 = 66000
66000 / 1000 = 66
66 + 259 = 325
```

The upper limit on the virtual size is 65K pages for RTE-6/VM systems and 6144 pages for RTE-A systems.

When increasing the working set size from the default, increase the 280-page minimum virtual size by the same amount. For example, after adding 20 pages to the default working set size, do not set the virtual size lower than 300 pages, or a VM82 error can result.

Programs that use extraordinarily large data structures (such as long strings or large, multidimensional arrays) can average more than 22 words per line, but the vast majority of programs require less. When BASIC applications programs are running successfully, experiment with the virtual size (substituting other values for 22 in calculating it), but never set it less than 280.

Each page in the virtual size requires eight 128-word blocks in the VMA file.

Compiler Virtual Size and Working Set Size

The compiler uses 500 pages. The virtual size is fixed and does not affect compilation speed.

For RTE-6/VM, set the compiler working set size to 70 pages. Increasing it beyond 70 pages does not significantly improve performance. With a 70-page working set, CBASC requires a 98K-word partition.

For RTE-A, set the working set size to 90 pages. Increasing it beyond 90 pages does not significantly improve performance. With a 90-page working set, CBASC requires a 120K-word partition.

The LINK command files provided with BASIC use the default working set size. Increase the working set size to improve performance.

Program priority, system load, and disc speed also affect performance.

RINTR Priority and Memory Requirements

The priority of RINTR can affect interrupt response time. Set the priority to a value that is appropriate to your system and application. Also, for faster response by RINTR, it is recommended that it be assigned to a reserved seven-page partition.

HEWLETT PACKARD
HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014

LINK Support Document

Revision 2302 of the LINK loader, which supports the EMA/VMA memory management scheme used by compiled BASIC/1000C programs, will be supplied with the BASIC/1000C product until revision 2326 for RTE-A systems and the 2340 PCO for RTE-6/VM.

With an RTE-A system, revision 2326 or later, use the LINK loader provided with the system instead of the LINK loader provided with BASIC/1000C.

The LINK parts supplied with BASIC/1000C are:

%LINKA	92077-16001
%LINKB	92077-16002
%LINKC	92077-16003
%LINKD	92077-16004
%LNKRA	92077-16107
%LNKR6	92077-16108
%LNKDA	92077-16112
%LNKD6	92077-16113

Installation of LINK revision 2302 is a step in the installation process for BASIC/1000C, and is described in the BASIC/1000C Installation and Configuration Guide (Part No. 92857-90002).

MANUAL PART NO. 92857-90004
Printed in U.S.A. March 1983
E0383