Ethernet-CDReader

VMS Booting from an Ethernet-CDReader DRAFT Specification

DIGITAL CONFIDENTIAL

Revision/Update Information: X1.0.0
28–July–1989

Abstract

This document is not final and may change.

Review Team

Names\titles of reviewers

Written by:

Barbara C. Leahy—VMS Development

Issued by:

VMS DEVELOPMENT—ZK3-4

Review Team:

Elliott Drayton—VMS Development

Digital Equipment Corporation

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright ©1989 by Digital Equipment Corporation

All Rights Reserved.
Printed in U.S.A.

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	UNIBUS
DEC/CMS	EduSystem	VAX
DEC/MMS	IAS	VAXcluster
DECnet	MASSBUS	VMS
DECsystem–10	PDP	VT
DECSYSTEM–20	PDT	
DECUS	RSTS	digital™
DECwriter	RSX	

This document was prepared using VAX DOCUMENT, Version 1.1

CONTENTS

Preface

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digitial Equipment Corporation assumes no responsibility for any errors that may occur in this document.

The information in this document does not describe any program or product currently available from Digital Equipment Corporation. Nor does Digital Equipment Corporation commit to implement this specification in any program or product. Digital Equipment Corporation makes no commitment that this document accurately describes any product which it might ever make.

Associated Documents

1. Mariah CDreader CLIENT DESIGN SPECIFICATION
2. Ethernet-CDReader An Overview

Change History

1 Product Overview

1.1 Product Abstract

This specification describes the operational details of booting a Mid-range system supporting Ethernet-CDReader code in the console from an Ethernet-CDReader. This specification details the assumptions and guidelines used by VMB in connection with the Ethernet-CDReader boot code, and conditions for passing control between the modules in the boot path.

1.2 Product Audience

This specification is intended for any operating systems software group intending to boot an image akin to Stand-alone backup from Ethernet-CDReader.

2 Product Goals

Ethernet-CDReader provides a method to load the Stand-alone backup type image from a Compact Disk over the Ethernet wire. As such it supports the company's goal of having all media distributed on compact disks.

Ethernet-CDReader consists of a RRD40 Compact Disk drive on a KA420 CPU platform with the LANCE chip as the Ethernet interface, plus at least four megabytes of memory running a stand alone version of the VMS operating system.

2.1 Environments

The Ethernet-CDReader boot driver is part of the VMB code and is loaded in with VMB at boot time by the console program.

2.2 Non Goals

It is not a goal of the Ethernet-CDReader VMB software to be able to boot an operating system image or image requiring write capabilities to the Compact Disc.

It is not a goal to be able to support customer boot servers within the context of VMB. In particular, bootstrapping of compact disk on any platform not specified is not supported.

3 Functional Description

An image or system console uses Ethernet-CDReader drivers to read in data from a remote compact disk. The data read in includes the modules necessary to bring up Stand-alone backup onto a system. The first image to use the Ethernet-CDReader drivers is VMB. VMB uses the drivers to read in secondary loaders such as Sysboot.

VMB itself is read in by the console code drivers, and those drivers and their actions are not within the scope of this document.

VMB builds the RPB (Restart Parameter Block) and secondary bootstrap argument list, finds the boot device, loads the secondary bootstrap from the boot device and transfers control to the secondary bootstrap image. VMB also provides bootstrap device drivers, which may be used by the secondary loader. The VMS secondary bootstrap image, Sysboot, continues reading in the rest of the operating image using the boot drivers loaded by VMB.

Ethernet-CDReader requires extra information not provided in a typical boot path operation. These departures from typical operating methods are described in this document.

3.1 VMB Inputs

On input to the Primary Bootstrap (VMB) the processor is running at IPL 31 on the interrupt stack with memory management disabled. The registers are as follows:

```
R0 = Boot device type code
R1 = Boot device Bus address
R2 = Controller letter designator(not necessary )
R3 = address of Available Target Identifier (built by console routine)
R4 = logical block number to boot from if Bit 3 is set in R5
R5 = software boot control flags

R10 = original PC at entry to VMB (Halt PC)
R11 = original PSL at entry to VMB
AP = Halt code
SP = address of 256KB memory block + 600 hex (first location in VMB)
```

For Ethernet-CDReader booting the first two pages of memory are reserved for a structure used by the boot driver. The RPB, Restart Parameter Block is loaded into the third page of memory, with VMB loaded into the next page.

The RBP format is shown below:

```
offset 0        +---------------------------------------+
                + Available Target Identifier           +
                +                                       +
offset 400      +---------------------------------------+
                + to be used as RPB                     +
offset 600 (hex)+---------------------------------------+
                + VMB Boot Code                         +
                +                                       +
vmb end label   +                                       +
                +---------------------------------------+
                + remainder of 256KB good memory        +
                +  to be used for SCB, stack            +
                +  and loading secondary bootstrap      +
                .                                       .
                .                                       .
                +---------------------------------------+
                .                                       .
                .                                       .
                .                                       .
                .                                       .
                .                                       .
                .                                       .
                .                                       .
                .                                       .
high memory     +---------------------------------------+
                . pfn bitmap                            .
                .                                       .
                .                                       .
                +---------------------------------------+
                top of physical
                memory
```

As can be seen from the RPB layout Ethernet-CDReader uses the lower two pages of memory for a structure called the Available Target Identifier (ATI). This structure contains information from the server about the service being used for the boot path. Contained in this structure is the destination Ethernet address, the service name, service characteristics and more. The VMB level boot driver uses the destination Ethernet address, and destination server and service name to access blocks of data from a remote compact disk on a server box. The service name, destination Ethernet address, and server name are required data.

The VMB code uses the R3 input register to point to the ATI. By default the starting address of the ATI is 400 hex less than the address of the RPB.

3.2 Booting

The standard VMB used for booting with disks, tapes, and MOP requests is used. There is no element of downline loading performed in this boot. VMB has a network boot path and a disk boot path. Ethernet-CDReader uses the disk boot path of VMB to access a remote disk on a Ethernet box, and some of the concepts of the network boot path for loading of the boot drivers and run-time drivers. The conceptual differences between a typical local disk boot and an remote Ethernet disk boot are discussed in this section.

3.2.1 VMB Changes

VMB has changes in it for the new server device types, similiar as to when other devices are added. Conditional branches are added to VMB to branch over the initialization of the Ethernet hardware when it is necesary to do so. These branches are similiar to what exists today for the already existing Ethernet devices on the particular hardware configuation of the VAX being booted from. Other than this branch code no other code has been added to the VMB.mar file, except for fixes to branching out of range.

3.2.2 BOOT Device Name Types

The device names signify a combination of a server and a specific Ethernet device type. This is necessary to indicate what set of drivers is to be loaded onto the operating system and used as the system drivers. The names have the format: BTD$K__SERVER__ethernet_device_nmenonic. This name indicates that the server drivers: (LAD, LAST) are to be loaded along with the correspondinbg Ethernet driver for the hardware being used.

3.2.3 PORT/CLASS Boot Drivers

The boot driver for this application has a PORT/CLASS architecture. There is one common class driver, and multiple port drivers. The Full Ethernet boot drivers comprise the port drivers and the LAD/LAST BEBTDRIVR is the class driver. The Ethernet boot drivers can normally be used on their own and have full boot driver capability of handling requests. The class boot driver cannot work without a port boot driver.

VMB loads one driver and relocates one driver. As such, the class driver and port driver must appear as one driver to VMB even though they are two distince drivers. In order to make this happen, the class driver must locate the port driver, relocate it right below the class driver, and increase the size of the class driver to account for the port driver, thereby appearing as one driver to VMB. VMB never knows that it is relocating two drivers instead of one.

The class driver also finds the BDT table of the Ethernet boot driver and copies that table into a table local to the class driver. The class driver uses the offsets in the local copy ot the Ethernet boot table to find the addresses of routines to call.

Because VMB does not directly support port/class boot drivers there are some restrictions to its implementation. To date the restriction for port/class boot drivers exists in the link command for forming VMB.exe. This restriction is that the class driver must appear before all port drivers in the order that the code is linked. This is required for the class driver to be able to find the port driver. The restriction can be lifted when either VMB supports the port/class interface, or the interface to the action routines changes to pass in the address of the top of the boot device descriptor table.

3.2.4 PORT/CLASS Driver Calling Interface

The class driver is responsible for setting up the calling interface the Port driver needs to perform its job. All Port drivers must use the same calling interface to prevent special casing of the calling interface. The calling interface is:

```
ACTION ROUTINE:

The class driver does not call the Ethernet drivers action routine
if any exists.

INITIALIZATION ROUTINE:

The class driver preserves the registers sent to it and passes the
information to the port driver:

  R9 points to the RPB
  AP points to argument list

The class driver also sets up R7 to point to the SCB for the port driver.
The port driver must return with a RET instruction.

MAIN DRIVER ROUTINE:

The class driver passes the argument pointer and the data the
port driver needs.   The port driver must expect that the calling
information it is receiving is equivalent to a BOO$QIO call.

The port driver should make no assumptions about the contents of
any register other than R9, and the AP.  It should pick up the
byte count of the transfer from the location off of the AP and
not from registers.  It must also pick up the mapping mode from
the AP and not from the VAX registers.

The format of the AP sent by the class driver to the port
driver on a write request is:

 ARGCNT number of arguments, (6)
 BUFFER  address of the buffer message to send
 SIZE    size of buffer to send not including the
  Ethernet header
 LBN Pointer to a parameter address structure with the format
  outlined below:
```

```
 31                                                      0
 +-------------------------------------------------------+
 |    Protocol              | Pad mode                   |
 +-------------------------------------------------------+
 |  Destination Ethernet Address                         |
 +---------------------------+                           +
                             |                           |
                             +---------------------------+
```

FUNC IO function to perform, IO$_READLBLK OR IO$_WRITELBLK
MODE Mapping mode

The format of the AP sent by the class driver to the port driver on a read request is:

ARGCNT number of arguments, (6)
BUFFER address of the buffer within the class driver to receive the message
SIZE size of buffer to receive not including the Ethernet header
LBN Pointer to a parameter address structure with the format outlined below:

```
 31                                                      0
 +-------------------------------------------------------+
 |    Protocol              | Pad mode                   |
 +-------------------------------------------------------+
 |  Destination Ethernet Address                         |
 +---------------------------+                           +
                             |                           |
                             +---------------------------+
```

FUNC IO function to perform, IO$_READLBLK OR IO$_WRITELBLK
MODE Mapping mode

NOTE: at present, the port driver must support a CALL interface at its driver routine, and supply a RET instruction at its end. This is a slight departure from other calling interfaces. This issue will be readdressed at a later date.

DISCONNECT ROUTINE:

R9 - RPB

4 OPERATIONS

VMB uses the drivers to read in Sysboot. Sysboot is responsible for locating the system drivers and reading them into memory. It is necessary to have three drivers read into memory to run Stand-alone Backup. The drivers are:

LASTDRIVER.EXE
DADDRIVER.EXE
Ethernet driver used for the boot hardware

To load all three drivers, a VMB version of 14 or greater is required with the auxillary driver list structure. In the auxillary driver list structure is the name of the Ethernet driver used by that system. It is the responsiblity of the class boot driver to load the auxillary driver list longword pointer with the address of the counted string containing the driver name. The class driver must perform this in its initilization routine if the string is within the driver, or it can perform it in the action routine if the string is located in a fixed accessible portion of memory.

5 SYSBOOT Changes

This boot path will utilize the code from other projects that support booting from a write-protected device. When necessary, flags or appropriate action will be taken to indicate that the device being booted from is write protected. These actions change as sysboot changes to accomodate this boot method. To date, the changes for other devices have involved adding code to watch for a flag indicating that the boot device is write protected. The Ethernet-CDReader will also make use of these code paths and set the flag to indicate its status.

6 SYSLOA changes

For any of the sysloas that require special casing for the boot device type, code will be added. For other sysloas that do not require this code, no changes are necessary.

The Ethernet-CDReader is a read-only device and as such does not allow crash dumps; therefore no changes will be made to the code which handles bugcheck operations.

7 OTHER changes

As the run-time drivers evolve and change, any necessary changes to the boot drivers will also take place. Currently the run-time drivers expect a controller letter of D and a unit number of 1. The class driver will take care of setting these up in the RPB at this time. The official device name for these devices is still transitional and needs to confirmed.

As other updates evolve this document will also evolve and change to reflect those updates.