PRELIMINARY

EMULOGIC
EMUNET SYSTEM
GUIDE

PRELIMINARY

Order Number: MAN-1012-00

CONTENTS

PAGE

CHAPTER 1

INTRODUCTION

This manual contains information necessary to install and use the Emulogic EMUNET system. The system consists of high-speed data link (HSL) hardware and virtual disk system (VDS) control software and files. This system provides virtual disk operation for the RT-11 based Emulogic ECL-3211 Microprocessor Development System (MDS) stations connected to an RSX-11M or VAX/VMS based host computer. Installation and operation of the system requires no knowledge of programming languages and only general knowledge of the RT-11, RSX-11, or VMS operating systems*.

OVERVIEW

In the Emulogic EMUNET system environment, RT-11 based computers -- called satellites -- use the resources of a physical disk located on an RSX-11M or VMS based computer -- called the host. The satellites are connected to the host by a high-speed communications network. A single host can support up to 4 EMUNET system lines (subnets) with as many as 15 satellites per line. This network, together with several software components, comprises the virtual disk system.

Each satellite operates as though it had a local physical disk. A special RT-11 handler on each satellite routes disk I/O requests to the host using the high-speed link network. Software on the host services the I/O requests, passing them to one of its own disks. Thus, the satellite virtual disks reside on the host physical disk and are connected to the satellite through the high-speed network.

*Users installing HSL network hardware on VAX 11/7XX host computers will require the assistance of a Digital Equipment Corporation (DEC) field service technician. To ensure expeditious installation, consult Chapter 3 and make an appointment with DEC Field Service to make the necessary backplane preparations.

The EMUNET system system uses Emulogic-developed high-speed link hardware to support the network communications. Special Emulogic software provides the simulation required to create the virtual disk imaging. The hardware and software components are described in the following sections.

THE HIGH-SPEED LINK

The EMUNET system high-speed link (HSL) is a communications device that can transfer information between satellite and host at a rate of one megabaud (one million bits per second). The device operates using direct memory access (DMA) and runs on any LSI-11, PDP-11, or VAX processor.

A number of satellites can be connected together using a single coaxial cable. The satellites are connected in what is known as "multi-drop topology." The HSL hardware plugs directly into an LSI-11 bus (Q-BUS) and communicates over a 75-ohm coaxial cable. The cable length can be as long as 30,000 feet, depending on the type of cable used.

VIRTUAL DISK IMAGING

The Emulogic EMUNET system provides abundant and efficient mass data storage in an installation where several ECS-3211 MDS stations are used. These stations, called satellites, can share the disk resources of a larger RSX-11M or VMS based computer system which serves as a network host. The network can consist of one or more primary lines (subnets) with up to 15 satellites per subnet. Each satellite can address the data on the host's disks as though it were calling information from its own storage. For this reason, we refer to this method of data storage and access as "virtual." Figure 1-1 presents a diagram showing a typical EMUNET system configuration.

Figure 1-1: Sample network configuration

EMUNET SYSTEM OPERATION

When a satellite in the EMUNET system network has a disk request, a special RT-11 handler in the satellite routes the disk request to the host over the high-speed link network. Software on the host services the request, from one of its own disks. Essentially, the satellite virtual disks reside on a host physical disk, and are connected to the satellite through the high-speed network.

Each satellite uses disk resources of the host disks which contain the physical representations of the virtual disk volumes. Since these virtual disk volumes contain the RT-11 operating system and file structure, there is no significant difference between using the ECL-3211 MDS as a stand-alone station or as a satellite on the EMUNET system network. However, you should observe the following minor exceptions when operating in the EMUNET system environment:

o The 8 virtual disk devices are referred to as VS0: . . . VS7: ,

o Read-only volumes must not be assigned to a system device,

o The storage space allocated to each virtual disk unit can be specified according to need. All units do not need to have the same storage capacity.

CHAPTER 2

EMUNET SYSTEM COMPONENTS

The Emulogic Multi-User system is a fully integrated package of hardware and software components. The hardware forms the physical connections between satellites and host computer. The software forms the logical interfaces at the satellite stations and at the host. Together, the hardware and software allow transparent use of host disk resources by the satellites and management of virtual files at the host.

HARDWARE COMPONENTS

The Emulogic Multi-User system (HSL) hardware kit is a custom package. A pair of HSL boards, a length of 75-ohm coaxial (coax) cable, and a set of bootstrap ROM chips are provided for each satellite ECL-3211 MDS. For the host, the hardware supplied depends on the configuration of your system; UNIBUS and Q-BUS versions are available. The host hardware kit includes a pair of HSL boards, 75-ohm cabling, and -- when necessary -- appropriate UNIBUS to Q-BUS connection hardware. A complete description of the hardware components is included in the installation section of this guide.

THE HSL BOARDS

The two HSL boards are designated as Direct Memory Access (DMA) and Data Communication (DATACOM). These two units fit into sequential slots on a Q-BUS (LSI-11 bus) backplane. Because of its high-speed bus communication activity, the DMA board must be installed electrically closer to the host CPU. The DMA and DATACOM boards are linked together by a short 40-wire cable or "jumper."

The DMA board provides intra-system data transfer (the path from the satellite or host memory to the HSL DATACOM board) using Digital Equipment Corporation's (DEC) direct memory access technique. The DATACOM board processes virtual disk requests for transmission over the bi-directional

HSL cable. The DATACOM board also processes data received over the link and passes it to the DMA. Together with the cable network, these boards comprise the physical high-speed link.

THE CABLE NETWORK

A network of 75-ohm coaxial cable forms the physical tie between satellite and host. A length of cable, electrically terminated at each end, extends along a route on which the host computer and one or more Emulogic ECL-3211 satellites reside. The host and satellites are tapped into this line using standard BNC "T" connectors and plugs. This cable configuration is capable of carrying transmissions at up to 1 million bits per second (1 megabaud). Depending upon the type of host, 1 to 4 such lines can be supported, each with as many as 15 satellites.

Satellite Hardware Bootstrap

ECL-3211 MDS bootstrapping is initiated by a ROM coded program. Normally, the ROM program calls the bootstrap from a local disk device. However, when satellites do not have their own local disk storage they must rely on the virtual disk resources for all disk requirements. To resolve this situation, Emulogic provides a special set of bootstrap ROMs for cold starting a satellite from its virtual disk. These special ROMs are used in place of the standard DEC bootstrap ROMs. At satellite start-up, this program calls and executes the RT-11 bootstrap from the virtual volume on the host.

SOFTWARE COMPONENTS

The Multi-User system supports identical functions whether hosted by a PDP-11 or VAX-11 computer system. The software components for these two systems are, however, necessarily different in content and structure. The software components used on the satellites and the network protocols are identical regardless of host type. The host is transparent to the satellites, allowing for easy host upgrades as facilities mature.

The Multi-User system software has 2 primary tasks. At the satellite station, software must convert standard RT-11 input/output (I/O) requests for disk-stored data to virtual disk requests, pass output requests and data to the HSL channel, and receive data responses to input requests from the HSL channel. At the host, software must receive virtual disk requests from the satellites, convert the requests to logical disk locations, and fetch or store data appropriately. When a satellite request is for data retrieval, the host software must also package the data and direct it back to the requesting station.

SATELLITE RT-11 SOFTWARE COMPONENTS

There are 2 software components that support the Emulogic ECL-3211 (satellite) end of the EMUNET network. These components concern the I/O request managment, satellite bootstrapping from virtual volumes, and network date/time synchronization functions.

Satellite RT-11 Handler

The RT-11 handler, VS.SYS, receives all disk I/O requests under the MDS's RT-11 operating system. In effect, the handler is a virtual disk device in that from the system's and programmer's point of view, disk requests are handled as if using the satellite's own disk resources. The handler appears as a 8-unit disk and permits programs on the satellite to do read and write operations on any of these disk units. The handler transforms these operations to requests for reads and writes on the host and passes these, with associated data, to the high-speed link.

From a programming standpoint, the RT-11 handler appears identical to any physical disk handler, and there are no special programming requirements needed to use it. The size of the virtual disk volume is determined by the size of the virtual disk file on the host which contains it.

Network Time and Date Synchronization

Usually it is desirable that all computer systems connected in a network use the same date and time references. In the EMUNET environment, this goal is met through a utility program: SETDAT. This program queries the host for the current date and time and then initializes the ECL-3211 MDS clock with these values. You can invoke this program when a satellite is booted from the host by placing the instruction "R SETDAT" in the STARTS.COM file.

MULTI-USER HOST SYSTEM SOFTWARE

To direct communications over the HSL network, the Emulogic Multi-User host system is provided with a set of controlling software programs. These programs are largely responsible for virtual I/O operations of the system and are thus part of the VDS software. Since there are significant differences in the operation of DEC's RSX-11 and VMS operating systems, Emulogic has created specific VDS software for each of these operating system environments.

RSX-11M SOFTWARE COMPONENTS

Two major programs support the host end of the EMUNET network under an RSX-11M host.

The File Server Task

The RSX-11M file server task, VIP, is responsible for servicing disk I/O requests. For each satellite, an image of VIP is installed under a different name. Each VIP image reads and writes to files on a host physical disk. These files represent the virtual disk volumes of the satellite which it services. Because there is one image (task) serving each satellite, the satellite virtual disks can operate in parallel, with I/O requests from one satellite being serviced independently of requests from another.

The linkage between a satellite virtual disk and the actual file on the host is determined by the command line entered when each VIP task is initialized. An initializing command statement, each task is supplied with information identifying which satellite it serves, what files it should attach as virtual disks, and whether write-protected status applies. (VIP also has the ability to attach entire devices as virtual volumes. This provides the facility for a satellite to work directly to RT-11 structured physical volumes present on host disk drives.)

RSX-11M Driver

The RSX-11M driver, ZVDRV, operates the host Multi-User system hardware. The driver processes I/O requests passed from the file server task (VIP). On command from VIP, the driver can receive information from an RT-11 satellite or send information to it. It is the function of the driver to relay information from RT-11 satellites to the VIP tasks.

VMS SOFTWARE COMPONENTS

There are three operating-system specific software packages that control the Multi-User system in the VAX/VMS environment. These packages direct internal data and address translation at the host and drive the HSL hardware. The components are described in the following sections.

Net Manager

Under a VMS operating system, VDS uses a small database and manager as the host software interface. This package, called the EMUNETMGR, receives packets, schedules the packets for processing and maintains the network data base. It is the central manager for all network operations on the VAX/VMS host. The database maintains a directory of all satellites, their requests, and other pertinent information.

The network manager module does not receive its direction from the packet. Instead, under program control, EMUNETMGR evaluates the information in each packet and, from it, determines the type of service required and calls the appropriate service routine.

VMS Driver

The VMS driver, ZVDRIVER, is an interface between the host HSL hardware and the VAX/VMS I/O subsystems (RMS and QIO). The driver allows a program to control the HSL through the RMS or QIO facilities. In particular, the driver passes packets back and forth between the network manager and the HSL network.

Management Utility

The network control program (ECP) is a utility program that allows the system manager to configure, control, and monitor the network system. It is this utility which allows you to specify and map virtual volumes for satellites. Refer to "Using the ECP" later in this guide for details.

HOST-INDEPENDENT SOFTWARE COMPONENTS

Besides the host system-specific sopftware packages described in the preceding sections, Emulogic's Multi-User includes two environment-independent utility programs. These programs simplify a variety of file manipulation and management tasks at the host level.

Virtual Disk File Transfer Utility

A virtual disk file transfer utility, VDX, runs on either RSX-11M or VAX/VMS hosts. This utility is used for transferring files in Files-11 format to virtual disk volumes, and *vice versa*. It is similar in function and operation to the file-exchange (FLX) utility. In addition to transfer functions, it can also initialize volumes, extend volumes, list directories, and write virtual disk boot blocks. The utility, therefore, permits access on the host to the virtual disk volumes to facilitate maintenance activities.

RT-11 Emulator

The RT-11 Emulator (RTEM-11), offered by DEC for RSX-11M and VMS, is totally compatible with EMUNET. Not only does it allow for the use of RT-11 utilities and development tools on the host machine, but it also has the ability to attach and directly use EMUNET virtual volumes. Because of this, RTEM-11 has applications in maintenance activities related to Multi-User files as well as facilitating the use of Emulogic software development tools on the host computer.

SUMMARY

Emulogic, Inc. provides a full range of control and utility software for its Multi-User system. Automated software provides transparent interaction between ECL-3211 satellite MDS stations and an RSX-11M or VAX/VMS host computer system. In addition, standard installation and utility packages allow complete system management and maintenance capability.

CHAPTER 3

INSTALLING EMUNET SYSTEM HARDWARE

Depending on the model of Digital Equipment Corporation computer system that is your host system, Emulogic has shipped you one or more hardware installation kits. The basic kit is intended for the Emulogic ECL-3211 Microprocessor Development System and all LSI-11 bus (Q-BUS) based host computers. The UNIBUS kit is for all UNIBUS based PDP-11 and VAX-11/7XX host computers.

The instructions below will guide you through the installation of EMUNET system hardware for any of these host and satellite configurations.

INSTALLATION EVENTS

Installation of the HSL VDS hardware consists of the following steps, although not necessarily in this order:

o Installing satellite hardware,
o Installing host hardware,
o Installing host software,
o Planning, running, and connecting network cabling.

These steps are described in detail in the following sections. You should read quickly through this information before beginning actual installation.

USING INSTALLATION INSTRUCTIONS

Follow the directions in "Board Installation" to insert the DMA and DATACOM boards in any Q-BUS backplane (ECL-3211 MDS, LSI-11 BUS, or auxilliary Q-BUS backplane). If your host computer is a UNIBUS based system, find the directions for connecting the bus converter, cabling, and auxilliary backplane assembly in "Installing a Bus Converter in a Unibus Host". When you have designed your network cable (75-ohm coax) layout, follow the directions under "Coaxial Cable Installation" to connect the cables to ECL-3211 satellites and to the host.

INSTALLING HSL HARDWARE IN YOUR Q-BUS SYSTEM

In the kit required for a Q-BUS HSL installation, Emulogic supplies:

- 1 DMA and 1 DATACOM board,
- 1 40-wire ribbon cable,
- 1 20-foot 75-ohm coax cable and 1 BNC "T" connector, or
- 1 100-foot 75-ohm coax cable, 2 BNC terminators, and 1 BNC "T" connector.

Kits for installation in the Emulogic ECL-3211 system use the 20-foot cable and "T" connector. Kits for PDP-11/23 hosts use the 100-foot cable, cable terminators, and "T" connector. The components of the Q-BUS installation kit are shown in Figure 3-1.

Figure 3-1. Q-BUS EMUNET system hardware kit.

BOARD INSTALLATION

Install the boards in your Q-BUS backplane according to the following steps. Check off each step ( ) as you complete it.

1. ( ) Log off the system and turn off the main power switch.

2. ( ) Disconnect the power cord from the A/C power source.

3. ( ) Open the back of the BA11-NE (or equivalent) cabinet containing DEC processor and other system boards.

4. ( ) Prepare space for the HSL boards. Installation requires at least 2 sequential slots on the bus. (You may have to reposition the 2 quad-sized Emulogic Map and Control boards of the ECL-3211 MDS to obtain the necessary space.) If three slots can be freed, this will simplify the installation.

5. ( ) Slide the DMA board into the upper open slot on the bus and seat it firmly in the backplane connectors.

6. ( ) Slide the DATACOM board into the lower open slot (leave an empty slot between the two boards if space is available) and seat it firmly in the backplane connectors.

7. ( ) Link the DMA and DATACOM boards by inserting the 40-wire jumper ribbon cable.

   a. ( ) Locate the jumper ribbon and positions it with the red-marked wire to the right.

   b. ( ) Insert one plug in the 40-pin connector on the DMA board.

   c. ( ) Insert the plug at the free end of the jumper in the 40-pin connector on the DATACOM board.

8. ( ) Route the thin white coax (pigtail) with the BNC connector so that it will exit from the rear of the cabinet without being crimped when the door is closed.

9. ( ) Close the cabinet rear door.

VDS BOOTSTRAP HARDWARE

Emulogic provides a small hardware kit which allows your satellite ECL-3211 station to obtain a bootstrap program from its private virtual disk. If you have purchased your ECL-3211 as a virtual disk satellite, it has already been factory configured to boot from the EMUNET network. If your ECL-3211 has local mass storage as well as access to VDS, it is not necessary to provide for VDS booting. If your Emulogic MDS stations have not been purchased specifically for use in the EMUNET environment but will not have local disk devices, you need to install the VDS bootstrap hardware kit.

The VDS bootstrap kit consists of two read only memory (ROM) chips which contain a small (256-word) bootstrap loading program. The two ROMs fit into sockets on the MXV-11 board; in addition several jumper wires must be reconfigured. Assuming that the MXV-11 is initially strapped to DEC factory configuration, the following steps describe the installation of the VDS bootstrap ROMs:

1. ( ) Insert the ROM chip labeled "HIGH" into socket E67 (the outer socket).

2. ( ) Insert the ROM chip labeled "LOW" into socket E57, (the inner socket).

3. ( ) Remove all jumpers from J33-J40 except J33 to J32.

4. ( ) Connect J33 to J37, J33 to J38, and J39 to J40.

5. ( ) Remove all jumpers from J29.

6. ( ) Connect J15 to J29.

INSTALLING A BUS CONVERTER IN A UNIBUS HOST

If your EMUNET system host computer is in the PDP-11 or VAX-11 families, it uses a UNIBUS design which is not directly compatible with the DMA board. Emulogic has included in your hardware installation kit:

- o 1 auxilliary Q-BUS backplane
- o 1 set of HSL boards (DMA and DATACOM);
- o 1 40-wire flat jumper cable (12-inch);
- o 1 quad-sized UNIBUS translator board (for host UNIBUS) and 1 dual-sizd translator board (for auxilliary Q-BUS);
- o 2 10-foot, 40-wire ribbon connector cables;
- o 1 100-foot 75-ohm coax cable, 2 75-ohm terminators, and 1 "T" connector.

Because host hardware configurations differ, the Q-BUS backplane assembly you receive may not be identical to the one shown in Figure 3-2. However, the unit for your system is similar, and the following discussions and instructions apply to any Emulogic-supplied backplane configuration.

BACKPLANE JUMPER RECONFIGURATION

To use the Emulogic-supplied UNIBUS/Q-BUS interface, an alteration of the UNIBUS backplane jumper wiring must be made. Since this is a small but intricate procedure, we strongly recommend that you contact your Digital Equipment Corporation service technician to assist with this task.

UNIBUS Jumper Reconfiguration Procedure

On current production backplanes, DEC places a wire-wrap jumper from pin CA1 to pin CB1 of each slot to preserve the "daisy chain" continuity of the non-processor grant (NPG) signal. On the slot to be used for the DMA board only, you must remove this jumper wire. This is the only preparation required for HSL hardware installation in UNIBUS host systems.

Q-BUS CONNECTION

Install the HSL hardware components in the VAX 11/750 host according to the following steps:

1. ( ) Position the Q-BUS backplane assembly. (If this unit is to be rack mounted, perform this operation before continuing with the installation procedures.)

2. ( ) Shut down the host computer system.

3. ( ) When shut down is complete, turn off power to the disk drive devices.

4. ( ) Turn off power to the processor.

5. ( ) Disconnect the processor's power cord from the A/C source.

6. ( ) Open the cabinet rear panel to gain access to the UNIBUS card cage.

7. ( ) In the re-jumpered slot prepared by your DEC field service technician, insert the quad-sized UNIBUS translator board (component side upward).

8. ( ) In first (top) slot of the auxilliary Q-BUS backplane, insert the dual-sized translator board (component side up).

9. ( ) Join the translator board in the host backplane to the one in the auxilliary backplane using the 2 10-foot ribbon cables. Be sure to match the polarity of the cables as you install each one by keeping the red-marked wire to the right side on both boards.

   a. ( ) Connect "J1" of the host translator board to "J1" of the auxilliary backplane translator board.

   b. ( ) Next, connect "J2" to "J2" in the same manner.

10. ( ) Slide the DMA board in the slot directly below the translator board in the auxilliary Q-BUS backplane.

11. ( ) Slide the DATACOM board in the slot below the DMA (leave an empty slot between, if space allows) in the auxilliary backplane.

12. ( ) Connect the DMA and DATACOM boards using the short 40-wire jumper cable.

   a. ( ) With the red-marked wire on the right, insert one plug into the 40-pin socket of the DMA board.

   b. ( ) Again with the red on the right, insert the remaining plug of the jumper into the 40-pin socket of the DATACOM board.

9. ( ) Check that all boards and cables on both the UNIBUS and Q-BUS backplanes are fully inserted in their respective connectors and correctly located.

10. ( ) Route the backplane connector cables, the jumper cables, and the small white pigtail so that these wires will not be crimped when the cabinet door or panel is closed.

11. ( ) Check that the power switches for the processor and the auxilliary backplane are in the "Off" position.

12. ( ) Plug in the power cords for the processor and auxilliary backplane.

13. ( ) Switch the power on for the auxilliary Q-BUS backplane.

14. ( ) Switch the power on for the processor.

CAUTION: It is very important that you always switch on the auxilliary Q-BUS backplane before switching on the processor and switch off Q-BUS only after the processor has been shut off.

COMMUNICATIONS CABLE

The cable design for the high-speed link is very simple. It consists of a length of 75-ohm coax cable electrically terminated at both ends. "T" connectors provide junctions for the satellite locations. If you use standard RG 59/U cable, the cable must be less than a 5000 feet long. Any line can have as many "T" connectors as necessary up to a maximum of 16 (one for the host and one each for up to 15 satellites). Emulogic supplies 100 feet of primary cabling material, which is sufficient to support the high-speed link if the satellites are located in relatively close proximity to the host. However, if your system is distributed over a large area, you need to arrange for additional coaxial cable.

The high-speed link is not sensitive to the physical location of the host or satellites along the cable. It is therefore not necessary to have the host located at or near the end of the cable, nor is it necessary to locate the satellites in any particular order. This is because HSL source and destination addresses have nothing to do with physical layout.

CABLE INSTALLATION

To connect the host and the first satellite take one of the 75-ohm terminators and connect it to one the horn ends of a "T" connector. Connect the primary coax cable to the opposite horn. Connect the white coax pigtail from the high speed link DATACOM board on the host to the stem plug on the "T" connector. This completes the cable connections at the host end.

Run a free end of the primary coaxial to the first satellite station. Connect one horn of another "T" connector to the end of the primary coax cable. Connect the remaining 75-ohm terminator to the opposite horn of the "T" connector. Join the 20-foot satellite coax cable to the coax pigtail from the satellite DATACOM board. Connect the free end of the satellite coax to the stem of the "T" connector.

ADDING SATELLITES TO THE HIGH SPEED LINK

To add a satellite to an existing high-speed link it is only necessary to remove the 75-ohm terminator from the "T" connector at the satellite end of the cable, connect a length of cable in its place, add a "T" connector to the end of the new length of cable, connect the coax stub from the new satellites DATACOM card to the new "T" connector and connect the 75-ohm terminator to the new "T" connector.

CHAPTER 4

RSX-11M SOFTWARE PREPARATION

The Emulogic EMUNET system software has been supplied to you on an RL02 disk cartridge. Included in this package is a software loading utility program. This program both prepares the files that serve as virtual disk volumes for the satellites and loads the control software modules for EMUNET system.

THE SOFTWARE LOADING UTILITY

The EMUNET system software distribution kit for RSX-11M provides an interactive program to aid in the installation of the system's control software. This program is contained in the executable file, VDSKGEN.CMD.

By running the VDSKGEN command procedure you can select parameters describing the EMUNET system in your machine environment. Among other factors, you can choose the host disk device(s) to contain the virtual volumes as well as volume size and number. When this interacive program receives all necessary parameters, it task-builds the virtual disk software, installs it, creates the virtual volumes, initializes the virtual volumes with the RT-11 operating system, writes virtual disk bootstraps to the virtual volumes, and places the EMUNET system in operation.

VDSKGEN creates a private virtual disk volume for each satellite as well as a common virtual disk volume. When the system is initialized, each satellite is connected to its private volume as VS0: and the common volume as VS7:. Thus all satellites have access to the same common volume.

The private disk volumes contain RT-11 system programs for bootstrapping and normal development activities and work and storage space for the satellite user. The common virtual disk contains RT-11 utilities, emulation software, and other materials of a read-only nature.

It is necessary to prohibit write access to virtual volumes held in common. However, any number of satellites can have simultaneous read access. To ensure this condition, maintenance of common volumes is acheived at the host level through the use of the VDX utility or RTEM-11 (RT-11 Emulator). In this way the system manager can have control over the contents of the common volume while protecting it from modification by satellite users.

CALCULATING VIRTUAL STORAGE REQUIREMENTS

Before running VDSKGEN you should consider which device is to contain the virtual disk volumes, how many satellites are being supported, and how much storage to assign to each volume.

The virtual volumes are actually Files-11 files which contain the RT-11 volume structure internally. When they are created, each is assigned a certain amount of storage. This storage is expressed in terms of 512-byte disk blocks, specified at the time of file creation. Therefore, if you wanted to create virtual disk volumes which each contained the storage capacity of an RX02 floppy disk, you would allocate 1000 blocks to each. You may allocate any amount of storage per virtual volume up to the capacity of the device on which it is supported.

To configure the virtual disk system for your host, you need to determine the size required for each of the private volumes and for the common volume. Usually the common volume is allocated larger than any private volume, since it will contain a considerable amount of data. Remember that each satellite has its own private volume and thus the amount of storage you select for the private volumes will be added up and the sum added to the size of the common volume. Both VDX and RTEM-11 have dynamic facilites to allow for the extension of virtual volumes (making them bigger) so it is not necessary to create virtual volumes which are padded for future expansion.

The storage allocated for a virtual disk volume can be calculated in the following manner. Each disk block contains 512 bytes, so there are 2000 blocks per megabyte. If you have a high-speed link with five satellites and each satellite has a private volume containing 0.5 megabyte, then the allocation for each satellite would be 1000 blocks, and the total for all private volumes would be 5000 blocks (2.5 megabytes). To this you must add the storage allocated to the common volume. If the common volume requires 6000 disk blocks (3 megabytes) of storage, then the total allocation for the entire virtual disk system will be 11000 blocks (5.5 megabytes). You should check the amount of free space on the intended host disk device before allocating space to the virtual disk system to insure that there will be sufficient room to hold volumes of the size you have chosen.

LOADING AND EXECUTING THE VDSKGEN PROGRAM

The EMUNET system software distribution disk contains an interactive program file which installs VDS on the host processor. To use this utility, log on under a privileged account, mount the distribution volume, and copy the file "[1,301]VDSKGEN.CMD" to your UIC. You can then invoke the loading program with an "@VDSKGEN" command.

VDSKGEN prompts you for parameters of the host configuation and the size of the virtual disk volumes. It will then automatically copy the distribution control programs and files to the host and create the virtual disk system.

It may be necessary, on some systems, to use the disk drive device on which the VDS distribution has been mounted as the destination for the virtual volumes. After the distribution files are copied, you will be prompted to remove the distribution disk. At this time you can replace the distribution disk with the destination disk for your virtual volumes.

The following statements represent the command sequence for loading and executing the interactive software loading utility, VDSKGEN. Remember that to load the file from the distribution medium, you must log on under a priviledged account.

```
>MOU dev:VIRDSK
>PIP /NV/FO=dev:[1,301]VDSKGEN.CMD
>@VDSKGEN
```

In the sequence above, "dev" represents the device name of the unit holding the distribution medium. ">" represents the system prompt, and "@" represents the system command file prefix.

VDSKGEN Dialogue

Once you have started the VDSKGEN program, it produces a series of prompts for information about the host machine and the structure of the desired virtual disk volumes. With each prompt there is an explanation of the parameter options you can choose. When you enter a selection, the program accepts your response and displays the next prompt. If you press the RETURN key alone for any prompt, the system selects the default value for that parameter. When you have replied to all prompts in the sequence, the VDSKGEN enters generation mode.

Using the parameters you supplied during the prompt sequence, the program first task-builds, installs, and initializes the required host tasks. Next it creates the virtual volumes, loading each private volume with the RT-11 operating system. Finally it assigns the virtual volumes to the appropriate satellites and places the virtual disk system on line.

Phase II

Continue - proceeds with task-building of driver (zvdrv), utilities (VIP, VDX, KED) and clean-up (if selected). Default: yes, continue with building.

Chaining - are you continuing from the previous phase (Phase I)? If "no," system asks environment questions that follow. If "yes," program uses existing environment for next phase. Default: yes, continue with existing parameters.

Distribution device - (if not chaining) provide the device and unit names on which the distribution medium is mounted. Default: DL1: .

VDX device - (if not chaining) provide the device code for VDX.TSK created in Phase I. Default: sysdev:

VDX unit - (if not chaining) provide the unit code (UIC) for VDX.TSK as created in Phase I. Default: [sysuic]

VDSKGEN work device - disk device for temporary use during system building. Unless there are exceptions, accept default. Default: SY: .

Work unit - unit for private volume workspace. Unless there are exceptions, accept default. Default: [200,302] .

Common unit - unit for common volume workspace. Unless there are exceptions, accept default. Default: [200,303] .

Marker unit - unit for node marker (index) workspace. Unless there are exceptions, accept default. Default: [200,304] .

Delete - you can request the system to delete workspace files after task-building. Default: yes, delete workspace units.

Disk ready - the EMUNET system distribution volume (02) must be mounted in a drive. Check that the medium is ready and enter "Y" for yes at the terminal. Default: no, not ready.

Inhibit copy - if distribution files are already copied -- as from a previous EMUNET system generation -- this option cuts system building time. Default: no, copy all files.

Copy successful - unless an error occurs, accept default. Default: yes, files successfully copied.

Virtual device - code of device holding private virtual volumes. Default: DL1: .

Virtual unit - code of the unit containing virtual volumes. Default: [7,2] .

Destination ready - mount medium to hold virtual disks. Enter "Y" for yes at the terminal. Default: no, not ready.

Private volume size - number of 512-byte blocks per unit; 2000=1 mbyte. Default; 500.

Common volume size - number of 512-byte blocks per unit; 2000=1 mbyte. Default; 500.

How many satellites - number of ECL-3211 MDS workstations connected via system network. The range is from 1 to 15. Default: 8 satellites.

Sufficient space - does the available space on the requested device exceed the requirements of the virtual volumes? Default: yes, there is enough space.

Distribution backed up - you should always make a copy of distribution software. If this has been done, accept the default. If not done, enter "N" for no and make back-up copies at this time. Default: yes, back-up already made.

Change answers - this is the last question and last opportunity to change specifications for Phase II. If you enter "Y" for yes, the system will restart the Phase II cycle. Default: no changes.

Phase III

Chaining - allows use of environment from previous phase (Phase II) for current system building. Unless this is a restart, accept the default. Default: yes, use existing parameters.

Disk on virdev - is the medium containing the virtual volumes physically on the device designated as the residence of those volumes? If chaining, enter "Y" for *yes*. If you are *not* chaining, check disk. Default: no, disk not correctly mounted.

Partition - ZVDRV and VIP must be located in a convenient partition to assure correct operation of the EMUNET system. Unless and exceptional condition exists, accept the default. Default: "GEN" (general) partition available on all systems.

Inhibit ONLINE - the system can automatically put the EMUNET system into on-line mode after system building is complete. If you enter "Y" for *yes*, you must invoke the ONLINE.CMD command file (as described under "System Cold Start After Initial Installation" in this chapter) to place the system on line. Unless you have other set-up procedures to carry out, accept the default. Default: no, *do not inhibit* automatic calling of ONLINE.CMD .

SYSTEM COLD START AFTER INITIAL INSTALLATION

Since placing the EMUNET system on line involves the activation of several software elements, any system cold start requires executing a procedure to restore the EMUNET system to its on-line status. A command file called ONLINE.CMD, created by VDSKGEN, contains the code necessary to accomplish this task.

The ONLINE command can be executed manually or included in the system start-up command file for automatic execution. To run the ONLINE command, type the following statement:

```
@[1,54]ONLINE
```

CHAPTER 5

VAX/VMS SOFTWARE INSTALLATION

The Emulogic EMUNET system software has been supplied to you on media appropriate to your installation. Thus, you will have one of the following sets:

o 3 single-sided, single-density floppy diskettes (RX01),
o 3 DEC-standard cartridge tapes (TU58),

Included in this media set is a software loading command procedure. This program both prepares the files that serve as virtual disk volumes for the satellites and loads the control software modules for the EMUNET system.

THE VAX/VMS SYSTEM UPDATE UTILITY

The Emulogic EMUNET system software package for VAX-11 computers running under the VMS operating system is provided on media appropriate for your installation. Usually this consists of a set of 3 DEC-tape II cartridges. Each set contains programs, file builders, tables, and other files necessary to control and maintain the EMUNET environment. To provide for correct and convenient installation and initialization of the software, you can use the VAX/VMS System Update (VMSUPDATE) procedure. The steps for using this utility are described in the following section.

USING THE SYSTEM UPDATE PROGRAM

The System Update (VMSUPDATE) utility allows for orderly installation of the VDS software modules. VMSUPDATE automatically allocates space to load the program files and creates the database for monitoring satellites and the virtual disk volumes. Load the software from the distribution tape cartridges according to the following steps.

1. ( ) Log on to the system under a privileged account.

```
Place the VAXHSL03 volume in the drive
Are you ready to continue?:
```

8. ( ) Mount the third tape, labeled "VAXHSL03", in the console drive.

9. ( ) At the terminal, enter "Y" for yes.

   The system begins copying the data from the last tape. When the copying is complete, all the common and private virtual volumes have been created.

   The system displays a prompt, indicating the end of tape copying:

```
Are there any more kits to process?:
```

10. ( ) Enter "N" for no to conclude the VDS software installation.

11. ( ) Retrieve the third VDS cartridge from the console drive. (Replace any cartridge that was mounted when you began these installation procedures.)

VAX/VMS EMUNET FILES

After using the System Update utility to load the EMUNET VDS software, you will have several new system files in addition to the virtual volumes for the satellites. Table 5-1 shows the files loaded from the Emulogic EMUNET system VAX/VMS distribution media.

File	Description
SYS$HELP:ECPHELP.HLB	A help library for ECP.
SYS$MANAGER:EMUCONFIG.COM	A command input file for ECP. It is used by EMUNET.COM. The distribution copy configures one satellite.
SYS$MANAGER:EMUNETUP.COM	A command procedure to load the HSL driver, and start the EMUNETMGR.
SYS$MANAGER:EMUNETDWN.COM	A command procedure to shutdown the network.
SYS$MANAGER:LOADZV.COM	A command procedure to load the HSL

	driver.
SYS$SYSTEM:COMMON.DSK	A virtual volume containing common RT-11 utilities.
SYS$SYSTEM:ECP.EXE	The EMUNET Control Program.
SYS$SYSTEM:EMUNETMGR.EXE	The network manager.
SYS$SYSTEM:PRIVATE.DSK	A prototype private (write access) virtual volume, that contains a boot for the HSL and RT-11 system software. This volume should be copied for each node.
SYS$SYSTEM:VDX.EXE	The virtual volume file transfer utility.
SYS$SYSTEM:ZVDRIVER.EXE	The driver for the HSL.

FINAL ADJUSTMENTS

To run the EMUNET system most conveniently, we suggest that you modify your system and manager bootstrap command files -- SYS$SYSTEM:STARTUP.COM and SYS$MANAGER:SYSTARTUP.COM. Once you have done this, each time the system is cold-started it will be correctly configured for EMUNET system operation. The additions are described in the following sections.

SYSTEM START-UP FILE MODIFICATION

First, log on under a privileged account. Then use a text editor to modify the file SYS$SYSTEM:STARTUP.COM. Re-writing the command "AUTOCONFIGURE ALL" to read:

```
AUTOCONFIGURE ALL/EXCLUDE=(XA)
```

This ensures that the system does not attempt to configure the XA-type HSL boards. That function will be performed by the network driver loader, as described in the next section. Exit from the text editor.

MANAGER START-UP FILE MODIFICATION

Now use the text editor to modify the command file, SYS$MANAGER:SYSTARTUP.COM. In this file, you should append the following 2 commands after those already present:

```
$ @SYS$MANAGER:LOADZV
$ @SYS$MANAGER:EMUNETUP
```

The first of these commands loads the VDS driver program and configures the host system to access the HSL boards. The second of the commands initializes the EMUNET system control software. Exit from the text editor.

SUMMARY

Loading and initialization of EMUNET system software is facilitated by an Emulogic software loading routine. When loading is complete, all files, programs, and virtual volumes required for the system are present and operating. To enable the EMUNET system hardware and software automatically during a system re-boot (cold start), you can modify the bootstrap command files using a text editor.

CHAPTER 6

VIRTUAL DISK UTILITY PROGRAM

The Virtual Disk Utility (VDX) is a feature of both RSX-11M and VAX/VMS versions of the Emulogic EMUNET system. It provides access to virtual disks from the FILES-11 host files and also includes several virtual disk maintenance functions. VDX allows you to perform the following:

- o Transfer files from host into virtual disks,
- o Transfer files from virtual disks into host files,
- o List RT-11 directories of virtual disks,
- o Delete files from virtual disks,
- o Initialize virtual disks in RT-11 format,
- o Load RT-11 bootstraps (same as RT-11's COPY/BOOT),
- o Create new virtual disk files,
- o Extend exisiting virtual disk files.

You can use VDX interactively or indirectly through a command file. When used in command files, VDX commands can be nested up to 3 deep. Although some functions are similar to those of other file transfer utilities, VDX is not a replacement for FLX or PIP. VDX supports only RT-11-formatted virtual disks and RT-11 operations.

CALLING THE VIRTUAL DISK UTILITY

There are unique commands for calling VDX under the RSX-11M or VAX/VMS operating system, respectively. To invoke VDX under RSX-11M, enter the following command:

```
VDX
```

Under VAX/VMS, enter the command below:

```
MCR VDX
```

When entered under its respective system, the command causes the system to execute the VDX program and display the prompt "VDX>" on your terminal. This prompt indicates that the system is ready to process any interactive VDX command.

EXITING THE VIRTUAL DISK UTILITY

To terminate the interactive VDX utility, press the CTRL and Z keys simultaneously (written <CTRL>Z). This action returns you to the system monitor.

VDX COMMAND LINE FORMATS

Although the formats of individual VDX commands vary, the general format is as follows:

```
outfile/sw=infile/sw,....,infile/sw
```

outfile represents the output file specification. This specification is not used in all VDX operations. When used, it may contain a Files-11 file specification or it may be limited to a device and UFD or directory.

infile represents the input file specification. This specification may be any Files-11 or RT-11 file specification (file name and extension only).

/sw a VDX switch. Specifying a switch is not necessary for some commands. VDX switches are described later in this chapter under "VDX Switches".

FILE NAMES

VDX supports 9-character file names for all Files-11 files. RT-11 file names are restricted or truncated to 6 characters. Wildcard symbols are valid only for input file specifications and are restricted to use in the file name and extension only. Version numbers are valid only for Files-11 files and cannot be specified as wildcards. The standard rules for updating version numbers apply.

COMMON VDX ACTIVITIES

The Virtual Disk Utility is quite flexible, and -- in time -- you may discover many additional uses for it. In this section, we have provided the command structures for performing some basic functions and routines. The examples provided are intended to clarify the use of the command syntax for the specified function. These examples may not always apply to conditions that exist on your system.

CREATING VIRTUAL DISKS

To create a virtual disk file use the following VDX command line:

```
outfile/CR/AL:n[/CO]
```

outfile the Files-11 file specification of the virtual disk file to be created. This specification can include a device, UFD, filename, and extension. Wildcards are not allowed. The default extension is .DSK.

/AL:n specifies the number of blocks to allocate for the virtual disk space.

/CO (optional) indicates that the virtual disk is to be contiguous. Use of /Co is recommended for faster access times.

The virtual disk file is created and each block is zeroed to clear any residual information from that file space. Note that before files can be transfered into the vitual volume, it must zeroed.

Examples:

```
VDX>UNIT0.DSK/CR/AL:2000/CO
VDX>UNIT1.DSK/CR/AL:1000
```

DEFINING THE ACTIVE VIRTUAL DISK

Before VDX can be used to transfer files, list directories, delete files, initialize volumes, or load bootstraps, a virtual disk must be designated for use. To define and open a virtual disk, use the following command line format:

```
infile/VD
```

infile — the Files-11 file specification or device specification of the virtual disk to be opened. This specification can include a device, UFD, filename, and extension. Wildcards are not allowed. You must have read/write access to the file for it to be opened successfully. The default extension is .DSK.

The virtual disk opened by this command line will remain open until over-ridden by another /VD operation. If a device is specified, it must be a directory structured device and must not be mounted or allocated by another user.

NOTE

VDX does not assign devices. You must take measures to ensure that your disk is not removed by another system user.

Examples:

```
VDX>UNIT0.DSK/VD
VDX>UNIT1/VD
```

EXTENDING VIRTUAL DISKS

VDX can extend existing virtual disk files using the following command line format:

```
infile/VD/EX:n(/CO)
```

infile — the Files-11 file specification of the virtual disk file to be extended. This file cannot be opened by other users or VIP.

/CO — the optional switch to allocate additional space contiguously; the resulting extended file will always be noncontiguous.

The virtual disk file will be opened and extended by "n" blocks. The additional blocks allocated to the file will be zeroed to clear out any residual information stored on those blocks. VDX will then attempt to alter the RT-11 directory to reflect the additional space allocated to the file. If the virtual disk does not have a valid RT-11 directory, the

INVALID RT-11 DIRECTORY error will result but the file will be extended.

The virtual disk file remains open and is the active virtual disk file until another /VD operation is performed.

Example:

```
VDX>UNIT1.DSK/VD/EX:1000/CO
```

FILE TRANSFERS

VDX transfers files between virtual disk volumes and other files maintained by the host. These transfers are described in the following two sections.

Virtual Disk to Files-11 File Transfers

To transfer files from a virtual disk to FILES-11 use the following command line format:

```
outUFD(/RS)(/IM)(/FA)=infile(/RT),....,infile
```

outUFD — a device and UFD specification. Defaults under RSX-11M for outUFD are SY: and the current UFD; for VAX/VMS, the default is the current default directory. The output file names are the same as the input file names and cannot be specified.

/RS — identifies the output as RSX-11M or VAX/VMS, depending on the host.

/IM — (optional) identifies the transfer as image mode. This is the default transfer mode. This switch may appear on either the input or output side.

/FA — (optional) identifies the transfer as formatted ASCII mode. This switch may appear on either the input or output side.

infile — are the RT-11 input file specifications. These may contain wild cards. The default input file is "*.*".

/RT — (optional) identifies the input as RT-11.

NOTES

1. Either /RS (on output side) or /RT (on input side) must be specified; the other is automatically assumed. /RT need only be specified for one of the input files.

2. All files are transferred using the same transfer mode. /FA and /IM may not be used on a per file basis.

Examples:

```
VDX>/RS=*.MAC/FA
VDX>DL1:[200,200]=RT11FB.SYS,SWAP.SYS/RT
```

Files-11 to Virtual Disk File Transfers

To transfer files from FILES-11 to a virtual disk use the following command line format:

```
[/RT][/IM][/FA]=infile[/RS], . . .,infile
```

The files created have the same file names as the input files (the file name is truncated to 6 characters).

/IM (optional) identifies the transfer as image mode. This switch may appear on either the input or output side. Default mode is determined from each input file.

/FA (optional) identifies the transfer as formatted ASCII. This switch may appear on either the input or output side. Default mode is determined from each input file.

infile a Files-11 input file specification. An input file specification may contain a wildcard symbol.

/RS identifies the input as RSX-11M or VAX/VMS, depending upon the host CPU.

NOTES

1. Either /RT (on output side) or /RS (on input side) must be specified; the other is automatically assumed. /RS need only be specified for one of the input files.

2. Files are transferred using the appropriate transfer mode (image or formatted ASCII) on a per-file basis (determined from the Files-11 file structure) unless over-ridden by /FA or /IM. If overridden by /FA or /IM, all files will be transferred in that mode.

Examples:

```
VDX>/RT=*.MAC
VDX>=VDX.TSK,SAMPLE.TXT/RS
```

DIRECTORY LISTINGS

Directory listings of virtual disks can be generated by VDX using the following command line format:

```
outfile[=infile]/LI
```

outfile — the Files-11 output file specification. The primary output device is the default.

infile — (optional) RT-11 file name specification. Wildcards may be used. Only one file specification can be given. *.* is the default specification.

VDX will produce the date and virtual disk specification followed by a standard RT-11 directory showing the file name, file name, file size, and date of creation. A summary giving the number of files listed, their total size, and the remaining free space on the disk is also generated.

```
VDX>DL0:[4,1]SYSTEM/VD
VDX>*.SAV/LI
```

Figure 6-1 shows a sample directory listing of a virtual disk given in response to the above command lines:

```
9-DEC-82          DL0:[4,1]SYSTEM.DSK;1

DUP        .SAV     41      1-FEB-82
PIP        .SAV     23     29-FEB-80
DIR        .SAV     17     29-FEB-80
RESORC     .SAV     15     29-FEB-80
KED        .SAV     60     29-FEB-80
LINK       .SAV     41     29-FEB-80
MACRO      .SAV     51     29-FEB-80
SETDAT     .SAV      2      8-AUG-82
HELP       .SAV    107      4-APR-82
CREF       .SAV      6     29-FEB-80

10 Files, 362 Blocks
580 Free Blocks
```

Figure 6-1: Sample Directory Listing

FILE DELETIONS

Files can be deleted from the RT-11 directory using the following command line format:

```
infile/DE/RT
```

infile — any valid RT-11 file name. Wildcards can be used for either the name or extension or both.

The files will be deleted from the RT-11 directory. Use /ZE/RT to delete all files; using *.* is not recommended. In the example below, ECP will delete all files with the .MAC extension.

Example:

```
VDX>*.MAC/DE/RT
```

INITIALIZING RT-11 VOLUMES

The active virtual disk can be initialized using the following VDX command line:

```
/ZE:n/RT
```

/ZE:n — is the number of directory segments to be allocated. VDX uses a default value based on the virtual disk size if "n" is not specified or is specified as zero.

No file specifications are allowed.

VDX will initialize the virtual disk's directory and wipe out any bootstrap code in the boot blocks (and load the "No boot on volume" message). The RT-11 INIT/RECOVER operation will recover the virtual disk directory if used before the directory is altered.

Size (Blocks)	Segments	Useable _Blocks
494	2	484
988	4	974
4,800	8	4,778
10,240	16	10,202
20,480	31	20,412

Table 6-1: Default Directory Segments

INSTALLING BOOTSTRAPS

VDX can be used to load the bootstrap for RT-11 monitors and devices. This feature is similar to the RT-11 COPY/BOOT operation. Use the following command line format:

```
infile/BO:dev
```

infile — the name of the RT-11 monitor file to be bootstrapped (it must have a .SYS extension). The file specification cannot contain wildcards.

/BO:dev — indicates that the bootstrap for device handler dev is to be loaded. Dev is the 2-character device name (X is appended automatically by VDX for the XM monitors).

The bootstrap code from the monitor file is loaded into blocks 2 to 5 of the virtual disk. The secondary bootstrap code from the device handler is loaded into block 0.

The following errors can result during bootstrap loading:

o VDX--No such files(s) - infile
o VDX--Device handler not found - dev.SYS
o VDX--Device not bootable - dev.SYS

NOTE

The state of the boostrap is questionable if an error occurs during processing.

Examples:

```
VDX>RT11FB.SYS/BO:VS
VDX>RT11XM.SYS/BO:VS
```

VDX SWITCHES

VDX switches are used to specify both operations and the operation conditions. The 13 switches are listed and described briefly below:

Switch Description

/AL:n Indicates the number of blocks (n) to be allocated to the output file.

This switch is only valid with the /CR (create) operation.

/AL is not valid for file transfers since the length of the output files is determined from the input file length.

/BO:dev Indicates that the bootstrap information from the monitor file (the input file) and device handler (dev) is to be written to the virtual disk.

This switch must be accompanied by a 2-character device name. VDX automatically appends an X when an XM monitor is being bootstrapped.

A virtual disk must be open to use this switch (see /VD).

/CO Indicates that the output file is to be contiguous.

INITIALIZING RT-11 VOLUMES

The active virtual disk can be initialized using the following VDX command line:

```
/ZE:n/RT
```

/ZE:n is the number of directory segments to be allocated. VDX uses a default value based on the virtual disk size if "n" is not specified or is specified as zero.

No file specifications are allowed.

VDX will initialize the virtual disk's directory and wipe out any bootstrap code in the boot blocks (and load the "No boot on volume" message). The RT-11 INIT/RECOVER operation will recover the virtual disk directory if used before the directory is altered.

Size (Blocks)	Segments	Useable _Blocks
494	2	484
988	4	974
4,800	8	4,778
10,240	16	10,202
20,480	31	20,412

Table 6-1: Default Directory Segments

INSTALLING BOOTSTRAPS

VDX can be used to load the bootstrap for RT-11 monitors and devices. This feature is similar to the RT-11 COPY/BOOT operation. Use the following command line format:

```
infile/BO:dev
```

infile the name of the RT-11 monitor file to be bootstrapped (it must have a .SYS extension). The file specification cannot contain wildcards.

/BO:dev indicates that the bootstrap for device handler dev is to be loaded. Dev is the 2-character device name (X is appended automatically by VDX for the XM monitors).

The bootstrap code from the monitor file is loaded into blocks 2 to 5 of the virtual disk. The secondary bootstrap code from the device handler is loaded into block 0.

The following errors can result during bootstrap loading:

o VDX--No such files(s) - infile
o VDX--Device handler not found - dev.SYS
o VDX--Device not bootable - dev.SYS

NOTE

The state of the boostrap is questionable if an error occurs during processing.

Examples:

```
VDX>RT11FB.SYS/BO:VS
VDX>RT11XM.SYS/BO:VS
```

VDX SWITCHES

VDX switches are used to specify both operations and the operation conditions. The 13 switches are listed and described briefly below:

Switch Description

/AL:n Indicates the number of blocks (n) to be allocated to the output file.

This switch is only valid with the /CR (create) operation.

/AL is not valid for file transfers since the length of the output files is determined from the input file length.

/BO:dev Indicates that the bootstrap information from the monitor file (the input file) and device handler (dev) is to be written to the virtual disk.

This switch must be accompanied by a 2-character device name. VDX automatically appends an X when an XM monitor is being bootstrapped.

A virtual disk must be open to use this switch (see /VD).

/CO Indicates that the output file is to be contiguous.

When the /CO swith is used with /CR, the resulting file will be contiguous.

When used with file transfers, the initial allocation of the output file will be contiguous but the resulting file may not be if additional space must be allocated.

/CR Indicates that a virtual disk file is to be created.

A Files-11 input file specification is required.

Use "/AL:n" to specify the size of the file being created; a default of 988 blocks is allocated if you do not use the allocate switch. The file can be made contiguous using the /CO switch.

The default extension for virtual disk files is .DSK.

VDX zeroes every block of the virtual disk file.

/DE Indicates that files are to be deleted.

The /RT switch must be specified with /DE. Only RT-11 files can be deleted.

Only one input file specification is allowed. It may include wildcards.

/EX:n Indicates that the virtual disk file is to be extended by n blocks.

This switch can only be specified with the /VD switch. If /CO is specified, the extended file space will be contiguous, but the resulting virtual disk file will be noncontiguous.

/FA Indicates that the file is to be transferred using formatted ASCII mode.

Formatted ASCII is defined as ASCII data records terminated by carriage return/line feed (CR/LF). The file is translated from the source format into the destination forma

/FA should be used with all text and source files.

/IM Indicates that the file is to be transferred using image mode.

Files in image mode are transferred in the format they are in; they are not translated.

/IM should be used with all executable and data files.

/LI Causes a directory listing of all current virtual disk files.

VDX cannot be used to list Files-11 volume directories. Only RT-11 virtual disk directories are generated.

/RT is implied and is optional.

If you do not give an output file specification, the directory list is output to TI:

If you do not specify an input file name and file type, *.* is assumed.

Only one input file specification can be supplied; it may include wildcards.

/RS Indentifies the input or output specifications as Files-11 specifications.

/RS should only be specified for file transfers.

/RT Identifies the input or output specifications as RT-11 specifications.

/RT is primarily used for file transfers. It is also required with the /DE and /ZE switches.

/VD Indicates that the input file specification is to be opened as a virtual disk.

This operation is usually always required as it defines the virtual disk to be used for all RT-11 operations.

"/Ex:n" can be used to extend the virtual disk.

Virtual disks are opened for shared access unless /EX:n is specified. /VD/EX:0 can be used to prevent VDX from using shared access on the virtual disk file.

/ZE:n Indicates that the virtual disk is to be initialized.

/ZE:n can be used to specify the number of directory segments to be allocated. The maximum number of segments is 31 decimal.

If /ZE or /ZE:0 is used, VDX uses a default value which is related to the virtual disk size (similar to DUP's defaults).

/RT is required with the /ZE operation.

VDX ERROR MESSAGES

Errors encountered by VDX during processing are reported on the terminal issuing the VDX command. VDX error messages and suggested user actions are described below:

VDX -- Cannot extend device - (device)

Reason: An attempt was made to extend a device; devices have fixed sizes and cannot be extended.

Corrective Action: Use a larger device or use multiple volumes.

VDX -- Command file error

Reason: An unexpected error during command processing was encountered from either an indirect command file or TI:

Corrective Action: Correct the condition that caused the error.

VDX -- Command input error

Reason: An error was detected while attempting to read a command line.

Corrective Action: Correct the condition that caused the error.

VDX -- Command syntax error

Reason: A command line entered does not follow the VDX syntax rules.

Corrective Action: Re-enter the command line correctly.

VDX -- Device handler not found - (file)

Reason: The device handler specified in the BO command was not found on the virtual disk.

Corrective Action: Transfer the device handler to the virtual disk and re-enter the command line.

VDX -- Device not bootable - (file)

Reason: The device handler specified does not contain a secondary bootstrap routine. The device cannot be bootstrapped.

Corrective Action: Use a bootable device.

VDX -- Device not file structured - (device)

Reason: An attempt was made to open a nonfile structured device as a virtual disk.

Corrective Action: Use a file-structured device.

VDX -- Formatted ASCII record too long - (file)

Reason: A record was encountered that was more than 512 bytes in length during a /FA transfer. Either the file is corrupted, or the file is not formatted ASCII.

Corrective Action: If the file is corrupted, no recovery is possible. If the file type is incorrect, retry the operation specifying the correct transfer mode switch.

VDX - Invalid command line

Reason: The file specification does not conform to proper syntax.

Corrective Action: Re-enter the command line correctly.

VDX - Invalid input file - (file)

Reason: The file specification does not conform to proper syntax.

Corrective Action: Re-enter the command line with the proper syntax.

VDX - Invalid output file - (file)

Reason: The file specification does not conform to proper syntax or too many output files were specified.

Corrective Action: Re-enter the command line with the proper syntax.

VDX - Invalid RT-11 directory

Reason: The virtual disk's directory is not in RT-11 format.

Corrective Action: Make sure the virtual disk is correct or initialize the virtual disk directory (use /ZE/RT).

VDX - Invalid RT-11 filename - (file)

Reason: The file specification does not conform to proper syntax.

Corrective Action: Re-enter the command line with the proper syntax.

VDX -- Invalid virtual disk file - (file)

Reason: The virtual disk file specification is invalid. Wildcards cannot be used for virtual disk names.

Corrective Action: Re-enter the command line correctly.

VDX -- No virtual disk file open

Reason: No virtual disk file has been opened.

Corrective Action: Open a virtual disk (see /VD) and retry the operation.

VDX -- Output file specification not allowed

Reason: An output file specification was entered for a command that does not allow one.

Corrective Action: Re-enter the command line without an output file specification (only a device name and UFD may be specified for transfers to RSX-11M).

VDX -- RT-11 directory full - (file)

Reason: The virtual disk's directory is full. No more file entries may be made.

Corrective Action: Either use an alternate virtual disk or squeeze the disk on a satellite system (use SQUEEZE).

VDX -- Virtual disk file too small - (file)

Reason: The size of the virtual disk file being created is too small. The minimum virtual disk file being created is too small. The minimum virtual disk file size is 100 blocks.

Corrective Action: Use a size greater than 100 blocks.

VDX -- Virtual disk full - (file)

Reason: The virtual disk directory does not contain a contiguous area large enough for the file being transferred.

Corrective Action: Extend the virtual disk file using /VD/EX or squeeze the disk on a satellite system.

VDX -- Virtual disk read error (message) - (file)

Reason: A read error message occurred while reading from the virtual disk. The (message) field specifies the nature of this error.

Corrective Action: Correct the condition that caused the error and retry the operation.

VDX -- Virtural disk write error (message) - (file).

Reason: A write error occurred while writing to the virtual disk. The (message) field contains the nature of this error.

Corrective Action: Correct the condition that caused the error and retry the operation.

VDX -- Wildcard version number(s) not allowed - (file)

Reason: A wildcard was detected in the version number field of a file specification.

Corrective Action: Re-enter the command line with all version numbers explcitly specified.

OTHER ERRORS

During execution of VDX operations, you may also receive standard RSX messages. You can find explanations for these messages in DEC documentation for RSX-11 systems.

CHAPTER 7

THE EMUNET CONTROL PROGRAM

The EMUNET Control Program (ECP) is VAX/VMS utility that allows the user to configure, control, and monitor the EMUNET HSL system. This chapter explains ECP and provides a command reference summary.

RUNNING ECP

To run ECP, issue the following DCL command:

```
   $ RUN SYS$SYSTEM:ECP
```

By using the DCL foreign command feature, single-line ECP commands may be issued as follows:

```
   $ ECP SHOW NODES
```

This assumes that the foreign command ECP has been defined as follows:

```
   $ ECP :== $ECP
```

ENTERING ECP COMMANDS

ECP commands are entered as keywords and parameters separated by spaces and or tabs. To continue a long command on the next line, use the standard continuation convention of a hyphen as the last character in a line. Continuation lines will be prompted for with an underscore. For example:

```
   ECP>SET NODE ZAPHOD LINE 1 STATION 2 -
   ECP_VOLUME 0 RT11PR.SYS -
   ECP_VOLUME 1 GROUP.DSK/READ
```

A comment is preceded by an exclamation point (!). ECP will ignore hyphens within and at the end of a comment, so that a continuation hyphen should precede the comment. For example:

```
ECP>SET NODE ZAPHOD - !  Located in conference room
ECP_LINE 1 STATION 2
```

ECP allows you to abbreviate command verbs and keywords to the fewest number of unique letters. For example, the following two commands perform the same function:

```
ECP>SET LINE 1 DEVICE ZVA0 MULTIDROP

ECP>SE LI 1 DEV ZVA0 MU
```

HELP Facility

ECP has an extensive online HELP facility, that provides information on each ECP command its parameters and general examples. The help information is tree-structured, which makes it easy to retrieve information quickly.

The HELP command enters the help facility. You may optionaly add topics and sub-topics to the HELP command to specify where in the help tree to start. Following the help information display, a list of keywords will appear in the "Additional information available" section. These keywords serve as topics for the next level of help available. An asterisk wild card symbol (*) will display help for all topics.

EXITING

To exit ECP, type the command "EXIT" or press <CTRL>Z after the prompt "ECP>".

COMMANDS

This section provides an alphabetical list of ECP commands. Each command begins with a general description, followed by the command format and a description of the user-supplied elements.

PARAMETER SYNTAX RULES

Many ECP commands require user-supplied information. For the most part, the syntax follows a standard set of rules. Any exceptions will be noted in the description of the parameter. All numeric values are in decimal and must be within the range of 0 to 65535. The syntax of parameters follows:

device-id a string of characters consisting of a device name, controller and unit number. The controller defaults to A and the unit defaults to 0.

file-id a file name of up to nine characters, optionally followed by a period and a file type of up to three characters; or a logical name of up to sixty-four characters. The file name may optionally be preceded by a device and directory specification.

node-name a string of up to six characters.

volume-id a numeric value in between zero and the site specific MAXVOL. Typically MAXVOL is eight.

CLEAR NODE

This command deletes node-defining parameters from the EMUNET database.

Format: CLEAR NODE node-name ALL

Effect: With the ALL option, the command deletes every volume assigned to the node (currently the only format option).

HELP

Use the HELP command to obtain general information about ECP commands and parameters.

Format: HELP [topic...] A "topic" is any keyword listed in the HELP display under "Additional information available".

Example:

```
ECP>HELP
```

This command format presents the primary HELP display. The display describes the purpose and features of the ECP utility. In addition, it lists the following legal topic names which may be used with the HELP command for information on a specific ECP function.

```
o CLEAR      o SHOW-12
o EXIT       o SHUTDOWN
o HELP       o WRITE
o SET
```

SET LINE

The SET LINE command creates or modifies line parameters in the EMUNET data base.

```
                                 [DEVICE device-id]
Format:      SET LINE line-id   [MULTIDROP]
                                 [OFFLINE]
                                 [ONLINE]
```

DEVICE device-id specifies the communications device associated with the line.

MULTIDROP specifies that the line may contain a number of nodes.

OFFLINE disables processing of all packets originating on the line.

ONLINE enables processing of all packets originating on the line.

Example:

```
NCP>SET LINE 1 DEVICE ZVA0:  MULTIDROP
```

This commands informs EMUNET to start listening for packets on device ZVA0, the first HSL unit.

SET NODE

The SET NODE command creates or modifies node parameters in the EMUNET data base.

```
Format:      SET NODE node-name  LINE line-id [STATION station-id]
                                    VOLUME volume-id file-id [/READONLY]
```

LINE line-id specifies the line that the node is attached to. This keyword must follow the node-name if the node has not been defined.

STATION station-id specifies the node number on a multi-drop line.

VOLUME . . . maps an VAX/VMS RT-11 virtual volume file to a specific volume number for the node. The /READONLY switch can be specified to prevent writing to the volume or allow sharing of volumes among satellites.

Example:

```
ECP>SET NODE ZAPHOD LINE 1 STATION 5
```

This command names node 5 on line 1 as ZAPHOD.

SHOW LINE

The SHOW LINE command displays line information maintained by EMUNET.

```
Format:     SHOW    LINE line-id
                     LINES
```

LINE line-id Indicates that information for a particular line be displayed.

LINES Indicates that information for all lines be displayed.

Example;

```
ECP>SHOW LINE 1
```

This command will present a display showing the device name, status, errors, and data packets communicated for the specified line (line 1).

SHOW NETWORK

This command displays statistical information and counters maintained by EMUNET.

```
Format:     SHOW NETWORK
```

This command presents a display showing the packets received and sent by the host and the number of times host or satellite buffers were filled when data was being sent to those buffers.

SHOW NODE

The SHOW NODE command displays node information maintained by EMUNET.

```
Format:      SHOW NODE node-name
                  NODES
```

NODE node-name indicates that information for a particular node be displayed.

NODES indicates that information for all nodes be displayed.

Example:

```
ECP>SHOW NODE RPS
```

This command presents a display showing the node name, line number and station number assigned to the node, RPS. On subsequent lines, the display will indicate numbers and names of the virtual volumes owned by the node.

SHUTDOWN

The SHUTDOWN command terminates the EMUNET software. The EMUNETMGR process ceases. No further commands should be entered.

```
Format:      SHUTDOWN
```

WRITE

The WRITE command creates a command file that contains the current state of the EMUNET system. This command file can in turn be used to initialize the EMUNET system via the indirect command file facility.

```
Format:      WRITE file-id
```

file-id the name of the file to be created. The default file extension is ".COM".

Examples

```
ECP>WRITE SYS$MANAGER:EMUCONFIG
```

This command creates the file SYS$MANAGER:EMUCONFIG.COM.

Indirect Command Processing

You can execute one or more ECP commands by placing the command sequence in a command file and invoking the file as a system command. Each time the command file is invoked, the system executes the ECP command sequence automatically. Nested command procedures are allowed.

Format: @file-id

file-id is the command procedure file name. The default file extension is ".COM".

Example:

```
ECP>@SYS$MANAGER:EMUCONFIG
```

ECP will read command from the specified file. When all commands have been read, ECP will then read commands from the terminal.

SAMPLE NETWORK CONFIGURATION

This section illustrates how to use ECP commands to configure a sample network. This network has two EMUNET lines. The first line contains the nodes ZAPHOD, WMICE and FORD. The second line contains the nodes Z80 and M68K.

Each node will have virtual volume 7 mapped to the file SYS$SYSTEM:COMMON.DSK. This is a virtual volume of all DEC-supplied RT-11 software. Each node will then have virtual volume 0 mapped to a private volume. Volume 0 should contain the device drivers, minimal RT11SJ software and the STARTS.COM file.

```
 $!++
 $! Sample procedure to configure a network.
 $!
 $! The network has the following topology:
 $!
 $!      VAX ----+--- line 1 ----+---- FORD
 $!       |      |               |
 $!       |      ZAPHOD          WMICE
 $!       |
 $!       +---- line 2 ----+---- Z80
 $!                        |
 $!                        M68K
 $!--
 $ RUN SYS$SYSTEM:ECP
 SET LINE 1 DEVICE ZVA0: MULTIDROP       ! Define the two lines
 SET LINE 2 DEVICE ZVA1: MULTIDROP
 !+
 ! Configure the nodes on line 1.
 !-
 SET NODE ZAPHOD LINE 1 STATION 1 -
     VOL 7 SYS$SYSTEM:COMMON.DSK/READ -
     VOL 0 ZAPHOD.DSK
 SET NODE WMICE LINE 1 STATION 2 -
     VOL 7 SYS$SYSTEM:COMMON.DSK/READ -
     VOL 0 WMICE.DSK
 SET NODE FORD LINE 1 STATION 3 -
     VOL 7 SYS$SYSTEM:COMMON.DSK/READ -
     VOL 0 FORD.DSK
 !+
 ! Configure the nodes on line 2.
 !-
 SET NODE M68K LINE 2 STATION 1 -
     VOL 7 SYS$SYSTEM:COMMON.DSK/READ -
     VOL 0 M68K.DSK
 SET NODE Z80 LINE 2 STATION 2 -
     VOL 7 SYS$SYSTEM:COMMON.DSK/READ -
     VOL 0 Z80.DSK
 EXIT
.sk 2
```

APPENDIX A

HSL JUMPER OPTIONS

The HSL communications device consists of two boards: one labeled DATACOM and one labeled DMA Bus. Each computer (the host and all satellites) must have one HSL device installed. The HSL device must be properly configured before installation.

The HSL DMA board contains jumper options for setting the device CSR address and vector. Each satellite has a different CSR address to allow the bootstrap program to identify which node it is booting. The same vector is used for all HSL boards, and was chosen to avoid conflicts with the standard DEC assign- ments.

Table A-1 lists the vector and CSR addresses for the host and each satellite for a network of 16 satellites. Note that the RSX-11M host computer and Satellite One have the same base address.

Tables A-2 and A-3 list which straps must be inserted (I) or remove (R) on the DMA to set each of the addresses in Table A-1.

		Base Vector
Computer	Address	Address
RSX-11M Host	175200	270
VAX/VMS Host	760320	270
Node 1	175200	270
Node 2	175210	270
Node 3	175220	270
Node 4	175230	270
Node 5	175240	270
Node 6	175250	270
Node 7	175260	270
Node 8	175270	270
Node 9	175300	270
Node 10	175310	270

Node 11	175320	270
Node 12	175330	270
Node 13	175340	270
Node 14	175350	270
Node 15	175360	270
Node 16	175370	270

Table A-1: HSL base and vector addresses

	V8	V7	V6	V5	V4	V3
270 (OCTAL)	R	I	R	I	I	I

Table A-2: HSL vector address jumper configuration

	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
175200	I	I	R	I	R	I	R	R	✓ R	R
175210	I	I	R	I	R	I	R	R	R	I
175220	I	I	R	I	R	I	R	R	I	R
175230	I	I	R	I	R	I	R	R	I	I
175240	I	I	R	I	R	I	R	I	R	R
175250	I	I	R	I	R	I	R	I	R	I
175260	I	I	R	I	R	I	R	I	I	R
175270	I	I	R	I	R	I	R	I	I	I
175300	I	I	R	I	R	I	I	R	R	R
175310	I	I	R	I	R	I	I	R	R	I
175320	I	I	R	I	R	I	I	R	I	R
175330	I	I	R	I	R	I	I	R	I	I
175340	I	I	R	I	R	I	I	I	R	R
175350	I	I	R	I	R	I	I	I	R	I
175360	I	I	R	I	R	I	I	I	I	R
175370	I	I	R	I	R	I	I	I	I	I

Table A-3: HSL base address jumper configuration

APPENDIX B

SYSTEM RESTRICTIONS

You should be aware of the following restrictions when using the EMULOGIC EMUNET system.

1. The RSX-11M ZV driver cannot be unloaded using the UNLOAD command. The ZV driver has device and timer activity even when no I/O opeations are pending. You must reboot the system to reload or remove the ZV driver. The UNLOAD command is not supported and will crash your system.

2. On systems with 22-bit QBUS addressing, the VIP tasks and the RSX-11M ZV driver must be run in the low 256k bytes of memory. To do this, create a partition for the ZV driver and the VIP tasks and install them in this partition.

3. The RT-11 VS driver does not support bad block scans. Bad block scan should be done by the RSX-11M host system on which the physical disks reside. Using the INIT/BADBLOCKS operation will not produce the desired results and may cause harm.

4. The RT-11 FORMAT program cannot opreate on a virtual disk. Although the virtual disk functions as an RT-11 disk, the files or devices which comprise it are physically part of an RSX-1M system and should be formatted by the RSX-11M host computer.

5. The RT-11 satellite VS driver and the DEC-supplied DM disk driver (RP06/RP07) cannot be used on the same system unless you patch the VS driver. If the VS driver and the DM driver must be used on the same system, you must change the device handler code of the VS driver.

6. The RT-11 VS driver and satellite bootstrap program make use of the

MTPS instruction, and, therefore, will not operate on the following processors without a patch:

- PDP-11/04
- PDP-11/05 and PDP-11/10
- PDP-11/15 and PDP-11/20

7. The RT-11 boot virtual volume (VS0:) must have read/write access, to allow the RT11SJ to write to the swap file.

APPENDIX C

SATELLITE BOOTSTRAP OPERATION

The bootstrap program is a 256-word program which will bootstrap either:

- o Virtual Disk Unit 0; or
- o RX01/RX02 Unit 0 or Unit 1.

The bootstrap sequence is as follows:

1. Size read/write memory, to a maximum of 30K words.

2. Perform a memory addressing test, writing each location with its address and verifying the contents.

3. Exercise memory data storage capabilities, moving a 1's pattern through a background of 0's and a 0's pattern through a background of 1's, to test all read/write memory.

4. Check for the presence of a HSL controller within the address range for the virtual disk. If not found, proceed to Step 6.

5. Attempt to bootstrap virtual disk unit 0. A maximum of 30 attempts are made to read block 0 from the host. If the re-try count is exhausted or an invalid bootstrap routine is read, proceed to Step 6. Otherwise the bootstrap routine is executed.

6. Check for the presence of RXV11 or RXV21 controller in the system. If none exists, proceed to Step 4.

7. Wait for a minimum of 2 seconds to allow the drive to spin up, then attempt to bootstrap unit 0 at the density of the media present in the drive at the time. If the drive is not ready or does not con-

tain a bootable medium, go to the other unit and try it. If neither drive contains a bootable disk, proceed to Step 4.

NOTE

The bootstrap routine will continue to loop indefinitely until a valid secondary bootstrap is found. If the bootstrap routine halts, a memory test has failed (see "Bootstrap Halts" below). A valid secondary bootstrap routine is defined by DEC as containing an NOP instruction in word 0 of the bootstrap routine.

BOOTSTRAP HALTS

Table C-1, below, lists the bootstrap halts and their addresses and meanings.

Table C-1: BOOTSTRAP HALTS

Address	Meanings/Diagnostic Information
173056	Memory Address Error R2 = Expected data and address of bad data
173102	Bad Memory Error R2 = Address of bad data R3 = Expected data

For more information on DEC bootstrap routines, see the DEC documentation on the BDV-11 and MXV-11 boards. Also, see the DEC RT-11 Software Support Manual (Chapter 7).

APPENDIX D

RT-11 HANDLER OPTIONS

The following "SET" commands can be issued for the RT-11 HSL device handler (VS.SYS). The system must be re-booted after these commands are issued for the change to take effect.

1. SET VS: VECTOR=n

Where n specifies the octal vector address of the HSL board in the RT-11 satellite. The default value is 270 (octal). Remember that if the vector address is changed, it must also be changed on the HSL DMA Bus board (see Section E.1).

2. SET VS: LUNS=n

Where n specifies the octal number of logical unit numbers. This number can not exceed the eight.

APPENDIX E

RSX-11M HOST SOFTWARE OPTIONS

There are several optional functions which may be set on the host software to customize its operation. These options will be described here for the host HSL driver (ZV), the host task (VIP), and the virtual disk file transfer utility (VDX).

If you are unable to use the automatic VDS installation procedure or if you wish to exercise some software option not available under VDSKGEN you will have to rebuild parts of the host VDS software. Command files are provided on the distribution media which you may edit to take advantage of options. It will in general be necessary to rebuild the parts of VDS which you want to modify. The following information will aid in the implementation of such modifications.

BUILDING ZVDRV FOR 22-BIT UNIBUS MAPPING

If the host computer has 22-bit UNIBUS mapping, then 22-bit UNIBUS mapping must be enabled when the ZV driver is task built. To do this, task build the ZV Driver with the command file ZVUNIMAP.CMD.

BUILDING ZVDRV FOR USE WITH Q-22

If the host computer has a Q-22 (22 bit Q-BUS) architecture build ZVDRV with the command file ZVQ22BLD.com

DISABLING VIP CHECKPOINTING

By default checkpointing is enabled for VIP. This allows the Executive to temporarily remove VIP from memory to make room for higher priority tasks. When memory is again available, VIP will be returned to memory. VIP can be task built with checkpointing disabled producing slightly improved virtual disk performance, at the expense of system memory resources.

To disable checkpointing edit the command file used to task build VIP (VIPBLD.CMD, for non-FCSRES version, VIPRESBLD-CMD for FCSRES version).

The following example shows how the command file should appear to disable checkpointing.

```
;CKPFLG is the checkpointing flag
;zero = no checkpointing
;nonzero = checkpointing enabled
;
 GBLEDEF=CKPFLG:0
```

LINKING VIP TO FSCSRES

VIP requres about 8k workds of memory per installed copy. If FCSRES (The RSX-11M file control system memory resident library) is available on the host system, linking it to VIP will save you about 4k words of memory per installed copy of VIP. Build VIP with the command file VIPRESBLD.CMD to utilize FCSRES.

VIP EXTENSION DURING OPERATION

You can specify the maximum number of 256-word blocks that VIP can extend itself when it is doing I/O. A larger number will provide better virtual disk performance at the expense of using more memory and increasing succeptibility to being checkpointed. This number can range from 0 to 74 (octal). To change the extension limit edit the command file which task builds VIP (VIPBLD.CMD, for non-FCSRES version, VIPRESBLD.CMD for FCSRES version). The command file contains the following instructions

```
;EXTMAX is the maximum task extension in 256-words blocks
;(octal!)
;Do not exceed 74 octal.
;
  GBLDEF=EXTMAX:6
```

Change the number following EXTMAX: to specify the number of blocks VIP can extend itself.

Larger extensions increase efficiency by limiting the number of disk accesses required to service a given request. Practical values are determined by disk speed and memory resources. Note that the more VIP extends itself the more succeptible it becomes to being check pointed, thus unrestricted extension may actually decrease throughput during periods of heavy memory contension.

LINKING VDX TO FCSRES

If FCSRES is available on the host system, VDX may be linked to it to reduce its size. To link VDX to FCSRES task build it with the command file VDXRESBLD.CMD.

RSX-11M HOST SOFTWARE INSTALLATION

To install the RSX-11M software components you must have loadable driver support. If you will be using RTEM-11 (the RT-11 emulator) you will also have to select the RTEM-11 support option. This option is not available in RSX-11M prior to version 4.0, and question 32A which allows you to select support is not asked unless Autopatch B has been applied. If you have chosen the standard function system SYSGEN option RTEM-11 support will have been included. If your system doesn't have required support then you will have to perform a SYSGEN in order to use the VDS software.

You must also ensure that the following two RSX-11M files reside in the accounts given below before attempting installation:

File	Account
EXEMC.MLB	LB:[1,1]
RSXMC.MAC	LB:[11,10]

File EXEMC.MLB is distributed with RSX-11M and file RSXMC.MAC is created by a SYSGEN.

Once the conditions above are met, follow these steps to install the RSX-11M host software:

1. Log into a privileged account on the RSX-11M host system;

2. Insert the HSL Virtual Disk System distribution disk into a suitable drive on the RSX-11M host computer.

3. Copy the following files from the distribution disk to the privileged account:

```
ZVDRVBLD.CMD    ZVDRV.OBJ
ZVUNIMAP.CMD
ZVQ22BLD.CMD    ZVTAB.OBJ
VIPBLD.CMD              VIP.OBJ
VIPRESBLD.CMD   VDX.OBJ
VDXBLD.CMD
VDXRESBLD.CMD
```

4. Task build the host HSL interface driver (ZV) by running the one of the following indirect command files:

```
TKB @ZVDRVBLD.CMD       ;for standard systems
TKB @ZVQ22BLD.CMD       ;for 22-bit UNIBUS mapping.
TKB @ZVQ22BLD.CMD       ;for systems with Q-22 architecture
```

5. Task-build the host task (VIP) by running one the following indirect command files:

```
TKB @VIPBLD.CMD ;for non-FCSRES version
TKB @VIPRESBLD.CMD      ;for FCSRES version
```

6. Task-build the host utility (VDX) by running one the following indirect command files:

```
TKB @VDXBLD.CMD ;for non-FCSRES version
TKB @VDXRESBLD.CMD      ;for FCSRES version
```

7. Copy all .TKB and .STB files to SY:[1,54]

8. Load the ZV driver. For example:

```
LOAD ZV:/PAR=GEN/HIGH
```

This load ZV: in the highest possible location in Gen. ZV: may be loaded in another portion if required.

9. Install a copy of VIP for each satellite connected to the host. Each copy of VIP must be installed under a different name; it is suggested that these names correspond to the satellite node addresses. The following sample shows installation of VIP tasks for a system with three satellites numbered 1 to 3:

```
INS VIP/TASK=...N01
INS VIP/TASK=...N02
INS VIP/TASK=...N03
```

Larger extensions increase efficiency by limiting the number of disk accesses required to service a given request. Practical values are determined by disk speed and memory resources. Note that the more VIP extends itself the more succeptible it becomes to being check pointed, thus unrestricted extension may actually decrease throughput during periods of heavy memory contension.

LINKING VDX TO FCSRES

If FCSRES is available on the host system, VDX may be linked to it to reduce its size. To link VDX to FCSRES task build it with the command file VDXRESBLD.CMD.

RSX-11M HOST SOFTWARE INSTALLATION

To install the RSX-11M software components you must have loadable driver support. If you will be using RTEM-11 (the RT-11 emulator) you will also have to select the RTEM-11 support option. This option is not available in RSX-11M prior to version 4.0, and question 32A which allows you to select support is not asked unless Autopatch B has been applied. If you have chosen the standard function system SYSGEN option RTEM-11 support will have been included. If your system doesn't have required support then you will have to perform a SYSGEN in order to use the VDS software.

You must also ensure that the following two RSX-11M files reside in the accounts given below before attempting installation:

File	Account
EXEMC.MLB	LB:[1,1]
RSXMC.MAC	LB:[11,10]

File EXEMC.MLB is distributed with RSX-11M and file RSXMC.MAC is created by a SYSGEN.

Once the conditions above are met, follow these steps to install the RSX-11M host software:

1. Log into a privileged account on the RSX-11M host system;

2. Insert the HSL Virtual Disk System distribution disk into a suitable drive on the RSX-11M host computer.

3. Copy the following files from the distribution disk to the privileged account:

```
ZVDRVBLD.CMD    ZVDRV.OBJ
ZVUNIMAP.CMD
ZVQ22BLD.CMD    ZVTAB.OBJ
VIPBLD.CMD              VIP.OBJ
VIPRESBLD.CMD   VDX.OBJ
VDXBLD.CMD
VDXRESBLD.CMD
```

4. Task build the host HSL interface driver (ZV) by running the one of the following indirect command files:

```
TKB @ZVDRVBLD.CMD         ;for standard systems
TKB @ZVQ22BLD.CMD         ;for 22-bit UNIBUS mapping.
TKB @ZVQ22BLD.CMD         ;for systems with Q-22 architecture
```

5. Task-build the host task (VIP) by running one the following indirect command files:

```
TKB @VIPBLD.CMD ;for non-FCSRES version
TKB @VIPRESBLD.CMD        ;for FCSRES version
```

6. Task-build the host utility (VDX) by running one the following indirect command files:

```
TKB @VDXBLD.CMD ;for non-FCSRES version
TKB @VDXRESBLD.CMD        ;for FCSRES version
```

7. Copy all .TKB and .STB files to SY:[1,54]

8. Load the ZV driver. For example:

```
LOAD ZV:/PAR=GEN/HIGH
```

This load ZV: in the highest possible location in Gen. ZV: may be loaded in another portion if required.

9. Install a copy of VIP for each satellite connected to the host. Each copy of VIP must be installed under a different name; it is suggested that these names correspond to the satellite node addresses. The following sample shows installation of VIP tasks for a system with three satellites numbered 1 to 3:

```
INS VIP/TASK=...N01
INS VIP/TASK=...N02
INS VIP/TASK=...N03
```

10. Run the installed VIP tasks using the following command line format:

```
NXX [dd:]/ND:n=spec1,spec2, . . .,spec8
```

[dd:] -(optional) the name of the device driver that the task will use for I/O. If not specified, the default is device ZV:.

/ND:n -the satellite node number that will be serviced. Legal values for n are 01 through 16 (decimal).

specn -the virtual disk specification for the satellite's virtual disk unit n.

Virtual disk volumes are assigned to virtual disk units based on position in the VIP command line. The first virtual volume specified is associated with VS0: the second with VS1: etc. If it is desired to assign to a unit which is not in sequence a null specification (ie. commas without virtual volume specifications between them) are used for the unassigned units which intervien.

A virtual disk specification is either an RSX-11M file specification, or an RSX-11M device specification. For example:

[4,1]UNIT0.DSK	For a virtual disk file.
DL2:	For a virtual disk device

The default extension for a file specification is .DSK.

NOTES

1. A satellite can have up to eight virtual disks. The file specifications are optional; for example, spec1 can be a null specification, meaning that unit has no disk.

2. To software write-lock the virtual disk unit specify /RO in the virtual device specification.

The following are examples of running the VIP task to assign virtual disks to a satellite:

```
>N01: /ND:1=DL0:[1,300]UNIT0.DSK,UNIT1.DSK/RO
```

To satellite node 1 this assigns VS0: the virtual disk volume in the RSX-11M file DL0:[1,300]UNIT0.DSK and VS1: the virtual disk volume

DL0:[1,300]UNIT1.DSK. DL0:[1,300] is taken as the default disk unit and UFD for the second virtual disk specification. Since the /RO switch is used in conjunction with UNIT1.DSK it will be attached to VS1: as a read only volume

```
   >NO2 /ND:2=UNIT0.DSK,DY0:
```

To the satellite at node 2, this assigns virtual disk unit VS0: the virtual disk volume UNIT0.DSK. UNIT0.DSK is taken from the users default disk and UFD. DY0: is attached to VS1: As a virtual disk device, and may be accessed as a device directly from the satellite, by referring to it as VS1:

```
   >FUD /ND:3=FRANK.DSK,,,,EXPCOM.DSK/RO,,COMMON DSK/RO
```

To the satellite a node 3 this assigns unit VS0: the virtual disk FRANK.DSK, VS4: the virtual disk EXPCOM.DSK and VS6: the virtual disk COMMON.DSK. Units VS1:, VS2:, VS3:, VS5: and VS7: have null assignments and can not be accessed from the satellite. EXPCOM.DSK and COMMON.DSK are attached for read only use.