

**Connection Machine System Software
VMEIO (Sun)
System Software
Installation Guide**

October 13, 1992

Version 6.1.2

This document assumes a working knowledge of UNIX system administration. Before beginning the installation, we strongly recommend that you acquaint yourself with the system administration documents appropriate to the operating system used on your VMEIO machine, especially:

SunOS Installation
SunOS System Administrators Manual
VMEIO Installation and Service Guide
CMIOP Installation and Service Guide

Also obtain and read the CM System Administrator's Guide, the CM User's Guide, and the release notes appropriate to this release of the Connection Machine System Software.

Table of Contents

1) VMEIO host requirements	3
2) Planning the installation	3
3) Reading the tape	3
4) Installing the software	4
5) What gets installed where	5
6) Testing the system	13
Appendix A) Installing the device drivers by hand	14

1. VMEIO Host Requirements

Release 6.1.2 of the VMEIO (Sun) CM System Software is based on SunOS 4.1.1. Prior to installing this release of VMEIO software, you should be sure that the machine you plan to install it on is running SunOS 4.1.1. In addition, the VMEIO driver will run on uniprocessor and multiprocessor machines running SunOS 4.1.2, although installation under these conditions is not automated.

Additionally, the full software release requires approximately 37 Mbytes of disk space. If your VMEIO machine is also a CM Front End running CMSS 6.1, then you will already have the majority of the files installed, and the space required for the VMEIO portion of the system software will be substantially less.

2. Planning the Installation

To install/upgrade the VMEIO system software, it will be necessary to perform the following steps as superuser:

- (1) Read the VMEIO files from tape and place them in a release directory.
- (2) Run the *cmiop-install* script to install the software. This script automates most of the installation scripts; for more custom installations this script may not be appropriate.
- (3) Reboot the system with a kernel that has the VMEIO drivers installed.
- (4) Clean up the release area
- (5) For a new installation, you should perform a checkout of the VMEIO hardware and edit the CM system configuration file to incorporate the VMEIO as a IO device attached to a specific bus.
- (6) Checkout data transfers between IO devices and the VMEIO

3. Reading the tape

To read the files from tape, create a directory that the files can be placed when read from tape. This directory can be temporary if you plan on moving the files to their final location, or it can be a permanent directory that you create symbolic links to.

```
# mkdir release-directory
# cd release-directory
```

Then read the tape using tar.

```
# tar xvf tape-device
```

If the host computer does not have a local tape drive, use a remote tape drive.

```
# rsh tapehost dd if=tape-device ibs=block-size | tar xBvf -
```

Where *tapehost* is the name of the computer which has the tape drive; *tape-device* is the name of the tape drive (it does not have to be a non-rewinding device); and *block-size* is 20b for a 9-track tape and 200b for a cart tape.

4. Installing the software

The *cmiop-install* script provides an easy, automated way to install the VMEIO host software and drivers. While it does not attempt to cover all possible installation possibilities, it simplifies installing the VMEIO software on a Thinking Machines CMIOP system, and allows reasonably flexible installations on other Sun systems (for instance, CM2/CM200 front ends).

To start the installation, run the *cmiop-install* script from the directory that the installation tape was untarred into:

```
# ./cmiop-install
```

The script will ask you if you are installing the software on a CMIOP system. If you are, it assumes default locations for all the software components that it will install (usually subdirectories of */usr/local*, for instance */usr/local/bin*). It will also configure a standard CMIOP kernel for you.

If you aren't installing on a CMIOP system, you will be prompted for the location to install each component of the software. Depending on the machine you are installing on, some of these components may be skipped -- for instance, if you are installing on a CM2/CM200 front-end, several of the components of the VMEIO release duplicate software that is shipped with the base system. These components can safely be skipped, although it won't hurt to install them "on top of" the base system.

Also, if you have installed any non-standard device drivers on your system, it is recommended you install the VMEIO device drivers by hand, as the installation script will supplant any modifications to kernel files like *conf.c* with its own (after backing up *conf.c*), thus effectively removing any extra device drivers.

You can type "help" at any prompt during the installation to get a brief description of what you're being asked to do.

Here's a sample run of the installation script, assuming you are installing the system on a CMIOP:

```
gpioc4!sean# ./cmiop-install
VMEIO 6.1.2 installation procedure
```

This procedure installs all software to use the VMEIO and StorageTek hardware on a Thinking Machines CMIOP system. Software installed includes:

- CM filesystem user commands
- CMIOP tape commands
- CM filesystem libraries and system commands
- VMEIO, Ciprico, and StorageTek device drivers

```
Is this install being done on a TMC CMIOP system [y/n/help]? [y] : y
```

```
Installing user commands and tape software in /usr/local/bin
```

```
Installing system commands and verifiers in /usr/local/etc
```

```
Installing library files in /usr/local/lib
```

```
Installing header files in /usr/include
```

```
Updating /etc/rc.local
```

```
/etc/rc.local saved as /etc/rc.local%TMC%Aug_13_1992
Connection machine daemons already in /etc/rc.local
vioconfig already in /etc/rc.local
Updating crontab
/usr/spool/cron/crontabs/root saved as /usr/spool/cron/crontabs/root%TMC%Aug_13_1992
Updating services
  Adding cm-logger-conn to /etc/services ...
You are running the Yellow Pages service
  The cm-logger service is correctly installed in the Yellow Pages
  The cm-logger-conn service is correctly installed in the Yellow Pages
```

```
Where do you build your kernels? [/sys] :
/sys/sundev/sc_conf.c saved as /sys/sundev/sc_conf.c%TMC%Aug_13_1992
/sys/sun/conf.c saved as /sys/sun/conf.c%TMC%Aug_13_1992
/sys/sun4/conf/files saved as /sys/sun4/conf/files%TMC%Aug_13_1992
```

```
Configuring kernel...
Doing a "make depend"
System configuration complete
Building kernel...
loading vmunix
rearranging symbols
text  data  bss  dec  hex
942080    150864 62912 1155856 11a310
New kernel built successfully
```

```
Installing new kernel as /vmunix
(old kernel saved as /vmunix.old)
Creating special device files
```

CMiop software installation is complete. You should edit /etc/rc.local to set up the proper CMIO bus id and station for the vioconfig command. Also set up the cm-logger parameters appropriate to your site.

Then reboot the system.

```
gpioc4!sean#
```

By indicating that the machine that the software is being installed on is a CMiop, the default configuration (installation in a subtree of /usr/local) was chosen. A standard kernel, called "CMiop" was built after the device drivers were installed, and the system now has to be rebooted to use the hardware.

Here's another installation run, choosing a more custom installation:

```
gpioc4!sean# ./cmiop-install
VMEIO 6.1.2 installation procedure
```

This procedure installs all software to use the VMEIO and StorageTek hardware on a Thinking Machines CMiop system. Software installed includes:

- CM filesystem user commands
- CMiop tape commands
- CM filesystem libraries and system commands

- VMEIO, Ciprico, and StorageTek device drivers

Is this install being done on a TMC CMIOP system [y/n/help]? [y] : n

Location to install user and tape commands (path,skip) ? [/usr/local/bin] : /home/bin

Installing user commands and tape software in /home/bin

Location to install system commands and verifiers (path,skip) ? [/usr/local/etc] : /home/etc

Installing system commands and verifiers in /home/etc

Location to install CMFS library (path,skip) ? [/usr/local/lib] : /home/lib

Installing library files in /home/lib

Location to install header files (path,skip) ? [/usr/include] : /home/include

Installing header files in /home/include

Do you want to update the system files (rc.local, crontab, etc)? [y] :

Updating /etc/rc.local

Connection machine daemons already in /etc/rc.local

vioconfig already in /etc/rc.local

Updating crontab

Updating services

Adding cm-logger-conn to /etc/services ...

You are running the Yellow Pages service

The cm-logger service is correctly installed in the Yellow Pages

The cm-logger-conn service is correctly installed in the Yellow Pages

Do you want to build a kernel for this system (y/n)? [y] :

Where do you build your kernels? [/sys] :

Configuring kernel...

Doing a "make depend"

System configuration complete

Building kernel...

loading vmunix

rearranging symbols

text data bss dec hex

942080 150864 62912 1155856 11a310

New kernel built successfully

Installing new kernel as /vmunix

(old kernel saved as /vmunix.old)

Creating special device files

CMIO software installation is complete. You should edit /etc/rc.local to set up the proper CMIO bus id and station for the vioconfig command. Also set up the cm-logger parameters appropriate to your site.

Then reboot the system.

```
gpioc4!sean#
```

In this installation, the administrator chose to install the software in subdirectories of a filesystem called */home*.

5. What gets installed where

In this section, it is assumed that you are not installing the VMEIO software in the standard CMIOP configuration.

The VMEIO software is broken up into several parts, each of which may be installed in a different location. The installation also allows you to install code in your system's *rc.local* startup script to enable the CM error logging daemon and to configure your hardware at boot time, and to install the VMEIO, Ciprico, and StorageTek device drivers and configure a kernel. The latter two functions are performed by separate shell scripts.

User and Tape commands

The user and tape commands consist of the CM Filesystem utility commands that correspond to the standard Unix filesystem utilities (for instance, *cmls*, *cmcp*, *cmmv*, etc, as well as *copyfromdv* and *copytodv*). The *ibmtape* utility is also included in this software component. With the exception of *ibmtape*, all of these commands are shipped with the CM Base System software release, so if you're installing on a CM2 or CM200 front-end, you can skip this section.

This software is found in the *usr/bin* subdirectory of the installation area.

System Commands and Verifiers

This section contains commands typically run by the system administrator such as *cmmknod*, daemons typically started in *rc.local* such as *cm-logger* and *fsserver*, the diagnostics for the VMEIO hardware, and a series of exercisers that allow you to verify the basic integrity of the hardware.

This software is found in the *usr/etc* subdirectory of the installation area.

CMFS library

The VMEIO software release also includes the CM Filesystem interface library. This component is only needed if you intend to write applications on the CMIOP that perform serial I/O; it can be safely skipped if you won't be doing this, or if you are installing on a CM2/CM200 front end.

This software is found in the *usr/lib* subdirectory of the installation area.

Header Files

Several header files needed to write serial I/O applications are included with this release -- they are typically installed in */usr/include/cm*. They can be safely skipped if you do not intend to write your own applications; however, they do not overlap the CM Base System release.

This software is found in the *usr/include/cm* subdirectory of the installation area.

Updating System Files

The shell script *update-system-files* automatically inserts code in the *rc.local* file of your machine that will start the CM error logger and the CMFS fileserver, as well as code to configure the VMEIO board in your system when it is rebooted. If you normally run code like this via a different mechanism, this section can be skipped. If you do use this section of the installation process, you will need to manually edit this file to insert the appropriate Bus ID and Station ID for your VMEIO board (this depends on how you have configured your CMIO Bus).

This section of the installation also starts a cron job that runs at 4:05am on Sunday which "rotates" the error logs produced by the *cm-logger* program. Up to five logs are kept in */usr/spool/cm*, with the oldest log being removed. This prevents spool space on your host from filling up over time. It also ensures that two new services, *cm-logger* and *cm-logger-conn* are in your services database.

Building Kernels

If you choose to build a preconfigured kernel, the installation script *update-device-driver* will install several new files in your kernel build area: */sys/sundev/sc_conf.c*, */sys/sun/conf.c*, and */sys/sun4/conf/CMIO.P*. Any existing copies of these files are backed up. It will then configure the "CMIO.P" kernel and install it in the root directory. If you have any custom or non-standard device drivers installed in your system, you should skip this option by answering "n" to the script's prompt, and install the device driver manually. If you answer "y", your system will not be damaged in any way, but support for the device drivers you have added before running the script will be removed.

For more information about installing the VMEIO device drivers by hand, consult Appendix A.

6. Testing the System

To test out the system, you should first run *viodiag* to check the VMEIO hardware. Instructions on running *viodiag* are given in the *VMEIO Installation and Service Guide* provided with your VMEIO hardware. This will allow you to confirm that you have installed the VMEIO hardware and that it is in working condition.

If you are installing this software on a CMIO.P system, you may wish to verify that the tape drive hardware is working correctly and that the connections from the CMIO.P to the Ciprico controller to the StorageTek tape drive are in proper working order. There is an exerciser provided with the release called *TapeDVxfervfr* (for "Tape to DataVault transfer verifier"). This utility writes a file from the DataVault to a scratch tape, reads it back, and compares the resulting data. It does this at a user-specifiable blocksize, and uses random lengths of data and offsets. Here is a sample run of the utility:

```
gpioc4!sean# TapeDVxfervfr
DIAGNOSTIC EXECUTIVE FOR DV/Tape Tests
State: RELEASE-6-1 Date: 1991/08/13 11:57:43 State: RELEASE-6-1
```

```
<test_DV_to_Tape-DIAG> tape-dv-xfer-vfr
Usage:
```

```
tape-dv-xfer-vfr
  Arguments:
  'tape_drive' = /dev/<drive-name>
  'DV_file'    = dvault:<path>/<file-name>
  'user_blksize' = Tape Block size
  'vmeio_unit' = Vmeio unit number
```

```
tape-dv-xfer-vfr
```


Arguments:

(null)
Please Input: tape_drive ? /dev/rstc0
Please Input: DV_file ? test-file
Please Input: user_blksize ? 2048
Please Input: vmeio_unit ? 0

. Tue Aug 25 14:00:24 1992 -> tape-dv-xfer-vfr

***TEST 0

Beginning at memory offset 0 for a length of 49 (disk block 0, offset 0)...
using System memory...
writing and reading to Disk file test-file...
*** Using VMEIO board 0 for serial transfer to Data Vault...***
writing and reading to Tape...

***TEST 1

Beginning at memory offset 49 for a length of 32768 (disk block 0, offset 49)...
using VME board 0 memory...
writing and reading to Disk file test-file...
*** Using VMEIO board 0 for serial transfer to Data Vault...***
writing and reading to Tape...

***TEST 2

Beginning at memory offset 32817 for a length of 65536 (disk block 2, offset 49)...
using System memory...
writing and reading to Disk file test-file...
*** Using VMEIO board 0 for serial transfer to Data Vault...***
writing and reading to Tape...

***TEST 3

Beginning at memory offset 98353 for a length of 16335 (disk block 6, offset 49)...
using VME board 0 memory...
writing and reading to Disk file test-file...
*** Using VMEIO board 0 for serial transfer to Data Vault...***
writing and reading to Tape...

[...etc...]

***TEST 31

Beginning at memory offset 2095616 for a length of 20 (disk block 127, offset 14848)...
using VME board 0 memory...
writing and reading to Disk file test-file...
*** Using VMEIO board 0 for serial transfer to Data Vault...***
writing and reading to Tape...Using fixed block mode with a block size of 2048...
Tape Transfer size is 0...

***TEST 32

Beginning at memory offset 2095616 for a length of 1536 (disk block 127, offset 14848)...
using System memory...
writing and reading to Disk file test-file...
*** Using VMEIO board 0 for serial transfer to Data Vault...***
writing and reading to Tape...
<test_DV_to_Tape-DIAG>

<test_DV_to_Tape-DIAG> quit

Appendix A) Installing the device drivers by hand

The *cmiop-install* script assumes that the VMEIO, Ciprico, and StorageTek drivers are the only non-standard drivers you will be installing under SunOS. If you have installed other device drivers, you will have to install support for these drivers by hand. It is highly recommended that you review the section on installing device drivers in the *SunOS System Administration Manual* before committing yourself to this procedure!

Installing the VMEIO device driver requires editing a few files in your kernel build area (usually */sys* or */usr/kvm/sys*). Note that the support for the StorageTek 4980 tape drives via the Ciprico and StorageTek drivers is not supported on standard Suns, only on a TMC CMIOP system.

First, entries for the three drivers must be added to the *cdevsw[]* jump table in */sys/sun/conf.c*. This can be done by appending the file *sys/sun/conf.c.additions* to *conf.c*, and noting what the major device numbers they're added at. The major device number is derived from the index of the device into the *cdevsw[]* array, and is usually denoted in a comment on the right side of the structure definition. For vanilla SunOS, the VMEIO is major device number 104.

Then, the *config* program must be told how to build a kernel that includes the VMEIO device driver. To do this, add the following line to */sys/sun4/conf/files*:

```
tmc/vmeio.c          optional vmeio device-driver
```

When the kernel is configured, this line will tell *config* what files to compile to support the device.

Finally, the actual device driver code has to be installed. The easiest way to do this is to copy the entire directory *sys/tmc* into */sys* in your kernel build area.

At this point, you should be ready to build a kernel with VMEIO support in it. To do so, pick your favorite kernel config file, and add the following definition:

```
#
# Connection Machine I/O system (VMEIO) interfaces
# Make sure that the interrupt vector is unique in
# your system!
#
device      vmeio0 at vme16d32 ? csr 0x8000 flags 0x80000000
             priority 2 vector vmeio_intr 0xcc
```

Then run *config* on the config file and make your kernel.

After rebooting with the new kernel, you have to make the device special files in */dev* to use the VMEIO. To do this, issue the following commands with the major device number from *conf.c*:

```
# mknod /dev/cmio0 c major-device-number 0
# chmod 666 /dev/cmio0
```

At this point you should be able to use the VMEIO.