1. too many encodings will cause a core dump.

octmm ..

1. does not understand '-o cell:view' only '-o :view'.

oct2ps ..

1. does not handle *pac man* shapes.

puppy ..

1. does not understand '-o :view' only '-o cell:view'.

sparcs ..

1. generates cryptic (and generally unusable) error messages.
2. **rpc** version can only be run once, and then must be restarted.

vem ..

1. can enter things via VEM that **sparcs** will complain about. Solution: run **symclean**.
2. occasionally VEM will not correctly reset after an RPC application dies and the user must issue the : `rpc-reset` command before running another RPC application.

Chapter 6

Known Bugs in Octtools 3.5

general ..

1. some tools are bad about reporting policy violations. They may exit when a property or bag can not be found without giving the user an error message that adequately describes the problem.

bdnet ..

1. `TECHNOLOGY`, `EDITSTYLE`, and `VIEWTYPE` are keywords and thus must be quoted if used for property names.
2. does not handle joined bags properly.

gem ..

1. does not place well contacts.

mag2oct ..

1. does not correctly handle point or line terminals.

misII ..

1. read_oct throws away mapping information.
2. write_oct does not handle joined bags properly.

mizer ..

1. does not understand '-o :view' only '-o cell:view'.

mustang ..

Chapter 5

Bug Reporting

We are very interested in getting bug reports. There is a bug reporting facility that can be used (**bug-report**); to run it:

```
~octtools/utils/bin/bug-report
```

This will put you into an editor (based on your EDITOR environment variable) with a template of a bug report. Please fill in the template, save the file, and exit the editor. The bug report will be sent to:

```
octtools-bugs-processor@eros.berkeley.edu
```

If your site can not mail directly to this address, you should modify the address in the **bug-report** script. This is an automated mail handler, so please do not send ordinary mail to the this address. If you can not use the bug reporting facility, please mail your bug reports to:

```
octtools-bugs@eros.berkeley.edu
...!ucbvax!eros!octtools-bugs
```

`bad inet connect`

Your **rpc** application and **vem** have been built with different versions of the operating system. This has been seen between ULTRIX 2.2 and ULTRIX 3.0, and between SUN OS 3.2 and SUN OS 4.0.

```
Can't open display dent:0
% printenv DISPLAY
dent:0                  <-- wrong display name
% setenv DISPLAY shambhala:0
% vem
```

3. **vem** beeps whenever you try to issue any command.

   Usually due to **vem** not being able to find its `vem.bindings` file. This should be installed in `octtools/lib/vem/vem.bindings.6,7-0`. If the file is in the proper place, make sure that you are running a version of **vem** make sure that you are not overriding the default location in your `.Xdefaults` file. You change change where **vem** looks for the `vem.bindings` file by setting the **vem.bindings** default. If you use a different `vem.bindings` file, make sure that it is up to date and uses `INCLUDESTD` to get the standard bindings.

4. **vem** beeps whenever I try certain commands (but not all commands).

   Make sure that you are in the correct window for that command. Some commands in **vem** are window-specific, for example, `save-all` only works in the console window and `create-geometry` only works in physical editing windows.

5. An **rpc** application is invoked from **vem**, but does not start.

   First, check to see if the message ``` ``NAME: Command not found'' ``` appears in the window that you started **vem** from. This means that the location specified for the **rpc** application (either at invocation time via `rpc-any` or in the `vem.bindings` file) is incorrect.

   Second, check to see if the message ``` ``Permission Denied'' ``` appears in the window that you started **vem** from. **rpc** uses the UNIX *rsh* mechanism for spawning off remote commands and `rsh` requires that a file called `.rhosts` exist that contains a list of machines (and optionally users) that can remotely execute commands (and remotely login) to your account. This file is in your home directory. For example, if you are running **vem** on a machine called 'shambhala' and want to spawn off a remote application to a machine called 'dent', you would need a `.rhosts` file in your home directory on 'shambhala' containing the line 'bob'. Based on your machine's configuration, even if you are spawning the remote application to the same machine that you are running **vem** on, you may have to put your local machines name in your local `.rhosts` file. So if you are running **vem** on 'shambhala' and spawning remote applications on 'shambhala', you need to put 'shambhala' in your `.rhosts` file.

   Finally, if after a minute or two, you get the error message:

Chapter 4

Common Installation / Initial Usage Problems

1. **vem** dies immediately with the message:

```
XIO Error: Broken Pipe
```

   You must `xhost` the machine running **vem** on the machine displaying **vem**.

```
(machine that is running vem)
% vem
XIO Error: Broken Pipe

(machine that should display vem)
% xhost +machine-that-is-running-vem
```

2. **vem** dies immediately with the message:

```
Can't open display NAME
```

   Either your `DISPLAY` environment variable is incorrect (wrong name or has not been set) or you have specified a bad display name on the **vem** command line.

```
% vem
```

Chapter 3

Changes From Oct Tools 3.4 to Oct Tools 3.5

See the file ./CHANGES for a list of changes from 3.4 to 3.5.

See the top-level `Makefile` for other installation options.

Machine (OS)	Approximate Build Time (hours)
DEC VAX 8650/8800	3-4 hours
Sequent Symmetry	2.5-3 hours (10 processors)
DECSystem 5400	3.5 hours
DECStation 3100	3-4 hours
SUN 4/280	4-5 hours (NFS file system)
SUN 3/160	7-8 hours (NFS file system)
DEC MicroVax II	16-17 hours (NFS file system)

- The following documents will get you started:

```
./document/PostScript/examples.ps.Z
./document/PostScript/physical.ps.Z
./document/PostScript/symbolic.ps.Z
./document/PostScript/schematic.ps.Z
```

- Please mail to:

```
octtools-maintainers-request@eros.berkeley.edu
..!ucbvax!eros!octtools-maintainers-request
```

so you can be placed on the updates and bug fixes mailing list. Whenever we make updates or fix bugs, we will mail out the updates and bug fixes to you.

./demo/vem - example cells to browse/edit in vem

./document/PostScript - tutorials, examples, installation notes.

./lib - contains library information for the programs, package libraries, and the technology descriptions. The technology descriptions are stored in:

./lib/technology/NAME/ - where "NAME" is the name of the technology.

The following technologies come with the distribution:

./lib/technology/scmos - a scalable CMOS technology. MCNC place and route benchmarks.

./lib/technology/gaas - a galium-arsenide technology.

./lib/technology/interconnect - a technlogy containing 9 interconnect layers, used for experimenting with N-layer routers.

./src - the source directories for each of the tools and packages. For example:

./src/avl - source directory for the AVL tree package.

./src/vem - source directory for the VEM graphics editor.

./unsupported - unsupported software.

- To compile and install the OCT tools, type:

```
INSTALL
```

This script will compile and install the programs, program libraries, technology descriptions, and documentation in the appropriate place.

The final directory structure will have the following directories added:

./bin - where the tools will be installed; created during the installation.

./bin/rpc - where the rpc tools will be installed.

./doc - contains users guides for some of the programs; created during the installation.

./include - contains the include files for the packages; created during the installation.

./man - contains the manual pages for each of the tools and packages; created during the installation. It is further subdivided into:

./man/man1 - man pages for programs

./man/man3 - man pages for packages

./man/man5 - man pages for file formats

Chapter 2

Installation

To install the tools, follow the steps outlined below:

- Create an account named **octtools**. This is where the Octtools 3.5 binaries, libraries, manual pages, and other documents will be installed.
- Read the contents of the Oct Tools tape (unpacking the required parts of the distribution requires 75 megabytes):

```
cd ~octtools

# the following is required
tar x ./base.tar.Z
zcat base.tar.Z | tar xf -
rm base.tar.Z

# the following steps are optional
tar x ./demo.tar.Z
zcat demo.tar.Z | tar xf -
rm demo.tar.Z

tar x ./document.tar.Z
zcat document.tar.Z | tar xf -
rm document.tar.Z

tar x ./unsupported.tar.Z
zcat unsupported.tar.Z | tar xf -
rm unsupported.tar.Z
```

The directory structure created will be:

./demo - contains demonstrations of some of the programs and libraries. For example:

./demo/synthesis - synthesis path examples

Chapter 1

Introduction

The Octtools 3.5 distribution is a collection of over 90 programs and libraries, by over 20 authors, which together form a system for IC design. The system includes tools for multi-level logic synthesis, standard- and macro-cell placement and routing, module-generation, custom-cell design, translation to and from standard data interchange formats, and a variety of utility programs for manipulating symbolic and geometric design data. Most tools are integrated with the OCT data manager and the VEM graphics editor.

The software is provided with complete source and documentation on one 9-track, 1600 BPI UNIX `tar` format tape, along with a printed copy of the documentation.

Please note that most of these tools are the results of previous or ongoing research projects and do not form a turnkey or production system. They have bugs that we have not found as of yet and do not always work together as well as they should. While they work for us (CAD researchers, IC designers, VLSI classes), other Universities, and Industrial Sites, you may have to modify them to fit them into your CAD system.

EDIF

We also have a distribution of `EDIF` tools (including translators to and from OCT). For more information contact Wendell Baker (wbaker@ic.berkeley.edu). To order "EDIF Tools", contact:

> Cindy Manly
> Industrial Liaison Office
> 479 Cory Hall
> University of California
> Berkeley, California 97420

Contents

1 Introduction 2

2 Installation 3

3 Changes From Oct Tools 3.4 to Oct Tools 3.5 6

4 Common Installation / Initial Usage Problems 7

5 Bug Reporting 10

6 Known Bugs in Octtools 3.5 11

Release Notes for Oct Tools 3.5

Rick L. Spickelmier

Electronics Research Laboratory
University of California, Berkeley

March 6, 1990