

OSCAR Cluster User's Guide  
Software Version 4.0  
Documentation Version 4.0

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Individual Packages</b>	<b>3</b>
2.1	Using C3	3
2.1.1	Overview	3
2.1.2	Basic File Format	4
2.1.3	Specifying ianges	5
2.1.4	Machine Definitions	5
2.1.5	Other Considerations	6
2.2	HDF5	6
2.3	Using LAM/MPI	6
2.3.1	Notes about OSCAR	



The default method of execution for the tools is to run the command on all cluster nodes concurrently.

### 2.1.3 Specifying ranges

Ranges can be specified in two ways, one as a range, and the other as a single node. Ranges are specified by the following format:  $m-n$ , where  $m$  is a positive integer (including zero) and  $n$  is a number larger than  $m$ .

### **2.1.5 Other Considerations**

In most cases, C3 has tried to mimic the standard Linux command it is based on. This is to make using

LAM features a full implementation of MPI-1, and much of MPI-2. (Compatibility)

```
$ lamboot my_hostfile
```

2. Repeat the following steps as many times as necessary:

(a)





The “hello world” example shipped with PVM demonstrates how one can compile and run a simple application outside of PBS. The following screen log highlights this for a standard user *sgrundy* (Solomon Grundy).

```
# Create default directory for PVM binaries (one time operation)
sgrundy: $ mkdir -p $HOME/pvm3/bin/$PVM_ARCH

# Copy examples to local 'hello' directory
sgrundy: $ cp $PVM_ROOT/examples/hello* $HOME/hello-example
sgrundy: $ cd $HOME/hello-example

# Compile a hello world, using necessary include (-I) and library
# (-L) search path info as well as the PVM3 lib.
sgrundy: $ gcc -I$PVM_ROOT/include hello.c -L$PVM_ROOT/lib/$PVM_ARCH \
> -lpvm3 -o hello
sgrundy: $ gcc -I$PVM_ROOT/include hello_other.c -L$PVM_ROOT/lib/$PVM_ARCH \
> -lpvm3 -o hello_other
```

or edit any “.dot” files. You can compile and run the “hello world” example in this fashion by using a simple hostfile as shown here.

The example below uses the same “hello world” program that was previously compiled, but using a hostfile with the appropriate options to override the default execution and working directory. Remember that the “hello” program exists in the /home/sgrundy/hello-example directory:

```
sgrumpy: $ cat myhostfile
*   ep=/home/sgrundy/hello-example  wd=/home/sgrundy/hello-example
oscarnode1
```

The options used here are:

## 2.8 An overview of SIS

The System Installation Suite, or SIS, is a tool for installing Linux systems over a network. It is used in OSCAR to install the client nodes. SIS also provides the database from which OSCAR obtains its cluster configuration information.

The main concept to understand about SIS is that it is an *image based* install tool. An image is basically a copy of all the files that get installed on a client. This image is stored on the server and can be accessed for customizations or updates. You can even `chroot` into the image and perform builds.

Once this image is built, clients are defined and associated with the image. When one of these clients

- mksidisk
- mksirange
- mksimachine
- systemconfigurator
- updateclient

You can also access the mailing lists and other docs through the sisuite home page, <http://sisuite.org/>.



the “default” attribute on the “mpi” tag to a given value will control which MPI implementation is loaded into the environment.

`env-switcher` operates at two different levels: system-level and user-level. The system-level tags, attributes, and values are stored in a central location. User-level tags, attributes, and values are stored in each user’s `$HOME` directory.

When `env-switcher` looks up entity that it manipulates (for example, to determine the value of the “default” attribute on the “mpi” tag), it attempts to resolve the value in a specific sequence:

1. Look for a “default” attribute value on the “mpi” tag in the user-level defaults
2. Look for a “default” attribute value on the “global” tag in the user-level defaults
3. Look for a “default” attribute value on the “mpi” tag in the system-level defaults
4. Look for a “default” attribute value on the “global” tag in the system-level defaults

In this way, a four-tiered set of defaults can be effected: specific user-level, general user-level, specific system-level, and general system-level.

The most common `env-switcher` commands that users will invoke are:

1. `switcher --list`  
List all available tags.
2. `switcher <tag> --list`  
List all defined attributes for the tag `<tag>`.
3. `switcher <tag> = <value> [--system]`  
A shortcut nomenclature to set the “default” attribute on `<tag>` equal to the value `<value>`. If the `--system` parameter is used, the change will affect the system-level defaults; otherwise, the user’s personal user-level defaults are changed.
4. `switcher <tag> --show`  
Show the all attribute / value pairs for the tag `<tag>`. The values shown will be for attributes that have a resolvable value (using the resolution sequence described above). Hence, this output may vary from user to user for a given `<tag>` depending on the values of user-level defaults.
5. `switcher <tag> --rm-attr <attr> [--system]`  
Remove the attribute `<attr>` from a given tag. If the `--system` parameter is used, the change will affect the system level defaults; otherwise, the user’s personal user-level defaults are used.

Section [2.9.3](#) shows an example scenario using the `switcher` command detailing how to change which MPI implementation is used, both at the system-level and user-level.

See the man page for `switcher(1)` and the output of `switcher --help` for more details on the `switcher` command.

### 2.9.3 Which MPI do you want to use?

OSCAR has a generalized mechanism to both set a system-level default MPI implementation, and also to allow users to override the system-level default with their own choice of MPI implementation.

This allows multiple MPI implementations to be installed on an OSCAR cluster (e.g., LAM/MPI and

#### **2.9.4 Setting the system-level default**

The system-level default MPI implementation can be set in two different (yet equivalent) ways:

- 1.



### 2.9.5 Setting the user-level default

Setting a user-level default is essentially the same as setting the system-level default, except without the `--system` argument. This will set the user-level default instead of the system-level default:

```
$ switcher mpi = lam-1.2.3
```

Using the special name `none` will indicate that no module should be loaded for the `mpi` tag. It is most often used by users to specify that they do not want a particular software package loaded.

```
$ switcher mpi = none
```

Removing a user default (and therefore reverting to the system-level default) is done by removing the `default` attribute:

```
$ switcher mpi --show
user:default=mpich-1.2.4
system:exists=true
$ switcher mpi --rm-attr default
$ switcher mpi --show
system:default=lam-6.5.6
system:exists=true
```

### 2.9.6 Use `switcher` with care!

`switcher` immediately affects the environment of all future shell invocations (including the environment of scripts). To get a full list of options available, read the `switcher(1)` man page, and/or run `switcher --help`.

## 3 Package Licenses and Copyrights

provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice appear in supporting documentation.

Neither the Oak Ridge National Laboratory nor the Authors make any representations about the suitability of this software for any purpose. This software is provided "as is" without express or implied warranty.

The C3 tools were funded by the U.S. Department of Energy.

The Python package contains:

This is Python version 2.1.1  
=====

Copyright (c) 2001 Python Software Foundation.  
All rights reserved.

Copyright (c) 2000 BeOpen.com.  
All rights reserved.

Copyright (c) 1995-2001 Corporation for National Research Initiatives.  
All rights reserved.

Copyright (c) 1991-1995 Stichting Mathematisch Centrum.  
All rights reserved.

License information  
-----

See the file "LICENSE" for information on the history of this software, terms & conditions for usage, and a DISCLAIMER OF ALL WARRANTIES.

This Python distribution contains no GNU General Public Licensed (GPLed) code so it may be used in proprietary projects just like prior Python distributions. There are interfaces to some GNU code but these are entirely optional.

All trademarks referenced herein are property of their respective holders.

## **3.2 Disable Services**

The `disable-services` package contains the following copyrights and licenses.

liability to any recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise.

3. In addition, redistributions of modified forms of the source or binary code must carry prominent notices stating that the original code was changed and the date of the change.
4. All publications or advertising materials mentioning features or use of this software are asked, but not required, to acknowledge that it was developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign and to credit the contributors.
5. Neither the name of the University nor the names of the Contributors may be used to endorse or promote products derived from this software without specific prior written permission from the University or the Contributors, as appropriate for the name(s) to be used.
6. THIS SOFTWARE IS PROVIDED BY THE UNIVERSITY AND THE CONTRIBUTORS "AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. In no event shall the University or the Contributors be liable for any damages suffered by the users arising out of the use of this software, even if advised of the possibility of such damage.

-----  
 Portions of HDF5 were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL).

The following statement applies to those portions of the product and must be retained in any redistribution of source code: -----d-a

statement of the University name of the University nor be used to endorse or promote products derived from this software without specific prior written permission from the University or the Contributors, as appropriate for the name(s) to be used.

disclosed, or represents that its use would not infringe privately-owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the









Resources, Inc.

THE SOFTWARE IS PROVIDED AS IS AND THE SUPERCLUSTER DEVELOPMENT GROUP (SDG) AND ALL CONTRIBUTING PARTIES DISCLAIM ALL WARRANTIES RELATING TO THE SOFTWARE, WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER SDG NOR ANYONE INVOLVED IN THE CREATION,

this code or derivations of this code in any form whatsoever, including parts of SOFTWARE incorporated into other software programs without express written permission from Cluster Resources, Inc.

#### 4. Modifications

SOFTWARE may be freely modified as necessary to meet the needs of the licensee's system. Licensees which wish to distribute their modifications (including government and academic institutions, and those with a written license to redistribute) must first send a copy of the modifications along with descriptive text to the Supercluster Development Group at [support@supercluster.org](mailto:support@supercluster.org). Those without license to redistribute may send modifications to the Supercluster Development Group for evaluation and possible incorporation into SOFTWARE. Failure to send a copy of distributed modifications renders the license invalid, as well as any licenses granted to third parties subsequent to the incorporation of the modifications into SOFTWARE.

Upon receipt of modification notice, Supercluster Development Group, may, at their disgression, send the submittor a nice hat or a short



The following is a notice of limited availability of the code, and disclaimer which must be included in the prologue of the code and in all source listings of the code.

Copyright Notice

- + 1993 University of Chicago
- + 1993 Mississippi State University

Permission is hereby granted to use, reproduce, prepare derivative works, and to redistribute to others. This software was authored by:

Argonne National Laboratory Group

W. Gropp: (630) 252-3r7.099 232 National Laboratory,xNimStail9se-mail:  
gropp@mcs.anl.gov





The Syslinux package contains:



Alternatively, this acknowledgement may appear in the software itself,