GA0014

Fermilab

OCS Installation/Administration Guide

Dorota Genser
Marilyn Schweitzer
George Szmuksta

8/5/99

Version 3.2

Abstract

Operator Communications Software (OCS) is a package that performs operator assisted tape mounts, tape drive allocation and tape drive tracking/management. This document how to install and administer OCS.

Fermi National Accelerator Laboratory
Batavia, IL 60510

Table of Contents

1.0 Introduction

In the following sections this document describes how to:

- Build Operator Communications Software (OCS) executables.
- Distribute OCS to remote hosts
- Setup the appropriate user environment for OCS
- Install OCS on hosts
- Administer tape drives into the OCS database, authorize user to have access to the drives and, if desired, administrative privileges.
- Setup Operator environment
- Test the OCS configuration.
- Perform routine maintenance.
- Troubleshooting problems

The first step is to unwind the OCS source archive. For example, if you wanted to unwind the archive in the directory:

```
/usr/products/ocs/v3_2
```

You would:

```
cd /usr/products/ocs/v3_2
tar xvf ocs_src.tar
```

where

```
ocs_src.tar
```

is the name of the OCS source archive. Since this archive contains source and documentation but no executables, thus the next step will be to build the executables. For the rest of this document we will assume the environment variable OCS_DIR has been set to the name of the directory that the source archive was unwound. For example:

C-shell Users	**Bourne/K-shell Users**
`setenv OCS_DIR /usr/products/ocs/v3_2`	`OCS_DIR=/usr/products/ocs/v3_2`
	`export OCS_DIR`

2.0 Building Executables

The ***ftt*** package[1] (version v2.0 or higher) is needed to build OCS. The FTT_DIR environmental variable needs to be set to the full name of the directory were the product is installed.

To build the OCS executables and libraries:

```
cd $OCS_DIR
make
```

Once the compilation is complete, the appropriate permissions must be granted to the OCS system daemons:

```
cd $OCSBIN
chown root.sys taped devd dbserv
chmod ug+s taped devd
```

Packaged with OCS are some demonstration/sample user programs. These programs are also useful for testing OCS. To build these executables, the OCS user environment must be setup. To set the OCS user environment:

C-shell Users	**Bourne/K-shell Users**
`cd OCS_DIR`	`cd OCS_DIR`
`source ups/setup.csh`	`. ups/setup.sh`

Then to build these executables:

```
cd examples
make
```

3.0 Other Makefile Features

To remove the intermediate object files used in building the executables, enter:

```
cd $OCS_DIR
make clean
```

To make an archive of $OCS_DIR including source files and excluding binaries, enter:

```
cd $OCS_DIR
make archive
```

The tar output file, *ocs_src.tar*, will be created in $OCS_DIR.

To make an archive of $OCS_DIR including binaries and excluding the source files, enter:

```
cd $OCS_DIR
make bintar
```

The tar output file, *ocs_bin.tar*, will be created in $OCS_DIR.

1. Fermi Tape Tools

To run make on only a sub-directory of the source, e.g. *src/xocs*:

C-shell Users	Bourne/K-shell Users
`cd OCS_DIR`	`cd OCS_DIR`
`source env.csh`	`. env.sh`
`cd src/xocs`	`cd src/xocs`
`make`	`make`

4.0 Binary Distribution

If a host does not require the source, but users need to link to OCS on that host, then the following files and directories should be copied to that host:

```
README, ups, bin, lib, include, doc, examples, templates
```

For example:

```
rdist -c  {README,ups,bin,lib,include,doc,examples,templates} \
          products@cdfsga:/usr/products/ocs/v3_2
```

5.0 User Environment Setup

To setup the correct environment for using OCS on hosts that have the Fermilab UPS product installed, users need only enter:

```
setup ocs
```

once OCS has been declared to UPS.

If a host does not have the Fermilab UPS product installed, users will have to set the OCS_DIR environment variable themselves and execute the appropriate setup script. For example:

C-shell Users	Bourne/K-shell Users
`cd OCS_DIR`	`cd OCS_DIR`
`source ups/setup.csh`	`. ups/setup.sh`

6.0 Installation

6.1 Installation Overview

After the OCS executables have been built and distributed to a host, the OCS system daemons need to be installed. There are three such daemons:

- The database server, *dbserv*, uses TCP and is started by *inetd*, the internet super-server, when the first RPC call is issued to it. The *dbserv* is totally passive; it replies to RPC calls and does not initiate any RPC calls. Any number of hosts can share an OCS database, however, when the node with the database is down, none of the hosts that use that database can access OCS for tape mounts. Of the hosts that share an OCS database, only one will have the *dbserv* installed on it. For example, if you have 4 hosts:

  a) You may decide to have all 4 hosts share the same database in which case only 1 host of the 4 will have a *dbserv* installed on it.

  b) You may decide to have 3 of the hosts share 1 database and the remaining host have its own database. In this case you will need to install a *dbserv* on 2 of the 4 hosts.

- A *taped* must be installed on each host which is going to issue OCS commands such as tape drive allocations, mount requests, mount replies, tape drive status, etc. The *taped* need not be installed on a host if:

  a) the host is just going to be used for linking executables with OCS

  b) all tape drive allocations and mount requests for the host will be done through a remote host and no device file manipulation is desired on the part of OCS

  This latter case is how the Fermi ACPMAPS platform uses OCS.

  Like the *dbserv*, the *taped* is started by *inetd* when the first RPC call is issued to it. The taped uses TCP for except to communicate with a *taped* on an other host. In that case, UDP is used along with a retry algorithm internal to OCS.

- The *devd* is started by the local *taped* on an as needed basis and uses TCP on a transient RPC number. The *devd* handles commands such as tape rewinds which can be time consuming and block until completed. At any given time there will be at most one *devd* running per tape drive. If a *devd* has no activity for 10 minutes it will shut itself down.

OCS provides two tools to install these daemons; a GUI interface called *ocs_install* and a command line interface called *ocs_terminst*. Both of these tools must be run as root and require the following information:

- Database Binaries Directory:

  This directory is the pathname where the *dbserv* executable is located and should be on a local file system so that *inetd* can find it when the host is re-booted. Thus, if the binary directory containing the binaries for the entire product is NFS mounted, the *dbserv* should be copied to a local file system. By default this directory is */usr/spool/ocs/current/bin*

- Database Directory:

  This directory is where the database will physically be stored on the disk. It too should be on a local file system. By default this directory is */usr/spool/ocs/DB*.

- Database RPC Number:

  This is the RPC program number used to communicate to the *dbserv*. By default this number is *211637* and should not be changed except in unusual circumstances.

- Database Instance Number:

  Normally there will only be 1 *dbserv* per host and NIS domain, so by default the this number is 1. However, if there is a need to install more than one *dbserv* per host or NIS domain, the Database Instance Number will be appended to the *inetd* and *portmap* service name (e.g. dbserv2). Care must be taken choosing the Database Binaries Directory, Database Directory, and Database RPC Number do not overlap with other instances and so that System Managers can easily discern the connectivity.

- Taped Binaries Directory:

  This directory is the pathname where the *taped* and *devd* executable is located and should be on a local file system so that *inetd* can find it when the host is re-booted. Thus, if the binary directory containing the binaries for the entire product is NFS mounted, the *taped* and *devd* should be copied to a local file system. By default this directory is */usr/spool/current/bin.*

- Database Hostname:

  This is the name of the host that a local *taped* and its clients will use to communicate to the *dbserv*. By default this is the local host name, but should be changed is a *dbserv* is not going to be installed on the local host.
  The database hostname must be of the form:

  <Database RPC Number>@<Database Hostname>

  e.g.:

  211999@cdfsga

  if the Database RPC Number is other than the default value.

- Taped Working Directory:

  This directory is where the *taped*'s temporary files and the name of the database host for that *taped* will be stored. This directory should also be on a local file system. By default this directory is */usr/spool/ocs/taped.*

- Taped RPC Number:

  This is the RPC program number used by clients to communicate to the local *taped* and by all *taped*s that share the same OCS database to communicate with each other. Thus, all *taped*s that share a database must have the same RPC number. By default this number is *217843*

- Taped Instance Number:

  Normally there will only be 1 *taped* per host and all *taped*s within a NIS domain will have the same Taped RPC Number (e.g. all will communicate with the same *dbserv*), Thus, by default the Taped Instance Number is 1. However, if there is a need to install more than one *taped* per host or have multiple OCS databases per NIS domain, the Taped Instance Number will be appended to the *inetd* and *portmap* service name (e.g. taped2). Care must be taken choosing the Taped Binaries Directory, Taped Working Directory, and Taped RPC number do not overlap with other instances and so that System Managers can easily discern the connectivity.

The *ocs_install* and *ocs_terminst* programs update the system *inetd.conf* and */etc/rpc* files so that *inetd* can register the start the *taped* and *dbserv* and start them when needed. Symbolic links for each OCS system daemon are also created in the standard system directory to point to the appropriate binary and working directory. The symbolic links are created in */usr/etc* in on AIX and IRIX platforms and in */usr/sbin* on SunOS (Solaris) and OSF1 platforms

If YP (NIS) is used and the OCS system daemons are being installed on the YP master host, the YP map for */etc/rpc* is updated. *Ocs_install* provides options to update the YP map for */etc/rpc* on the YP master host even if the *dbserv* or none of the OCS system daemons are to be installed on the YP master. *Ocs_install* refers to updating just the YP map for */etc/rpc* as "Update RPC" opposed to "Install".

When either a *taped* or a *dbserv* is installed for the first time and an entry is inserted into the *inetd.conf* file, *inetd* must re-read its configurations file. *Ocs_install* and *ocs_terminst* do this by sending *inetd* a hang-up signal, SIGHUP.

After the OCS system daemons have been installed, *ocs_install* and *ocs_terminst* can be used to:

• Create OCS database tables. The tables will be initially empty.

• Insert the local host into the database.

• Insert user names and ids from */etc/passwd* into the OCS database.

To ensure that OCS is appropriately initialized after a host is re-booted, a customized OCS initialization script should be installed on each taped and dbserv host. This script should execute the *ocs_reboot* command that will deallocate all the tape drives on the local host. Other actions such as automatically starting the Operator GUI program, *xtape*, may also be desirable. *$OCS_DIR/templates* has examples of such scripts for different Operating System Platforms:

Host	**Script**	**Copy To**	**How To Install**
IRIX	ocs	/etc/init.d	`execute /etc/init.d/ocs config`
SunOS (Solaris)	ocs	/etc/init.d	`execute /etc/init.d/ocs config`
OSF1	ocs	/sbin/init.d	`execute /sbin/init.d/ocs config`
AIX	startocs	/etc	`edit /etc/rc.local`

The templates (if installed as is) require that the executable *ocs_reboot* be copied to the same local directory containing the *devd*, *taped* and *dbserv*, e.g. */usr/spool/ocs/current/bin*.

To ensure the Tape Daemon RPC Program Number is set correctly for users, the following files should be copied to */usr/local/etc* and the value of TAPED_RPC_NUM edited to the Tape Daemon RPC Program Number to be used with *ocs_install* or *ocs_terminst*:

```
$OCS_DIR/templates/ocs_local_setup.csh
$OCS_DIR/templates/ocs_local_setup.sh
```

These files may be customized further to turn on/off the OCS round-robin feature via the environment variable OCS_ROUNDROBIN. When OCS_ROUNDROBIN is defined, then a tapedrive will be moved to the end of the database table whenever the drive is deallocated. Thus, all users on a single OCS system need to have this variable define for the round-robin to be effective. When OCS_ROUNDROBIN is not defined, drives at the top of the database will tend to be used more than drives towards the end.

6.2 Example Installation Procedure

These are some pretty generic directions as to how OCS is installed on most Fermilab systems. To simplify the discussion, let suppose that:

- The "make" or binary distribution completed successfully.
- The name of the host you are installing OCS on is:

  `myhost`

- The local file system (NOT one that is NFS mounted) that you wish to use for OCS working directories is:

  `/usr/spool/ocs`

- The OCS database server is to run on this host with RPC program number:

  `211637`

- The OCS tape daemon RPC program number to use for this group of hosts is:

  `217854`

Change these as needed. Now perform the following steps.

a) login to myhost as root.

b) To ensure the Tape Daemon RPC Program Number is set correctly for users:

```
cd $OCS_DIR
cp templates/ocs_local_setup.csh /usr/local/etc
cp templates/ocs_local_setup.sh /usr/local/etc
```

and edit the value of TAPED_RPC_NUM in:

```
/usr/local/etc/ocs_local_setup.csh
/usr/local/etc/ocs_local_setup.sh
```

to match the Tape Daemon RPC Program Number to be used with *ocs_install*, e.g. 217854.

These files may be customized further to turn on/off the OCS round-robin feature (OCS_ROUNDROBIN) for tape drive allocation.

c) Mot OCS installation is done as root using the *ocs_install* command. *Ocs_install* requires the DB_HOST environment to be set to the host where the Database Server will run in addition to the variables set by *ups/setup.csh* or *ups/setup.sh* script. For example, to set all the necessary environment variables for *ocs_install*:

C-shell Users	**Bourne/K-shell Users**
`setenv OCS_DIR /usr/products/ocs/v3_2`	`OCS_DIR=/usr/products/ocs/v3_2`
`setenv DB_HOST myhost`	`DB_HOST=myhost`
	`export OCS_DIR DB_HOST`
`cd OCS_DIR`	`cd OCS_DIR`
`source ups/setup.csh`	`. ups/setup.sh`

d) Make directories for the working directories:

```
mkdir /usr/spool/ocs
mkdir /usr/spool/ocs/DB
mkdir /usr/spool/ocs/taped
```

e) If $OCS_DIR IS a local file system:

```
ln -s $OCS_DIR /usr/spool/ocs/current
```

If $OCS_DIR is NOT a local file system:

```
mkdir /usr/spool/ocs/current/bin
cp $OCSBIN/taped /usr/spool/ocs/current/bin
cp $OCSBIN/devd /usr/spool/ocs/current/bin
cp $OCSBIN/ocs_reboot /usr/spool/ocs/current/bin
cp $OCSBIN/dbserv /usr/spool/ocs/current/bin
```

f) Make sure the permissions are correct on the OCS system daemons, e.g.:

```
chown root.sys /usr/spool/ocs/current/bin/dbserv
chown root.sys /usr/spool/ocs/current/bin/taped
chown root.sys /usr/spool/ocs/current/bin/devd
chmod ug+s /usr/spool/ocs/current/bin/taped
chmod ug+s /usr/spool/ocs/current/bin/devd
```

g) Enter:

```
ocs_install &
```

In the “Install Database Server” window:

a) Set the database directory to “/usr/spool/ocs/DB”

b) Set the binary directory to “/usr/spool/ocs/current/bin”

c) Leave the RPC program number as is, i.e. 211637

d) Press all the “OK” buttons that turned red

e) Press the “Install” button in that window.

In the “Install Tape Daemon” window:

a) Set the working directory to “/usr/spool/ocs/taped”

b) Set the binary directory to “/usr/spool/ocs/current/bin”

c) Set the database host name to “myhost”.

d) Set the RPC program number to 217854.

e) Press all the “OK” buttons that turned red.vi)Press the “Install” button in that window.

Click on the following buttons:

a) “Create DB Tables”

b) “Insert This CPU”

c) “Insert Users”

Click on the “Exit” button to exit *ocs_install*.

h) As a user other than root, execute:

```
ocs_tape
```

You should get the header and a message “No matches found”. If you get an error, something went wrong in the previous steps.

i) Install the OCS initialization script:

IRIX Platforms

```
cd $OCS_DIR/templates
cp ocs /etc/init.d/ocs
cd /etc/init.d
chmod +x ocs
./ocs config
```

If the operator GUI program, *xtape,* is to run on the local host, edit *ocs* to invoke it.

AIX Platforms

```
cd $OCS_DIR/templates
cp startocs /etc
cd /etc
chmod +x startocs
```

edit *rc.local* to execute *startocs*

If the operator GUI program, *xtape,* is to run on the local host, edit *startocs* to invoke it.

SunOS (Solaris) Platforms

```
cd $OCS_DIR/templates
cp ocs /etc/init.d/ocs
cd /etc/init.d
chmod +x ocs
./ocs config
```

If the operator GUI program, xtape, is to run on the local host, edit ocs to invoke it.

OSF1 Platforms

```
cd $OCS_DIR/templates
cp ocs /sbin/init.d/ocs
cd /sbin/init.d
chmod +x ocs
./ocs config
```

If the operator GUI program, xtape, is to run on the local host, edit ocs to invoke it.

7.0 Administration

7.1 Administration Overview

Most administrative tasks are done using the *xocs* and *ocs_group* GUI commands. A user other than root may use these commands after that user has been to the OCS admin group via the *ocs_group* command. The capabilities of these utilities are discussed in the following subsections.

7.2 Administration using *Xocs*

The main *xocs* window is used to display the current status of tape drives. Clicking on the “Find Matches” button will display only those tape drives that meet the specified criteria (e.g. only allocated taped drives on a particular host). The column headings are:

- Device — The logical name (external label) assigned to a tape drive.
- Hostname — The host name on which the corresponding device is located.
- Device Type — The physical type of the tape drive, e.g. an EXB-8200.
- Allocation — Whether the device is allocated to a user or unallocated.

- Username The name of the user to whom the device is allocated.
- UID The system UID corresponding to the username.
- Status Whether the drive is in working condition or believed to be broken.
- VSN The VSN of the last tape that was mount on the device.
- CPU Status The CPU status of the node the tape drive is on.

The default criteria is to display all tape drives on all hosts.

*X*ocs also has three pull down menus:

- View
- Action
- DB

These pull down menus are discussed in the following subsections, however only the “DB” menu is mostly required for OCS administration.

7.2.1 View

This menu allows access to the following views of the OCS database:

- Mount Requests
  Display all pending mount requests. The column headings should be self explanatory.
- Mount Request Log
  When a mount request has been satisfied (either successfully or as a failure), an entry is created in the OCS database MOUNTLOG. The entry will remain in the log until it is removed via *ocs_xfer*. This view displays the MOUNTLOG and provides additional selection criteria, e.g. display only mounts for a particular user. The column headings should be self-explanatory
- Device Files
  Display the device files and device file attributes corresponding to a tape drive selected from the main *xocs* window. The drive should be selected on the list of the drives on the main window (See “Selecting Drives for the XOCS” on page 29.).
- Tape Drive Stats Log
  When a user calls *ocs_report_stat()* or executes *ocs_report_stat*, an entry is created in the OCS database STATLOG. The entry will remain in the log until it is removed via *ocs_list_reports()* or *ocs_reports*. This view displays the STATLOG and provides additional selection criteria, e.g. display only mounts for a particular device. Refer to *ocs_list_reports()* in the “OCS Reference Guide” for an explanation of the column headings.

• Tape Drive Stats

Display the summary statistics for a set of the tape drives selected from the main *xocs* window. Refer to *ocs_list_stats()* in the “OCS Reference Guide” for an explanation of the column headings.The drive should be selected on the list of the drives on the main window (See “Selecting Drives for the XOCS” on page 29.).

• Cleaning History

This view displays the cleaning history of the tape drive. This menu has additional selection criteria on it.

7.2.2 Action

This menu allows actions to be taken on a set of tape drives selected (See “Selecting Drives for the XOCS” on page 29.) from the main *xocs* window. The actions in this group are primarily used for routine maintenance, rather than for administration. They are:

• Allocate

Allocate the selected tape drives.

• Deallocate

Deallocate the selected tape drives.

• Mark Broken

Mark the selected tape drives broken so that they will not be allocated until they have been replaced or until they have been set working.

• Mark Working

Set the selected taped drives back to working. This action does not reset the installation date nor other associated counters. It should **not** be used if a tape drive **was** physically replaced.

• Cleaned

Update the last cleaning date and zero the associated counters for the selected tape drives.

• Replaced

Set the selected taped drives back to working **and** reset the installation date and zero the associated counters. It should be used **only if** a tape drive **was** physically replaced.

• CPU Status UP

Set the selected taped drive host’s CPU status to “up”. If no drive is selected, the user will be prompted for the host name in a separate window.

• CPU Status DOWN

Set the selected taped drive host’s CPU status to “down”. If no drive is selected, the user will be prompted for the host name in a separate window.

7.2.3 DB

This menu provides the key administrative actions, namely:

• Hosts Info

This window shows list of all available hosts and current information in the OCS database about selected one. The host can be selected by pressing left mouse button on the list and then pressing “refresh” to update information on the window. The other way to see information about particular host is type the host name in the text field and the press “refresh”. If the host is found it is selected on the list and information is updated.

Additional actions:

a) “Add” new host that will **not** have a *taped* running on it.
   Typically this only used for ACPMAPS platforms where tape drive allocation, mount requests, etc. are done remotely.
   A host that **does** have a *taped* running on it, is added to the OCS database via the *ocs_terminst* or *ocs_install* commands.

b) Change existing information in the OCS database by editing one or more text field (all but “Host Name”) and pressing “Update” button.

c) “Delete” a host from the OCS database.
   All tape drives on the host must be already deleted. This actions does not delete the hosts from the system files.

The various entry are:

a) Hostname: Name of the host, e.g. cdfsga.

b) Host O/S Flavor: Type of host, e.g. IRIX, AIX, OSF1, SunOS, ACPMAPS. (Use SunOS for Solaris)

c) Host O/S Release: Current release of the OS system (like 5.3 for IRIX)

d) Host O/S Version:

e) Mail Address - All: This list will receive messages for administration changes such as when tape drives are marked working or broken for tape drives on that host. An example of a valid list is:
   farm-admin@fnsg01, farm-users-cdf@fnal)

f) Mail Address - Service Only:
   This list will be to be used to alert service personnel that a tape drive is broken, but will not send them other mail messages. For example, farm-service@fnal. The string can be “None” if there is no such service personnel or they are included in the “Mail Address - All” list. Default if no value is provided is “None”.

g) Host CPU Status
   CPU status of the host. Choices are “up” or “down”. Default if no value is provided is “up”

- Users Info

It shows the list of users and the current information about selected one (see “Hosts Info” above how to manipulate the list - it works the same way)

It adds a user login and uid into the OCS database via OCS installation or whenever a user first accesses OCS. If a user has subsequently been removed from the local or NIS password file, this action may be used to delete the user from the OCS database.

- Tape Drive Info

  Shows device information from the OCS database for the selected drive (See “Selecting Drives for the XOCS” on page 29.) and (optionally) shows the current information read from the drive.

  Additional actions:

  a) Add a tapedrive to the OCS database.
     The drive name host to be unique for the entire OCS database and the basename has to be unique for on host. The serial number and device type should be given but it can be overwritten later. All other information is set to default which should be later corrected using “update” option.

  b) Update the information from the drive or the serial number, device type if the getting statisticians from the drive failed.

  c) Delete a tapedrive from the OCS database.

The various entry are:

Host Name:	The name of the host that has the tape drive on it.
Device Name:	The logical name (external label) assigned to a tape drive. It is recommended, though not necessary, to have unique logical name for tape drives in a particular OCS database. If the names are not unique, users must specify the host as well as the tape drive name when they wish to use a particular tape drive.
Device Basename:	Since there are multiple device files for a single device, this string should be the common part of all device files for a particular device.
Vendor Id:	Vendor unique name (like Quantum or Exabyte)
Product Id:	Product ID (like DLT400 or EXB-850085QANXRC)
Controller:	Type of the controller (like SCSI)
Firmware:	Current firmware loaded to device
Serial Number:	The manufacturers serial number of the tape drive.
Device Type:	The type of tape drive such as EXB-8500 or DLT2000.

7.3 Administration using *Ocs_group*

Ocs_group is a GUI utility for administering and viewing OCS authorization and device groups. *Ocs_group* has 3 interrelated windows:

- OCS_GROUP
- Authorization Groups
- Device Groups

These windows are discussed in the following subsections.

7.3.1 OCS_GROUP

This window displays the elements that may be used to create either an authorization group or a device group. The elements are:

- "CPU Pool Display" Window
  The list of all hosts in the OCS database.

- "Tape Drive Pool" Display Window
  The list of all tape drives in the OCS database. The host that the tape drive is on is shown in parenthesis.

- "User Pool" Display Window
  The list of all users in the OCS database.

- "New Authorization Group" Insertion Field & "Insert" Button
  Adding text in the "New Authorization Group" field and pressing "Insert" will create a new authorization group by that name in the "Authorization Groups" window. The new authorization group automatically will become the "Current Group" selected on the "Authorization Groups" window. Refer to the discussion in section 7.3.2 for what an authorization group is.

- "New Device Group" Insertion Fields & "Insert" Button
  Adding text in the "New Device Group" field and pressing "Insert" will create a new device group by that name in the "Device Groups" window. The new device group automatically will become the "Current Group" selected on the "Device Groups" window. Refer to the discussion in section 7.3.3 for what a device group is.

- "Add to Authorization Group" Button
  Add the selected items on "CPU Pool", "Tape Drive Pool" and "User Pool" windows to the "Current Group" selected on the "Authorization Groups" window.

- "Add to Device Group" Button
  Add the selected items on "CPU Pool" and "Tape Drive Pool" windows to the "Current Group" selected on the "Device Groups" window.

- "Clear Selections" Button"
  Clear any selections made on the "CPU Pool", "Tape Drive Pool" and "User Pool" windows.

- “Status” Display Window

  The result of any *ocs_group* button press, not only for the “OCS_GROUP” window, but for the “Authorization Groups” and “Device Groups” window as well.

- QUIT Button

  Exit *ocs_group*.

7.3.2 Authorization Groups

An authorization group defines what users are allowed access to which tape drives. For example, suppose 2 authorization groups are created as:

group1	Consisting of a list of tape drives and users which should be granted access to those drives.
group2	Consisting of a list of hosts and users which should be granted access to all tape drives on those hosts.

Users would then be able to do any of the following allocation requests:

`ocs_allocate`

Allocate a tape drive on the local host searching through all authorization groups for drives that grants the user access.

`ocs_allocate -h any -t group1`

Allocate a tape drive that is explicitly listed in the authorization group “group1” provided the user is also a member of “group1”. The tape drive may be on any host.

`ocs_allocate -h localhost -t group1`

Allocate a tape drive that is explicitly listed in the authorization group “group1” which is on the local host provided the user is a member of “group1”.

`ocs_allocate -h group2`

Allocate a tape drive that is on any of the hosts explicitly listed in the authorization group “group2” provided the user is also a member of “group2”.

`ocs_allocate -h any -t group2`

Allocate a tape drive that is on any of the hosts explicitly listed in the authorization group “group2” provided the user is also a member of “group2”.

```
ocs_allocate -h group2 -t group1
```

Allocate a tape drive that is explicitly in the authorization group “group1” and for which the host is explicitly listed in the authorization group “group2”. The user must also be a member of “group1” and group2”.

There are 2 special authorization groups:

admin
: Users in this group have full OCS administration privileges. This includes being able to allocate, add, and delete any tape drive. Initially only “root” is in this group.

public
: If this group is created, all users in the “User Pool” are automatically added as members of the group. If a host is then inserted in the group, then all users will be granted access to all tape drives on that host. If a tape drive is inserted in the group, then all users will be granted access to that particular tape drive.

The elements of this window are:

- “Current Group” Selection Button
  Selecting an authorization group name on this pull down button will cause the hosts, tape drives and users that are members of the group to be displayed in the “CPUs” “Tape Drives” and “Users” windows discussed below.

- “CPUs” Display Window
  The hosts that were added to the authorization group by selecting hosts from the “CPU Pool” window and pressing the “Add to Authorization Group” button.

- “Tape Drives” Display Window
  The tape drives that were added to the authorization group by selecting tape drives from the Tape Drive Pool” window and pressing the “Add to Authorization Group” button.

- “Users” Display Window
  The users that were added to the authorization group by selecting users from the “User Pool” window and pressing the “Add to Authorization Group” button.

- “Remove Items” Button
  Pressing this button will remove items selected in the “CPU”, “Tape Drives” or “Users” window from the authorization group selected by the “Current Group” button.

- “Delete Group” Button
  Pressing this button will delete the authorization group selected on the “Current Group” button.

After creating authorization group, use the *ocs_tape* command with the **-h** and **-t** options to a verify that the authorization groups provide the desired access.

7.3.3 Device Groups

A device group is simply a logical grouping of tape drives. For a user to be able to access a tape drive from a particular device group, the user must also be authorized to access the drive via some authorization group. For example, the tape drives in an authorization group named "public" may be partitioned into 2 device groups named "test" and "production". Users would then be able to do any of the following allocation requests:

`ocs_allocate`
: Allocate a tape drive on the local host searching through all authorization groups for drives that grants the user access.

`ocs_allocate -h any -t public`
: Allocate a tape drive on any host and that is also in the authorization group "public".

`ocs_allocate -t test`
: Allocate a tape drive on the local host that is in the device group "test" and for which there is also an authorization group that grants the user access.

`ocs_allocate -h any -t production`
: Allocate a tape drive on any host that is in the device group "production" and for which there is also an authorization group that grants the user access.

Defining device groups is optional, e.g. many OCS installations have no device groups. Again, using the *ocs_tape* command with the **-h** and **-t** options is a good way to verify that the authorization and device groups provide the desired access.

The elements of the "Device Groups" window are:

- "Current Group" Selection Button

  Selecting a device group name on this pull down button will cause the hosts and tape drives that are members of the group to be displayed in the "CPUs" and "Tape Drives" windows discussed below.

- "CPUs" Display Window

  The hosts that were added to the device group by selecting hosts from the "CPU Pool" window and pressing the "Add to Device Group" button.

- "Tape Drives" Display Window

  The tape drives that were added to the device group by selecting tape drives from the "Tape Drive Pool" window and pressing the "Add to Device Group" button.

- “Remove Items” Button

  Pressing this button will remove items selected in the “CPU” or “Tape Drives” window from the device group selected by the “Current Group” button.

- “Delete Group” Button

  Pressing this button will delete the device group selected on the “Current Group” button.

7.4 Example Administrative Procedure

These notes describe how to use *xocs* and *ocs_group* initially to add a tape drive and authorize users to use the drive.

For the sake of this discussion, say:

- the name of the host you are installing OCS on is:

  `myhost`

- the email addresses of administrators and users who should be notified of broken, working, and replaced taped drives are:

  `myadmingrp@fnal, myusers@fnal`

- the email address of service personnel who need to be notified what a tape drive is mark broken is:

  `hdwfolks@fnal`

- the logical name that you want to assign your drive is:

  `mydev1`

- the manufacturer serial number of the drive is:

  `02087118`

- the drive is an:

  `Exabyte 8500`

- the base device file name of your drive is:

  `/dev/rmt/tps0d4`

Change these as needed. Now perform the following steps.

a) login to myhost as root

b) Setup the OCS environment:

C-shell Users	**Bourne/K-shell Users**
`setenv OCS_DIR /usr/products/ocs/v3_2`	`OCS_DIR= /usr/products/ocs/v3_2`
	`export OCS_DIR`
`cd OCS_DIR`	`cd OCS_DIR`
`source ups/setup.csh`	`. ups/setup.sh`

c) Enter:

```
xocs &
```

The first line of the *xocs* main window is a series of 4 pull down menus, namely:

```
View Action DB
```

The 3rd pull down menu, DB, has the following items:

```
Hosts Info
Users Info
Tape Drive Info
```

Select:

```
Tape Drive Info
```

from the DB pull down menu. This will bring up another pop-up menu. Ignore message about selecting the drive and fill in each field as follows:

a) Hostname: myhost

b) Device Name: mydrive

c) Device Basename: /dev/rmt/tps0d4

d) Serial Number: 02087118

e) Device Type: 8500

Click on the “Add” button and then turn on “Show Drive Statistics” and click on “Refresh” button to get all current information from OCS database as well as information directly read from the drive. To current all drive information click on “Update” button. If the serial number or device type is not given the user is asked if those read from the drive can be used to update database.

Click on “Dismiss” to exit the “Tape Drive Info” menu.

Select:

```
Host Info
```

from the DB pull down menu. This will bring up another pop-up menu. Fill in each field as follows:

a) Hostname: myhost

b) Host O/S Flavor: IRIX

c) Host O/S Release: (can be omitted)

d) Host O/S Version: (can be omitted)

e) Mail Address - All: myadmingrp@fnal, myusers@fnal (can be None)

f) Mail Address - Service Only: hdwfolks@fnal. (can be omitted)

g) Host CPU Status: set to “up” or “down”. Default is “up”.

Click on the “Add” button

Click on “Dismiss” to exit the “Host Info” menu.

Exit *xocs* by clicking on “Dismiss”

h) Enter:

```
ocs_tape
```

You should now see mydev1 listed.

i) Enter:

```
ocs_group
```

and three windows will pop up:

OCS_GROUP
Device Groups
Authorization Groups

Ignore or close the window called:

Device Groups

The “Current Group” in the “Authorization Groups” window should be “admin” and should only have “root” in the “Users” window.

If you want to make OCS administrative changes from other user ids:

a) Select other users by highlighting them in the “User Pool” box in the “OCS_GROUP window”.

b) Click on the “Add to Authorization Group” button

Now these users can make changes via *xocs* and *ocs_group*, although *ocs_install* is still limited to root.

- If you wish to allow all users who have access to “myhost” to be able to allocate the tape drive (this is the way many installations are done):

a) Enter “public” in the “New Authorization Group” box in the “OCS_GROUP” window.

b) Click on “Insert”

The “Current Group” in the “Authorization Groups” window should now be “public”

If you wish to give all users access to any tape drives that are on “myhost”:

a) highlight “myhost” in the “CP Pool” Window.

OR if you wish to give all users access to only specific tape drives that are on "myhost"

a) highlight instead "mydev1" in the "Tape Drive Pool" Window.

Click on "Add to Authorization Group"

Click on the "QUIT" button to exit *ocs_group*.

8.0 Operator Environment

The OCS GUI for tape mounting, *xtape*, may run on any host sharing an OCS database. These are some considerations for setting up the operator environment:

- Which operator consoles are appropriate to display an *xtape*? For operator convenience, there may be more than one *xtape* serving a single OCS database. For example, both on a console near to the tape drives being served and on a console centrally located.
- Which hosts should run the *xtape*(s)? It is most convenient that *xtape* run on the same host as the *dbserv*, since that host must be up for proper OCS client-server communication.
- Should the *xtape* come up automatically after the host is re-booted? This is highly recommended and is done for most Fermilab OCS installations.
- What remote host (if any) should be the central host for starting ALL *xtape*s remotely? For example, at the Fermilab FCC installation, an account on a central host has scripts to remotely start each *xtape*.
- What printer should be used to print mounts via *xtape*?
- Is there an appropriate database of tape vault numbers to be used with a particular *xtape*? If so, how and when will that database be updated.
- Should a tape drive be automatically marked broken if an operator fails a tape mount because a tape is stuck in the tape drive or the tape drive is otherwise broken? This is highly recommended.
- What file should be used to record all mounts and replies? This file is totally independent of the OCS database.
- What standard X-resources should be set for the *xtape?* Examples are initial screen position, font, colors.
- Are the logical names of the tape drives clearly labeled so that the operators can locate the device?

There is a sample start up script for and*.Xdefaults* file for *xtape* in *$OCS_DIR/templates*, but the "XTape" document is the ultimate resource for understanding *xtape* options and functionality.

9.0 Testing

$OCS_DIR/*examples* contain example programs to help users understand how OCS works. These programs are also very useful in testing an OCS installation since they can exercise almost 90% of OCS functionality. The programs were implemented in three different flavors; C language, Fortran, and Bourne shell. The executables are named *ocs_ctest*, *ocs_ftest*, and *ocs_btest* respectively. See $OCS_DIR/*examples/README* for further information.

10.0 Routine Maintenance

Two tasks should be scheduled to routinely execute:

- Flush OCS log files. OCS keeps a log of each mount request and each report of tape drive use (e.g. kilobytes written). These logs periodically need to be flushed or reduced (e.g keep the records for only the last 10 days). The *ocs_xfer* and *ocs_report* commands may be used to keep these logs down to a reasonable size.
- Backup the OCS database. The *ocs_backup* command creates an ASCII copy of the OCS database which can be used to re-populate an empty database via the OCS *sql_int* command.

There are samples of scripts to do these tasks in $OCS_DIR/*templates*.

11.0 Trouble Shooting Problems

This section provides general information and techniques for diagnosing problems. It is not, nor intended to be, all inclusive.

11.1 Error Messages

OCS commands and routines return an error code and message to the caller. Many problems return a unique code in the message identifying the severity of the problem. The OCS system daemons (*dbserv*, *taped*, and *devd*) log errors they detect or receive to the system error log (e.g. */usr/adm/SYSLOG*). To give a flavor of the type and severity of OCS errors, below are some examples of messages that may appear in the system error log:

- DEVD/I511: Starting (dev=horus, db=bastet, pn=1090000000)

  This is an example of any informational message that may be ignored. In the string "I511" the "I" indicates that the severity of the message, in this case "informational" and the "511" is unique within the OCS source code. Each OCS system daemon logs an informational message when it starts up.

- TAPED/W490: PMAP failure for farmx4:217843 - giving up on host

  As indicated by the "W" in the string "W490", this is an example of a warning message and should not be totally ignored. In this case, the *taped* on one host could not communicate to a *taped* on another host, farmx4. This would not be a problem if the host farmx4 is known to be down or has had no OCS activity since it was re-booted.

- TAPED/E564: No such file or directory, errno = 2
  TAPED/E565: Failed to chown device file

  A indicated by the "E" in the string "E564" and "E565", this are examples of errors that require some attention. In this case, either the character special devices do not exist for a taped drive or the OCS mapping between logical tape drive and character special devices is incorrect.

- TAPED/F530: Cannot open taped DBM file

  This is an example of a fatal error. In this case, the taped failed to start up because it could not open a file in its working directory, thus no OCS commands on that host will succeed until the problem is fixed. Again searching the OCS source code for the error number "530" will point to the exact line of code. The user in the above instance may have received an error message like:

- /E709: Failed to make RPC call to get DB host
  Client initialization failed

  which displays the unique error number "709". This is an error rather than a fatal problem because from the clients perspective, only that client process is affected.

11.2 Investigating RPC/INETD Problems

OCS relies heavily on *portmap* and *inetd*. These are some general tips to verify and ensure that the OCS daemons are interacting properly with *portmap* and *inetd*.

- The most straight forward way to verify that the OCS system daemons are properly installed and running is to enter:

  ```
  ocs_tape
  ```

  on each host sharing an OCS database. (It is best to run this as a user other than root to test verify the ownership and set effective user bit.) The RPC timeout for client commands such as *ocs_tape* is 2 minutes, so be patient and wait for the error message. If the *taped* not running properly, you will see a message similar to:

```
ocs_tape: Oct 11 09:18:32 ocs_tape/E709: Failed to make RPC call to get DB host
ocs_tape: Client initialization failed
```

  If the *dbserv* not running properly, you will see a message similar to:

```
ocs_tape: OCT 28 13:35:17 ocs_tape/E62: [767]SQL/68: Failed to make RPC call (sqlcode=-2)
ocs_tape: Client initialization failed
```

  If you get an error on the host where the *dbserv* is installed, fixed the problem on that host before worrying about errors from a host with just a *taped* installed.

  If you do not get an error on any of the hosts sharing a database, the *taped*(s) and *dbserv* are accessible and running.

- To check if *inetd* properly registered the *taped* and *dbserv*, enter:

  ```
  rpcinfo -p myhost
  ```

  There should be an entry for the *taped* such as:

  ```
  217843 1 tcp 1034 taped
  ```

  If the taped process is currently running, there will also be an entry for UDP such as:

  ```
  217843 1 udp 793 taped
  ```

  If the host has the *dbserv* installed on it, there should be an entry such as:

  ```
  211637 1 tcp 1033 dbserv
  ```

  If a TCP entry is missing:

  a) verify the appropriate entries are in */etc/rpc*.

  b) verify the appropriate entries are in *inetd.conf* and that the symbolic links are pointing to the correct executables and working directories. The symbolic links will be */usr/etc* in on AIX and IRIX platforms and in */usr/sbin* on SunOS (Solaris) and OSF1 platforms

c) check the system error log (e.g. */usr/adm/SYSLOG*) for errors from *inetd*, *taped* or *dbserv*.

- If the OCS system daemons are properly registered, you can easily check whether the TCP connection is accessible by sending a procedure 0 call to the program number with the appropriate protocol, e.g.:

```
rpcinfo -t myhost taped
```

or:

```
rpcinfo -t myhost 217843 1
```

or:

```
rpcinfo -n 1034 -t myhost 217843 1
```

If this fails for an OCS daemon, but succeeds for an system daemon (e.g. *ypbind*), most likely the OCS daemon needs to be restarted. Otherwise it is a more general system problem.

Checking the ports via *netstat* can also be useful.

- Do not start the *taped* or *dbserv* by hand or you will end up with multiple *dbserv*s and *taped*s running on the host. This is very bad because they will be writing the same working files. Instead, use a command like *ocs_tape* which will cause them to start via *inetd*.

- Do not restart *inetd* without first killing *taped* and *dbserv*. Otherwise, you will end up with multiple *dbserv*s and *taped*s running on the host. It is safest to kill *inetd*, *taped*, *dbserv* (in that order) and then start *inetd*.

- Do not explicitly delete any *portmap* entries, e.g.:

```
rpcinfo -d 1073741829 1
```

Instead, let the *portmap* clean up entries that were not properly unregistered (e.g. if a program was killed.) This can be done by sending a procedure 0 call to the program number with the appropriate protocol, e.g.:

```
rpcinfo -t myhost 1073741829 1
```

11.3 Rebuilding an OCS Database from a Backup

There are times where it may be desired to rebuild the database from an ASCII backup, rather than reinstalling OCS. For example, if:

- hosts or tape drives were extensively renamed
- a re-configuration of hosts requires splitting one database into multiple databases
- the database was destroyed or corrupted

To do this:

a) Set the environment variable DB_HOST to the name of the OCS database host.

b) Get a copy of an ASCII backup of the database and edit it as necessary, e.g change all occurrence of the host fndp1 to fnhppc.

c) Kill the *xtape, devd, taped*, and *dbserv* processes.

d) Move the old database directory (e.g. */usr/spool/ocs/DB*) and create a new, empty directory in the same place. (Note, if users programs are running, the system daemons may have been automatically restarted by *inetd* so that step C) may need to be repeated.

e) Create empty database tables either with *ocs_install* or with *ocs_terminst*, e.g:

```
ocs_terminst
do tables
quit
```

f) Populate the data base from the backup or edited copy of the backup, e.g.

```
sqlint < /usr/spool/ocs/dbbkup.102094
```

Verify the command succeeded by echoing $status or $? depending on whether or ar a C-shell or Bourne/K-shell.

g) Make sure all is working. Minimally, execute (as a user other than root):

```
ocs_tape -h any
```

and make sure all the tape drives are listed.

h) Restart the appropriate *xtape* processes.

i) Remove the old database directory.

11.4 Verifying Device File Names

When a tape drive is added via *xocs*, a relationship is created in the OCS database mapping a physical device file name to a logical tape drive name and selected attributes such as “no automatic rewind on close”. It is very important that this relationship is correct.

The following sequence of commands is one approach that can be used to check the relationship:

a) Enter:

```
ocs_devfile -t cdf102
```

The device file name with the default attributes were be printed on standard output, e.g.:

```
/dev/rmt/tps131d3nrnsv.8500
```

b) Using the device file name displayed in step A) enter:

```
ocs_logdev -f /dev/rmt/tps131d3nrnsv.8500
```

The attributes that OCS has associated with the device will be printed on standard output, e.g.:

```
Characteristics for /dev/rmt/tps131d3nrnsv.8500 (cdf102@cdfsgb):
density=8500, rewind=0, retension=-1, swap=0, fixed=0, compres-
sion=-1
```

c) Make sure that the device file name displayed in step A) exists on the host, e.g:

```
mt -t /dev/rmt/tps131d3nrnsv.8500 status
```

Should give valid results, e.g:

```
Controller: SCSI
Device: EXABYTE: EXB-85058SQANXR106M0
Status: 0x200
Drive type: 8mm(8500) cartridge
Media: Not READY
```

d) On some devices you can verify the serial number by entering:

```
ocs_devstat -t cdf102
```

which will return for example:

```
---------------------------------------------------------------------------
Collecting Tape Drive Statistic ------------------- Wed Feb  5 09:59:10 1997
---------------------------------------------------------------------------
Host Name             = fndaub
Device Name           = ols02
Device Filename       = /dev/rmt/tps4d3
Device Type           = EXB-8500
Controller            = SCSI
Vendor Id             = EXABYTE
Product Id            = EXB-850085QANXRC
Firmware Id           = 05E0
Serial Number         = 02517640
Number of Hours On    =         -1   In Motion =         -1
Count/KBytes    Read =          0       Write =         -1
Errors          Read =          0       Write =         -1
Comp Ratio      Read =         -1       Write =         -1
Compresion            = NO
Density               = 8500
Media Type            = 133
Block SizeBlock Total          = 4585536
Count Origin          = Exabyte_Extended_Sense
Remain Tape/KBytes    = 4585536
SCSI Sense Code       = 0
SCSI ASCQ             = 0
Track Retry           = -1
Stop/Start Count      = -1
SCSI Test Unit Ready  = 0
SCSI Sense Key        = NO_SENSE
---------------------------------------------------------------------------
Tape Not Present:  N | Write Prot:   N | Clean Bit:    N | Drive Cleaned: N
Beginning of Tape: Y | At File Mark: N | End of Media: Y | End of Tape:   N
Ready Bit:         Y | Power Fail:   N | SCSI ILI Bit: N |
---------------------------------------------------------------------------
```

e) If you know that a tape is loaded in the drive but the device is not ready, the problem is not with the OCS database but more likely that the external label on the tape drive is incorrect.

12.0 Appendixes

12.1 Selecting Drives for the XOCS

The main window shows list of available drives. The user can select one or more drives for some of the others windows and “action” calls.

- to select one drive click left mouse button
- to select a few successive lines form the list select fist line and then press “Shift” button and left mouse button
- to select next one or set lines press “Ctrl” button and left mouse button

If the window showing some one drive related information has “next” button and the is only one drive selected the next drive from the list on the main window is selected and the information is updated for this drive. When there two or more drives selected on the main window the “next” button shows information of the next selected drive and the list of the selected drives remain the same.