modsShutdown
MODS Server Shutdown Script
### Distribution List

<table>
<thead>
<tr>
<th>Recipient</th>
<th>Institution/Company</th>
<th>Number of Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Richard Pogge</td>
<td>OSU</td>
<td>1 (master file)</td>
</tr>
<tr>
<td>Kellee Summers</td>
<td>LBTO</td>
<td>1</td>
</tr>
</tbody>
</table>

### Document Change Record

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changes/Notes</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2014-10-16</td>
<td>Draft submitted for discussion</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>2014-11-07</td>
<td>Changes after initial site test</td>
<td></td>
</tr>
</tbody>
</table>
Contents

1 Introduction ........................................................................................................................................... 4
   1.1 Scope ............................................................................................................................................... 4
   1.2 Reference Documents .................................................................................................................... 4
   1.3 List of Abbreviations and Acronyms .............................................................................................. 5

2 Motivation: Graceful Shutdown of a MODS Host Computer .......................................................... 6

3 The modsShutdown Script .................................................................................................................... 6
   3.1 modsShutdown Command .............................................................................................................. 7
   3.2 modsShutdown Local Usage .......................................................................................................... 7
   3.3 modsShutdown Remote Usage ...................................................................................................... 7

4 MODS Software Shutdown Details ................................................................................................... 8
   4.1 Programs terminated with SIGKILL ............................................................................................... 8
   4.2 Programs terminated with SIGINT ............................................................................................... 8
   4.3 Programs with Custom Termination Scripts ............... Error! Bookmark not defined.
   4.4 Future Developments .................................................................................................................... 8
1 Introduction

1.1 Scope

This document describes the modsShutdown script that gracefully terminates MODS software services running on a MODS instrument control or data-acquisition host computer, and then power off that host. This script is designed to be executed either by root, a user with super user (“sudo”) privileges, or a privileged remote user as part of a facility-wide shutdown process.

The program is based on various engineering convenience programs, specifically “reaper”, created during MODS instrument lab testing, and it was developed in collaboration with LBTO software personnel, particularly Kellee Summers, as part of deployment of a observatory-wide system shutdown facility for the LBT.

1.2 Reference Documents

*LBT Documents describing the UPS and facility shutdown system???
1.3 List of Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>LBT</td>
<td>Large Binocular Telescope</td>
</tr>
<tr>
<td>LBTO</td>
<td>Large Binocular Telescope Observatory (operational arm)</td>
</tr>
<tr>
<td>MODS</td>
<td>Multi-Object Double Spectrograph</td>
</tr>
<tr>
<td>OSU</td>
<td>The Ohio State University</td>
</tr>
<tr>
<td>TBD</td>
<td>To Be Determined</td>
</tr>
<tr>
<td>UPS</td>
<td>Uninterruptable Power Supply</td>
</tr>
<tr>
<td>UTC</td>
<td>Coordinated Universal Time</td>
</tr>
</tbody>
</table>
2 Motivation: Graceful Shutdown of a MODS Host Computer

There are four (4) Linux host computers required to operate the MODS1 and MODS2 instruments in their current configuration:

- mods1 – MODS1 instrument server
- mods1data – MODS1 raw data server
- mods2 – MODS2 instrument server
- mods2data – MODS2 raw data server

In normal operation, each host computer runs a suite of programs that control the various instrument services: mechanisms control, environmental monitoring, telescope control system interface, user GUIs, etc.

When power is lost to the LBT facility, the instruments and their control computers are kept running on the LBT UPS system. If power is restored (either commercial power comes back or the mountain-top backup generator comes online), nothing needs to be done. However, in some circumstances it becomes necessary to automatically power-down the instruments in advance of loss of UPS power when it becomes clear that power will not be restored before the UPS battery capacity is exceeded.

Simply powering off the MODS host computers, or allowing them to power off abruptly when UPS power is cut-off can result in unintended consequences. To avoid this, we require a way to gracefully terminate running instrument software services and then power down the individual host computers in a single step. This shutdown procedures needs to take care of all termination and power-off for the user hiding the details behind a single directive.

In addition, with the increasing complexity of the LBT observatory facility, it is necessary to integrate this MODS-specific shutdown script into a larger facility-wide shutdown utility so that LBTO personnel can take down MODS along with all of the other systems either by a central command or triggered automatically by a UPS power watchdog system.

These variables report measurements of the glycol coolant entering and leaving the instrument’s glycol coolant distribution manifold.

3 The modsShutdown Script

The modsShutdown script is a Unix csh command script that performs the following functions:

1. Senses which MODS linux host computer it is running on (e.g., mods1).
2. Shuts down the MODS instrument services relevant to that host computer in the correct way for each program.
3. Shuts down and powers off that host computer.

Because the user invoking the modsShutdown script may not be the same user that originally started the various MODS instrument programs, and because host computer shutdown is a root-level command, modsScript needs to be executed with root or superuser (“sudo”) privileges.
3.1 modsShutdown Command

The modsShutdown script command usage is:

```bash
modsShutdown [all|soft|now] [sleepTime]
```

where:

- **all**  terminate all MODS software services the power off the host computer
- **soft**  terminate all MODS software services but leave the host computer running
- **now**   power off the host computer **immediately**, skipping MODS service shutdown

modsShutdown executed without arguments will print a usage message and exit immediately with no further actions.

The optional `sleepTime` parameter sets the time to sleep in seconds that the script pauses between terminating MODS software services and powering down the host computer. The default `sleepTime` is **15 seconds**, which allows sufficient time for graceful termination of all MODS instrument services before executing the host shutdown command, avoiding a potential race condition. Setting `sleepTime` on affects the `all` option, and it is ignored by the `soft` and `now` options.

The modsShutdown script is normally executed by root or by a user with super user privileges.

3.2 modsShutdown Local Usage

The modsShutdown script is installed in the `/usr/local/bin` directory on each of the relevant MODS host computers. It would normally be executed as follows:

```bash
sudo /usr/local/bin/modsShutdown all
```

If root executes modsShutdown, `sudo` may be omitted. If executed by a non-root user with `sudo`, you may be required to enter a password, unless the user has no-password `sudo` privileges as defined in the `/etc/sudoers` configuration.

If executed by an unprivileged user, only those MODS instrument services originally executed and therefore owned by that user will be terminated. Services started by other users will be left running, and Linux host shutdown and power-off commands will not be executed.

3.3 modsShutdown Remote Usage

To execute modsShutdown remotely, you need to ssh to the MODS host computer as a user with root or super user privileges via sudo. The usual invocation is via ssh, for example:

```bash
ssh -t userID@mods1 sudo /usr/local/bin/modsShutdown all
```

Here `userID` is a user with superuser privileges. At LBT, the remote `userID` will be “ups”, a special user with key-level ssh access to the MODS Linux hosts associated with the LBT facility UPS monitoring system tasked with performing a facility-wide shutdown of all critical systems in the event of imminent power failure upon exhaustion of UPS battery capacity.
4 MODS Software Shutdown Details

4.1 Programs terminated with SIGKILL

Most of the MODS instrument control programs can be terminated directly using the Unix SIGKILL signal, as they do not use system resources that require explicit release upon termination.

These programs are terminated by modsShutdown using the Linux “killall” command.

4.2 Programs terminated with SIGINT

The MODS mechanism control programs use TCP socket connections to networked serial port servers to provide command and control of the individual mechanism stepper motor controllers. If these programs are abruptly terminated with SIGKILL, they can leave “dangling sockets” that complicate reconnecting to the serial port servers upon instrument restart, requiring a hardware reboot of the port server units.

To address this problem, the MODS mechanism control programs are designed to trap the Unix SIGINT interrupt signal and pass control to handler functions that release the TCP socket connections and terminate the program.

These programs are terminated by modsShutdown using the Linux killall command with the “-s SIGINT” command-line option.

4.3 Future Developments

The current MODS data-taking system uses DOS PC computers to operate the science CCD controllers. These computers are not shutdown by the modsShutdown script, but are robust enough to be powered off in the most graceless fashion possible without bad consequences.

If we ever deploy next-generation Linux CCD control hosts using PCI-based fiber sequence interfaces, we will need to extend the number of Linux hosts to be powered down using some version of modsShutdown.