# ControlTower™

Console Management System

# User's Guide

Release 3.0

15-10104-00 Rev. C

## Limitation of Liability

Information contained in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty.

Aurora Technologies, Inc. makes no warranty, expressed or implied, with respect to this manual and any related items, their quality, performance, merchantability, or fitness for any particular use. It is solely the purchaser's responsibility to determine its suitability for any particular use.

In the interest of improving internal design, operational function, and/or reliability, Aurora Technologies, Inc. reserves the right to make changes to the products described in this document without notice. No guarantee, express or implied, is made that products of Aurora Technologies, Inc. will be compatible with future versions of the hardware systems and operating systems specified in this manual. YOU MUST READ THE SOFTWARE LICENSE AGREEMENT IN THE BACK OF THIS MANUAL AND RETURN THE PRODUCT UNOPENED IF YOU DO NOT AGREE TO BE BOUND BY ITS CONDITIONS.

## Trademarks

PREFACE

# *About this Book*

The *ControlTower User's Guide* describes how to install, configure and use ControlTower software. It also provides reference information.

This manual is organized as follows:

| | |
|---|---|
| CHAPTER 1, *Introduction to ControlTower™ Console Management System* | Provides introductory information about ControlTower. |
| CHAPTER 2, *Getting Started* | Describes how to prepare for installation. |
| CHAPTER 3, *Installing ControlTower Software* | Describes installation. |
| CHAPTER 4, *Security and Configuration Concepts* | Provides security and advanced configuration concepts. |
| CHAPTER 5, *Configuring ControlTower using the Command Line Interface* | Provides configuration instructions using the Command Line Interface. |
| CHAPTER 6, *Administering Managed Devices using the Command Line Interface* | Provides information on administering managed devices using the command line interface. |

| | |
|---|---|
| APPENDIX A, *Reference* | Lists man pages of ControlTower related User Commands. |
| APPENDIX B, *DEFAULT Configuration File* | Lists the default configuration file. |
| APPENDIX C *Glossary* | Presents frequently used terms and definitions. |
| APPENDIX D *An Example Configuration* | Shows examples of a LOCAL file, a group file, and device files. |
| APPENDIX E *Warranty* | Outlines Aurora Technologies Warranty and Maintenance information. |

## Who Should Use This Book

This book is a user's guide and reference for System Administrators who are using ControlTower to manage servers.

## Documentation Conventions

Unless otherwise noted in the text, this document uses the following symbolic conventions:

| | |
|---|---|
| screen display | Graphic text that appears on menus and dialog boxes appears in sans serif font. |
| **`literal input`** | Bold words or characters in formats and command descriptions represent commands that you must type literally. Literal values are in the **`bold typewriter`** font. |
| <***non-literal input***> | <***Bold, italic, bracketed***> words or characters in formats and command descriptions represent values that you must supply. Do not type the brackets. |
| `command` | ASCII text that the system displays and pathnames and commands in the text appear in `plain typewriter` font. |
| *emphasis* | *Italics* are used in the text for emphasis, titles, and variables. |
|  | This symbol marks cautionary notes about possible damage to your equipment or data. |
|  | This symbol marks procedures. |
| ❒ | This symbol marks the end of a chapter |

## Related Manuals

For more information, refer to the following manuals:

- Your Aurora Multiport Serial Controller User's Manual
- Your Sun Microsystems™ documentation
- On-line man pages

## Getting Help

If you need to reach us, you can contact us by

- The Web: **`www.auroratech.com`** for product literature, phone numbers and address.
- Phone service: 508.588.6110 or U.S. Toll Free 1.877.428.4277 Mon–Fri, 8:30AM–6:00PM Eastern Time. To expedite service, have your product serial number and your system information available.
- FAX: 781.290.5358. Attn: Customer Service and Support
- Email: **`support@auroratech.com`**

## Registration

To receive standard warranty coverage on your Aurora product, including 90 days of free technical support, you must print, fill out, and fax or mail back the Aurora Warranty Registration Card that is located in the back of this manual. Phone support can only be provided after product registration is complete. Extended Hardware and Software Support Agreements can be purchased to provide additional coverage.

Sending in this card also lets us keep you up-to-date on the complete line of Aurora Technologies' products.

If you have any questions or comments on your Aurora Technologies' product, contact our Customer Service and Support Department at **`support@auroratech.com`** or our Sales Department at **`sales@auroratech.com`**. ❐

# Contents

CHAPTER 1

# *Introduction to ControlTower™ Console Management System*

ControlTower™ Console Management System provides a reliable time and cost saving solution for monitoring and controlling multiple devices through an RS-232 interface from a central location or by remote access. It enables a single SPARC™ Solaris™ system to function as a common console (monitor and keyboard) for managed devices. The ControlTower System is available for either SBus or PCI bus multiport serial controller.

ControlTower Console Management System consists of both software and hardware components. ControlTower Software, consisting of Server and Viewer Client packages, resides on a Sun Solaris system. The ControlTower Host System provides a common console and maintains system logs for all managed devices. No additional software is required on the managed devices.

ControlTower Viewer Client software, in addition to residing on the Host, may reside on multiple systems that have network or modem capabilities to the ControlTower Host System. Any function that can be performed from a managed devices's keyboard and display can be performed remotely from a ControlTower Viewer Client, including monitoring log files, running diagnostics, and rebooting managed devices. In addition, you can use the

ControlTower Viewer Client to access one or more ControlTower Hosts via the network, enabling you to monitor and administer any number of systems in any number of locations from a single, central location, or from any number of locations you choose, as is shown in Figure 1.

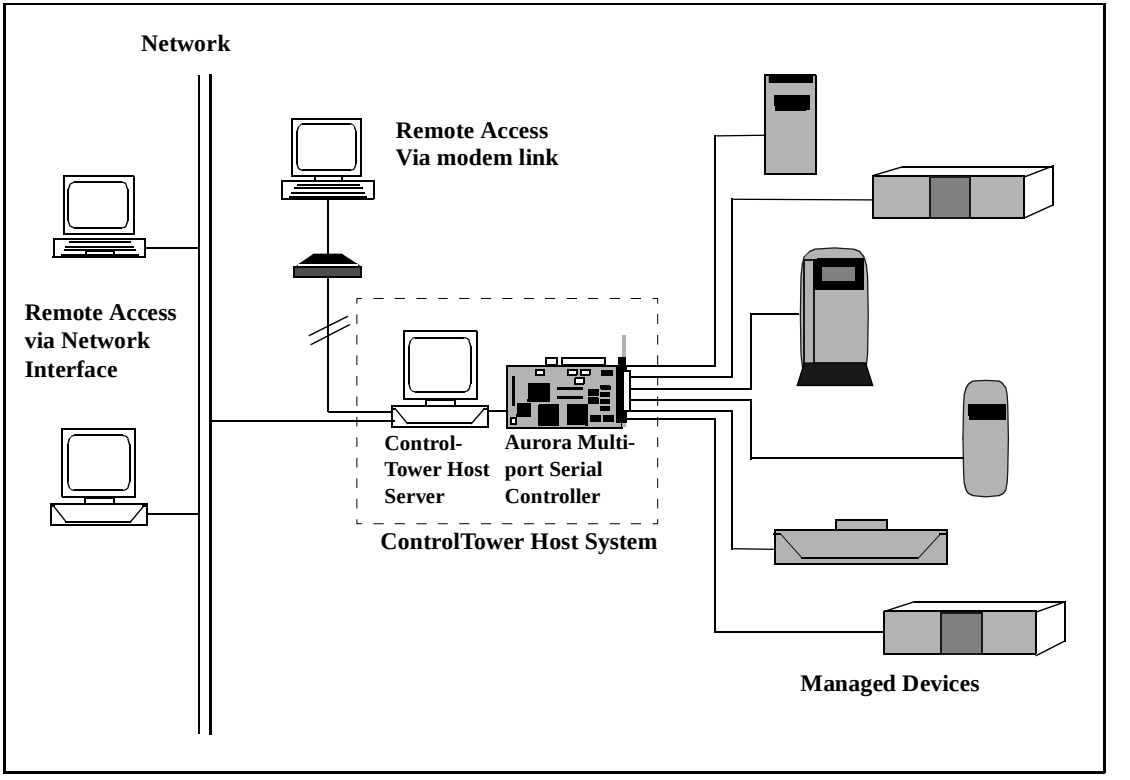


***FIGURE 1.*** *Console Management with ControlTower*

❒

# CHAPTER 2 *Getting Started*

This chapter describes steps you must take before installing ControlTower software. It tells how to select an appropriate ControlTower Host and how to properly identify managed devices. It also lists materials you need for installation.

## Before Installing

Before installing the ControlTower software you must complete the following tasks:

- Select the host machine
- Select appropriate systems as remote viewer clients
- Identify managed devices
- Verify that you have all necessary parts and materials, including cables and null modem adapters
- Install new hardware and drivers
- Obtain a license key
- Connect managed devices

Instructions on these tasks are found in this chapter. When these tasks are complete, you can proceed with installation.

## Select the Host Machine

You can use either an SBus or PCI Bus SPARC® machine as the ControlTower Host. The machine you choose must meet the following minimum requirements:

### PCI Systems:

| | |
|---|---|
| Host: | Sun™ Ultra™ 5 (or Ultra system of comparable performance) |
| Memory: | 64 MB RAM (minimum) |
| Operating System: | Solaris™ 2.6 or above |
| Serial Controller Hardware: | Aurora Aries Multiport™ 8000P or 16000P XP-7R™ |
| Disk Space: | 5 MB free in `/opt`; 50 MB free in `/var` |

### SBus Systems:

| | |
|---|---|
| Host: | SPARC 5 (or SPARC system of comparable performance) |
| Memory: | 64 MB RAM (minimum) |
| Operating System: | Solaris 2.6 or above |
| Serial Controller Hardware: | Aurora Nova Multiport™ 1600SE or LMS 1000 Series |
| Disk Space: | 5 MB free in `/opt`; 50 MB free in `/var` |

**Note:** The indicated memory requirements are based on the assumption that ControlTower software is run on a dedicated server. Running additional services on the host is NOT recommended.

## Important Host Selection and Set-up Considerations

Your ControlTower Host is a critical component of your console management solution. Aurora Technologies recommends the following steps to increase the security, availability and performance of your ControlTower Host:

- The ControlTower Host system should be a dedicated system. It should not be used by applications or users that might interfere with its console management role.
- The host machine should be attached to a UPS (uninterruptible power supply) of sufficient capacity to ensure that it will be up as long or longer than all managed devices.
- The host should not depend on NFS-mounted disks for its operation.
- The host should not depend on NIS (Yellow Pages) or NIS+ for its operation.
- Minimize the number of user accounts.
- Minimize host access, both physically and through the network (via filters/firewalls.)

## Break Signal Considerations

The supported Aurora multiport serial hardware has been thoroughly tested to verify that it does not transmit spurious break signals. Nevertheless, Aurora Technologies recommends that you take the following precautions:

- Attach all DB25 connectors with screws, and ensure that all RJ45 connectors are firmly latched.
- Avoid disconnecting and reconnecting the network connection on running systems.
- Avoid power-cycling the ControlTower Host at times when managed device operation is critical.
- If it is necessary to stop ControlTower processes, use `/opt/AURAcmgr/sbin/stop`.
- After connecting (or reconnecting) a managed device console port to the ControlTower Host, verify that the managed device

is operational by connecting using a Viewer Client. See the `cmgr(1)` man page for further information.

- Verify operation of *all* systems after power-cycling the ControlTower Host or reloading the Aurora Multiport Serial Driver.
- See the `kbd(1)` man page for information on how to enable/disable break on the console serial port.
- Attach your host machine to a UPS (uninterruptible power supply) of sufficient capacity to ensure that it will be up as long or longer than all managed devices.

## Select Appropriate Systems as Remote Viewer Clients

In order to install the Viewer Client software on a remote system, the remote system must be:

```
Solaris™ 2.6, 7 or 8
```

In addition, any machine connected to the ControlTower Host System through either the network or a modem can be used as a remote viewer client without installing additional software using telnet, rlogin, etc.

## Identify Managed Devices

ControlTower allows you to manage devices which have an RS-232 console port. Systems other than those running Sun Solaris must be tested for compatibility with ControlTower.

### Managed Devices Worksheet

Complete the Managed Device Worksheet (page 2 – 5) to help plan the types of devices you will be managing with ControlTower. Some examples are provided. Photocopy the worksheet for additional managed devices.

***TABLE 1.*** *Managed Devices Worksheet*

| Port # | Managed Device Name | Device Type | Console Port Connector Type | DCE or DTE |
|---|---|---|---|---|
| | | (e.g. modems, printers, workstations, servers) | (e.g. DB25, DB9, RJ45) | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

## Verify Materials

Before installing ControlTower, verify that you have all necessary materials. They are listed in the following hardware and software charts:

***TABLE 2.*** *Hardware Parts List*

| Qty. | Description |
|---|---|
| 1 | Dedicated server host—enter host ID#_____ |
| **var* | User's Manuals for Sun (or other system used as host) |
| 1 | Multiport Serial Controller Hardware |
| 1 | Serial Controller Card User's Manual with Device Driver CD-ROM |
| 1 | Driver Release Notes |
| 1 | Distribution cable or Breakout Box |
| 1 | Serial Test Plug |
| **var* | Adapters for Managed Devices (optional) |

***TABLE 3.*** *Software Parts List*

| Qty. | Description |
|---|---|
| 1 | ControlTower CD ROM—enter serial#_____ |
| 1 | ControlTower User's Guide |
| 1 | ControlTower Extended Support Agreement |

**Note:** **var*=Variable Quantity--depends on situation

## Install New Hardware and Drivers

Install new Aurora hardware on your chosen ControlTower Host system before you begin the ControlTower software installation. For information on installing the hardware, see the Aurora Technologies *User's Guide* for the hardware you are installing.

> **Note:** Third party serial hardware is not supported.

## Obtain License Key File

ControlTower requires a license key file for correct operation.

To obtain a license key, please contact Aurora Technologies Customer Service and Support. The product serial number and license information will be posted on the inside of the CD case. Contact information is as follows:

- The Web: **`www.auroratech.com`** for product literature, phone numbers and address.
- Phone service: From US exchanges 508.588.6110 Mon–Fri, 8:30AM–6:00PM Eastern Time. To expedite service, have your product serial number and your system information available.
- FAX: 508.588.0498; Attn: Customer Service and Support
- Email: **`support@auroratech.com`**

> **Note:** Telephone numbers occasionally change. Please see web site for current contact information.

When you contact Customer Service and Support you will need to provide:

- your ControlTower serial number.
- the `hostid` of the system on which you have installed ControlTower.

# Set Up Managed Devices

## Connect Managed Devices to Host

Whether or not you are installing new Aurora hardware, you will need to connect managed devices to the ControlTower Host via asynchronous null modem cables. You can also use a straight through cable and asynchronous null modem cable adapters.

Use one of the cable connections shown in Figure 2 and Figure 3 (or a straight cable with an asynchronous null modem adapter) to connect managed devices to the Aurora hardware. For additional pinouts, contact Customer Support at Aurora Technologies. (Contact information is found in the section "Obtain License Key File" on page 7).

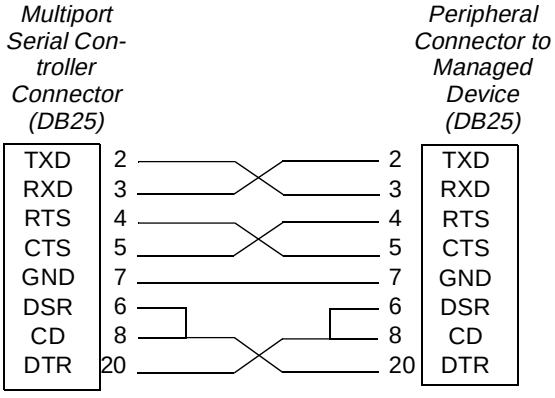


***FIGURE 2.*** *Asynchronous DB25-to-DB25 Null Modem Cable (XON/XOFF Handshaking)*

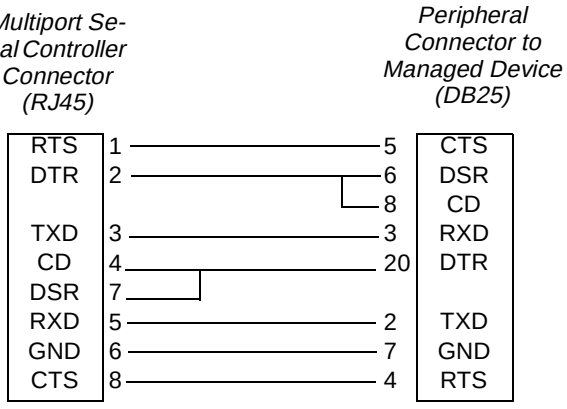


***FIGURE 3.*** *Asynchronous RJ45-to-DB25 Null Modem Adapter (Out-of Band Flow Control)*

## Preparing Managed Devices for Serial Communication

Follow the procedures below to prepare your managed devices for serial communication with the ControlTower Host system.

> **IMPORTANT:** When preparing managed devices for serial communication, you *must* disconnect the keyboard from the managed device as described in Step 2 below.

### Hardware.

1. Connect serial port “A” on the managed device to the ControlTower Host system serial port

2. Shut down and power off the managed device. Disconnect the keyboard. When a Sun machine boots and does not find a keyboard attached, it uses serial port “A” as the console port. Do NOT enable logins on ttya. The standard console entry in `/etc/inittab` is all that is needed.

3. Power on the managed device using the power switch. You should be able to watch the system boot using the Command Line Viewer Client.

**Software.** You do not need to install any ControlTower software on managed devices. ❒

# CHAPTER 3 *Installing ControlTower Software*

This chapter tells how to install ControlTower software. Prior to installation, you must complete the steps in CHAPTER 2, Getting Started.

Installation includes several tasks:

- Remove existing ControlTower Software
- Determine if Volume Manager is running
- Mount the CD-ROM
- Add packages
- Install Acrobat Reader
- Install the License Key
- Install ControlTower Software on remote systems

The above tasks are explained in this chapter.

## If a Previous Version of ControlTower Exists on your ControlTower Host

If there is a previous version of ControlTower installed on your ControlTower Host, you must remove it before proceeding to install your new ControlTower software.

> **Note:** With the single exception of the DEFAULT file, no existing ControlTower configuration or log file is affected (removed or overwritten) when you remove old ControlTower software packages. If you have made configuration changes to the DEFAULT file, you should preserve that information as described in "To remove existing ControlTower software:" below.

If you are unsure whether previous ControlTower software exists on your system, perform the following procedure to find out:

### *To check for existing ControlTower software:*

1. Log in as root:
   ```
   login: root
   Password: <root_password>
   ```
2. Check for existing ControlTower software by typing:
   ```
   system# pkginfo -l | grep AURA | more
   ```
   This command displays a list of Aurora packages currently installed on your system.

If either `AURAcmgr` or `AURAcmgrd` appears in the list, perform the following procedure to remove the old ControlTower software.

### *To remove existing ControlTower software:*

1. Log in as root:
   ```
   login: root
   Password: <root_password>
   ```

2. Change directory to the **AURAcmgr** config directory:

   ```
   system# cd /opt/AURAcmgr/config
   ```

3. Aurora always strongly recommends that you make all configuration changes for all devices in the LOCAL file, as this file is not overwritten when new ControlTower software is installed. If modifications have been made to the DEFAULT file, save the DEFAULT file to the LOCAL file to prevent the information in the DEFAULT file from being overwritten.

   ```
   system# cp DEFAULT LOCAL
   ```

   **Note:** It is not necessary to save configuration files other than DEFAULT because they will not be removed or overwritten.

4. To remove the package for ControlTower software, type:

   ```
   system# pkgrm AURAcmgr
   ```

5. Type ‘y’ in response to all resulting prompts.

6. To remove the ControlTower documentation package type:

   ```
   system# pkgrm AURAcmgrd
   ```

7. Type ‘y’ in response to all resulting prompts.

You may now proceed to install your new ControlTower software.

## Installing ControlTower Software

To install ControlTower software, you will need a host that is equipped with a CD-ROM drive.

If the host does not have a CD-ROM drive, you will need to install the software through another machine on the network that does. Contact Aurora Technologies Customer Service and Support for instructions on installing ControlTower software over a network.

## Files and Directories

ControlTower software is installed in the following directories:

| | |
|---|---|
| `/opt/AURAcmgr/bin` | User (Viewer) binaries |
| `/opt/AURAcmgr/sbin` | System (Server) binaries |
| `/var/log/cmgrlog` | Default directory for log files |
| `/opt/AURAcmgr/config` | Configuration files |
| `/opt/AURAcmgr/man` | Man pages |
| `/opt/AURAcmgr/doc` | Online (.pdf) version of this manual |

## Determine if Volume Manager is running

Before installing, you will need to determine whether your system is running Volume Manager (`vold`)

### *To Check if Volume Manager* (`vold`) *is running on your system*

1. Log in as root:

   ```
   login: root
   Password: <root_password>
   ```

2. Type:

   ```
   system# ps -elf | grep vold
   ```

   If you see a line with '`/usr/sbin/vold`' in the far right column, `vold` IS running on the system. Skip the section below and proceed to "Mounting the CD-ROM Using vold" on page 3 – 6.

   If you do not see a line with '`/usr/sbin/vold`' in the far right column, `vold` is NOT running on the system. Follow instructions below.

## Mounting the CD-ROM Manually

Use this procedure to mount the CD-ROM if `vold` is not running.

### *To Mount the ControlTower CD-ROM manually:*

3. Log in as root:
   ```
   login: root
   Password: <root_password>
   ```
4. Insert the CD-ROM into the CD-ROM drive.
5. Create a directory for the mount point:
   ```
   system# mkdir /mnt
   ```
   **Note:** If this directory already exists, you will see an error message that can be ignored.
6. Mount the CD-ROM manually. To do this, you must know the device pathname for the CD-ROM drive.

   On SCSI-connected CD-ROM drives, the path is typically `/dev/dsk/c0t6d0s0`.

   For IDE CD-ROM drives, the path is typically `/dev/dsk/c0t2d0s0`.

   If you do not know your CD-ROM pathname, see your System Administrator.

   Type:
   ```
   system# mount -r -F hsfs <device_pathname> /mnt
   ```
   where **<*device_pathname*>** is the device name of your CD-ROM drive.

   For example, for a SCSI CD-ROM device type:
   ```
   system# mount -r -F hsfs /dev/dsk/c0t6d0s0 /mnt
   ```
   See the `mount(1M)` man page for additional information.

If you are unable to mount the CD-ROM, ask your System Administrator for assistance.

## Mounting the CD-ROM Using `vold`

### *To Mount the ControlTower CD-ROM using* `vold`:

1. Log in as root:
   ```
   login: root
   Password: <root_password>
   ```
2. Insert the CD-ROM into the CD-ROM drive.
3. Type:
   ```
   system# volcheck
   ```
4. The CD-ROM should be mounted as:
   ```
   /cdrom/auracmgr_3_00
   ```
   This is for version 3.0 of ControlTower. For other versions, please consult our web site.

## Adding a Package

There are three packages that are included with ControlTower.

| | |
|---|---|
| `AURAcmgr` | the command line “viewer” package |
| `AURAcmgrs` | the server package; requires `AURAcmgr` |
| `AURAcmgrd` | the documentation package |

Select which packages to install. The ControlTower Host requires `AURAcmgr` and `AURAcmgrs` at a minimum. Remote viewers require `AURAcmgr`.

***To Add a Package:***

1. Change to the package directory using *one* of the following commands:

   •If you mounted the CD-ROM drive manually:

   ```
   system# cd /mnt/sparc-sos5
   ```

   •If *vold* mounted the CD-ROM drive:

   ```
   system# cd /cdrom/auracmgr_2_00/sparc-sos5
   ```

2. Start the installation script. You can either install all packages on the CD, or add selective packages.

   To install all packages, type:

   ```
   system# pkgadd -d .
   ```

   To add selective packages, type:

   ```
   system# pkgadd -d . <package_name>
   ```

   where <***package_name***> is the name of the package you are installing, for example, `AURAcmgrs` or `AURAcmgrd.`

   (See the `pkgadd(1M)` man page for additional information.)

3. Type `y` in response to the resulting prompts.

   When you see the message:

   ```
   Installation of <package_name> was successful
   ```

   (where <***package_name***> is a single installed package or the last of multiple packages) the software installation is complete.

4. To verify that the software has been properly installed on your host, type:

   ```
   system# pkginfo -l <package_name>
   ```

   You should see <***package_name***> in the output.

   If you prefer to see all packages on your system, type:

   ```
   system# pkginfo -l | more
   ```

   ControlTower packages begin with the prefix `AURA.`

5. Do not dismount or eject the CD-ROM at this time.

6. Read below to determine if you will need to install Acrobat Reader. If you will NOT be installing Acrobat Reader, go to the procedure "To remove the CD-ROM" on page 3 – 9.

## Installing the Acrobat Reader

An electronic version of this manual is available on the CD-ROM in the documentation package `AURAcmgrd`. After installation, it is found on the host computer in the directory `/opt/AURAcmgr/doc/auracmgr.pdf`. Adobe Acrobat Reader is necessary to read this pdf file. If you want to use the electronic version of this manual and do not already have the Acrobat Reader installed, use the following procedure to install it.

### *To install the Acrobat Reader*

1. Change to the Acrobat directory using *one* of the following commands:

   •If you mounted the CD-ROM drive manually:

   ```
   system# cd /mnt/AcroRead
   ```

   •If *vold* mounted the CD-ROM drive:

   ```
   system# cd /cdrom/auracmgr_3_00/AcroRead
   ```

2. Start the installation script:

   ```
   system# ./INSTALL
   ```

3. Reply to the resulting prompts. When you are prompted to enter the installation directory:

   ```
   Enter installation directory for Acrobat
   3.01 [/opt/Acrobat3]
   ```

   Enter <RETURN> to accept the default, or specify another directory.

   When you see the messages

   ```
   Installing platform independent files ... Done
   Installing platform dependent files ... Done
   ```

the software installation is complete.

***To remove the CD-ROM***

1. If you mounted the CD-ROM manually, dismount the CD-ROM drive. *(If the CD-ROM was mounted by* `vold`*, skip this step and go to Step 2.)*

   ```
   system# cd /
   system# umount /mnt
   ```

2. Eject the CD-ROM:

   ```
   system# eject cdrom
   ```

   Store the CD-ROM in the sleeve inside the back cover of this manual.

## Install License Key File

ControlTower software requires a license key file for correct operation. See "Obtain License Key File" on page 2 – 7.

The license key file is:

```
/opt/AURAcmgr/config/license.dat
```

ControlTower server software `AURAcmgrs` requires a license key containing the hostid of the machine on which it is installed or it will not start. Therefore, a license key will only work for one host machine. However, the command line Viewer Client package `AURAcmgr` doesn't require a key and may be installed on as many systems as desired.

The license key also contains the number of allowable ports associated with that license. If you need more ports, or would like to move ControlTower to a new machine, a new license key will be required.

## Installing ControlTower Software on Remote Systems

In order to install ControlTower Viewer Client software on a remote system, the remote system must be

`Solaris™ 2.6, 7 or 8`

If you want to use a Command Line Interface (CLI) on a remote Solaris system, you will need to install the ControlTower Viewer Client software package `AURAcmgr` on the remote system. To install the `AURAcmgr` package, see the instructions "Installing ControlTower Software" on page 3 – 3. ❒

CHAPTER 4

# *Security and Configuration Concepts*

This chapter presents important ControlTower security issues. It also provides information on how to configure:

- Encryption
- Log file management
- User-access to ControlTower servers
- Error logging
- Compatibility with previous versions

## Configuration Information

All ControlTower parameters are applied in a hierarchy depending on where the parameters are set. Parameters set in the LOCAL or DEFAULT files at the top level (`/opt/AURAcmgr/config`) apply to all managed devices unless overridden by settings at a lower level. Parameters set in a group configuration file (`/opt/AURAcmgr/config/<`***group***`>/<`***group***`>.grp`) override settings at the top level and device configuration files (`/opt/AURAcmgr/config/<`***device***`>.cfg` or `/opt/AURAcmgr/config/<`***group***`>/<`***device***`>.cfg`) override group and

top level settings. For more information refer to “Configuring Groups” on page 5 – 4.

Parameter settings only override parameters of the same name (except for the `device` and `stty` settings which are transparent). For instance, logdir set in a device file will override the logdir setting of the `LOCAL` file. However, there are parameters that interact with parameters of a different name and these each have their own hierarchy. Examples of this will be described as they are encountered.

See APPENDIX B, DEFAULT Configuration File, for a complete listing of the `DEFAULT` file. Also see the `config(4)` man page.

## Security Information

Since ControlTower sessions may involve the use of the root password, or may involve root access on a managed device or remote communications between the Viewer the ControlTower Host, you will want to keep security issues in mind when setting up and maintaining ControlTower.

### Remote Access Security

This version of ControlTower supports encryption of communications between a Viewer client running on a remote system and the ControlTower Host. This feature mitigates the security risks of transmitting sensitive data over TCP/IP networks.

Aurora recommends that you *always* enable encryption when using the Viewer remotely , unless your TCP/IP connection to the server is over a secure LAN environment. You can also use SSH to encrypt a remote connection to a Viewer running on the ControlTower Host.

To enable encryption for a managed system, include the line

```
DefaultEncrypt=128
```

in its configuration file. As an alternative, you can add this line to the LOCAL file to enable encryption for all managed systems by default.

The line

```
ForceEncrypt=true
```

will cause any requests for remote connections that do not support encryption at the default level to be refused by the ControlTower Host.

Also, if encryption is not enabled for a managed system, the Viewer client can enable it on connect (with V. 3.0 Hosts only) using the command

```
cmgr -f 128 <server_name>
```

## Other ControlTower Security Features

In addtion to remote access encryption, the following security features are provided by ControlTower:

- All remote connections over a network require entry of a password. If the `authuser` parameter is set, all remote users will share the password for the user specified in `authuser`. Alternatively, access can be managed on a per-user basis using the `authfile` parameter.
- All network connections are checked using IP address access control lists to permit or deny connections from specific hosts or entire networks (or net blocks). This is achieved with the `tcpallow` and `tcpdeny` parameters.

  The following parameters control remote access over a network or modem.

  `tcpenable:` enable use of TCP/IP connections

  `authuser:` name of user in password file to use for password verification

`authfile:` specifies the name of a file containing a list of authorized users and their permissions.

- The user specified in `authuser` must be a valid user, but the account need not have a usable shell (i.e., `/usr/bin/false`). This parameter is ignored if `authfile` is set.
- If you are using the `authfile` parameter, the parameter should contain the name of a file with a relative or absolute path. Each entry in the file specified by `authfile` should contain a valid account name. The `localauth` parameter must be set to true for the permissions in the `authfile` to take effect on the local host.

  *NOTE: Relative paths are relative to the location of the configuration file for the device to which that parameter is applied. For instance, if a device is under a group, but the authfile that applies to that device is set in the LOCAL file, the authfile file will need to reside under the group directory. If the same authfile applies to multiple groups, a copy of the file will need to reside under each group directory.*

  **IMPORTANT:** *If you enable remote access (`tcpenable=true`, and `authuser=`<**user_ with_ password**> or `authfile=` <**file_with_list_of_users**>, you should consider setting up network access control lists using `tcpallow` and `tcpdeny`.* See the `config(4)` man page.

## Log File Management

You can configure a number of aspects of log file management, including:

- Where log files are stored
- The contents of log files
- The maximum size of log files
- The number of saved log files
- How saved log files are compressed
- The log file timestamping frequency
- Log file protections

## Storage Directory for Log Files

By default, the ControlTower Host stores log files in the directory `/var/log/cmgr log`. Use the `logdir` parameter to specify a different storage directory for log files. If a device is a member of a group, the log file for that device will be placed under a group directory below the `logdir` directory.

This behavior can be overriden using the `logfile` parameter, which explicitly sets the pathname for the log output. Adding this parameter to the LOCAL file concatenates all log output to a single large file. Aurora recommends that, if you use the logfile parameter, you disable the logstamp feature and enable the lognameprepend feature, to make the resulting output more readable.

## Contents of the Log File

The ControlTower Host creates and populates a log file (<***device_name***>) for each managed device. All system console output is saved to this file.

By default, user input is not logged (`loginput=false`); user log messages are logged (`logmessages=true`).

> **CAUTION:** Logged user input *includes passwords*.

What is written to a log file can be controlled on a line-by-line basis using `logfilter`. A file specified by the `logfilter` parameter contains drop and keep commands that use regular expressions to determine which messages are written to the log file.

## Log File Rotation

The ControlTower Host stores old log files, up to the number specified in the `logmaxfiles` configuration parameter.

Every ten minutes, the `logcheck` program is run from the root `crontab` file to check the size of each managed device's active log file. If the log file size exceeds that specified in the `logmaxsize` configuration parameter, the active file is compressed, using the standard `compress` utility or other compression algorithm specified in the configuration file for the managed device. Refer to the `compress(1)` man page. The log files are then *rotated*, with the following result:

- The newly compressed active logfile becomes <***system_name***>**`.1.Z`**
- The next most recently compressed logfile becomes <***system_name***>**`.2.Z`**
- ...and so on, up to the value of the `logmaxfiles` configuration parameter.

## Log File Compression

By default, log files are compressed using the Sun Microsytems supplied `compress` program. If you wish to use an alternate compression program, such as GNU zip (gzip), there will be three parameters that need to be changed: `logcompress`, `logcompressopt`, and `logcompressext`. For example, to use `gzip` instead of the default compress program set:

```
logcompress=/usr/local/bin/gzip
logcompressopt=-f
logompressext=.gz
```

The `logcompressext` parameter must be set to the extension that the logfiles will have when compressed. Otherwise, the log files won't be rotated.

## Log File Timestamping

Time stamps are periodically placed into the log files. The timestamp frequency is determined by the `logstamp` parameter. The default setting is 60 minutes. Periodic timestamp format is determined by the `logstampformat` parameter. For more informa-

tion about `logstampformat` see "logstampformat" on page 5 – 9.

In addition, each log entry begins with a timestamp. This can be turned off by setting `loglinestamp` to null (`loglinestamp=`). You can control the appearance of the time stamp by changing the format characters. See also *logstamp*, *logstampformat* and *loglinestamp* on page 5 – 9.

## Log File Protections

The default protection mode, owner, and group for a managed device's logfiles are as follows:

```
logmode=u=rw
logowner=root
loggroup=sys
```

Values you can specify for these are as follows:

Protection Mode
The value specified for `logmode` can be expressed either as an octal number (e.g., 600), or as a comma-separated sequence of absolute modes strings (e.g., `u=r,o=rw`). See the `chmod` (1) man page for a detailed description of these possible values.

Owner
The value specified for `logowner` can be expressed as a decimal user-id, or as a username from the password database.

Group
The value of `loggroup` can be expressed as a decimal group-id, or as a group name from the groups database.

**CAUTION:** Log files may contain sensitive system information (including passwords). You should carefully consider to whom you make them accessible. Through the use of Regular Expressions in the log filter, sensitive

information may be removed. Refer to log filtering on page 4 – 8.

## Disk Space for Log Files

Your disk space requirements for log files are affected by two factors:

- Managed device activity (i.e., how much data is output to the log files in a given period)
- Your logging requirement (i.e., the length of time for which you want to save logged data)

You can roughly calculate your disk space usage using the following formula:

*<#_of_managed_devices >* `* (logmaxsize + logmaxfiles * logmaxsize/2)`

For 128 managed devices, using the default values (`logmaxfiles=7, logmaxsize=50000`), and assuming a compression ratio of 1/2 (`logmaxsize/2`), this calculates to ~27 MB.

At the worst-case output rate of 960 char/sec, however, this space will suffice for only ~6 minutes of logfile storage, making your worst-case log file storage requirement for 1 hour's worth of data 270 MB. You may want to modify the defaults, depending on your log file activity, your available disk space, and the length of time for which you require log file coverage.

See *logcompress* on page 5 – 11 and the `config(4)` man page for more information on compressing files.

## Log Filtering

Log filtering selects which lines of information are written to the log file based on sequences of characters found within the line using Regular Expression matching. Log filtering is configured using the `logfilter` parameter to specify a file name and populating a file of that name with Regular Expression commands that will 'drop' (or 'keep') lines that would otherwise be written

(or not) to the log file. This can conserve disk space. See *logcompress* on page 5 – 11 and the `config(4)` man page for information about other ways of conserving disk space.

## Authorization Parameters

You can configure a number of aspects of user-access to the ControlTower Host, including:

- Whether and how local-domain access to a ControlTower Host is permitted
- Whether, and from what hosts, remote TCP/IP access to the ControlTower Host is permitted
- The usernames to use for remote access to a ControlTower Host
- Setting permissions for access to managed devices

### Local Access Control

UNIX-domain access is used for local Command Line Viewer Client access. This is the case when the **`CONSOLE_SERVERS`** environment variable is not set, and the Command Line Viewer is started without specifying a remote server (<***device_name***>@<***server_name***>).

Use the `localenable` parameter to permit or deny local-domain access to ControlTower Hosts. By default, local-domain access is permitted (`localenable=true`).

Use the `localauth` parameter to specify whether users must enter a password for local access to ControlTower Hosts. By default, password entry is not required (`localauth=false`).

### Remote (TCP/IP) Access Control

Use the `tcpenable` parameter to permit or deny access via TCP/IP to ControlTower Hosts. By default, TCP/IP access is permitted (`tcpenable=true`).

TCP/IP access can be controlled on a system-by-system basis by entering the IP addresses of servers into `tcpallow` and `tcpdeny` in a comma-delimited list.

### Username for Remote Access

- All remote connections over a network require entry of a password. This password may be the same for all devices managed by a ControlTower Host using the `authfile` parameter. Alternatively, authorization can be managed through the `authfile` parameter. This is the recommended method for authorization since it gives much better control over access.
- All network connections are checked using IP address access control lists to permit or deny connections from specific hosts or entire networks (or net blocks).
- The configuration parameters for TCP/IP network connections are the following:

  `tcpenable`: enable use of TCP/IP connections

  `authuser`: name of the user in the password file to use for password verification by all users. Ignored if `authfile is` set.

  `authfile`: name of a file containing a list of authorized users and their permissions.
- To use network client access using `authuser`, the user specified in the `authuser` parameter must be a valid account.
- To use network client access using `authfile`, set the parameter to the name of a file containing users and their permissions. This file name can contain an absolute path, or if a path is not given, the file is expected to be in the directory containing the .cfg file specifying this file name. Each entry in the `authfile` file should have a valid account name. Valid account names are specified by your system administrator.

The account need not have a usable shell. (i.e., use `/usr/bin/false`.)

**IMPORTANT:** *If you enable network access (`tcpenable=true, authuser=<user_with_password>`, or `authfile=<file_with_list_of_users>`, you should consider setting up network access control lists using `tcpallow` and `tcpdeny`.* See the `config(4)` man page.

### User Permissions to Access Managed Devices

Using the `authfile` parameter is the recommended method for controlling access to managed devices. Using `authfile`, security can be configured so that each user has different permissions for each managed device and different sets of users can have access to different devices or sets of devices (groups).

## Error Logging

All messages output by the ControlTower Host program that runs for each device (`conserv`) are sent to `syslog` tagged with the `daemon` facility code (except for security-related messages, which are tagged with the `auth` facility code.) See the `syslogd(1M)` man page for information on configuring the syslog daemon.

If you are having difficulty using `syslog` to debug problems, contact Customer Service and Support. See “Getting Help” on page vi.

## Compatibility With Previous Versions of ControlTower

ControlTower 3.0 is backward compatible with all previous versions as follows:

- Version 3.0 Viewers can connect to Version 1.0 servers, but encryption cannot be enabled for these connections.
- Version 1.0 Viewers can connect to Version 3.0 servers, but authorization will use “authuser”, and encryption cannot be enabled.
- If the ForceEncrypt parameter is enabled, pre-Version 3.0 Viewers will refuse remote connections.
- Configuration files from previous releases are migrated to Version 3.0 automatically. Note however that configuration changes made to the DEFAULT file will be lost unless it is copied to the LOCAL file before Version 3.0 is installed.

CHAPTER 5

# *Configuring ControlTower using the Command Line Interface*

This chapter tells how to configure ControlTower software using the Command Line Interface (CLI). This includes how to set up configuration files for each managed device.

This chapter assumes strong knowledge of UNIX™ commands. If any listed commands are unknown or their usage is unclear, please see the man page for the command (`man` <***command***>).

**IMPORTANT:** *Use of ControlTower software involves important security issues. Be sure to read CHAPTER 4, Security and Configuration Concepts.*

## Configuration Tasks

Configuration of the ControlTower software consists of the following tasks:

- Set up managed device configuration files
- Set up the environment

## Set Up Managed Device Configuration Files

The default configuration for all devices managed by a ControlTower server is specified in the `DEFAULT` configuration file in `/opt/AURAcmgr/config`.

In addition, each device is represented by a configuration file in the format <***managed_device_name***>`.cfg`. This configuration file can override the configuration specified in the `DEFAULT` file.

> **Note:** If is strongly recommended that you make all modifications to parameters affecting all devices in the `LOCAL` file. The `DEFAULT` file is removed when an old version of ControlTower is removed and the `LOCAL` file is not.

The name you give this configuration file is the name by which the managed device will be known to ControlTower. It is recommended that the configuration file for the managed device have the same name as the managed device. Configuration file names:

- may be from 1 to 64 characters long
- may be the same as the network name, but are not required to be
- must have the extension "`.cfg`" (or "`.grp`" for group configuration files).
- must reside in `/opt/AURAcmgr/config` or a group directory directly under this directory.
- may not begin with a period "."

The configuration file for a managed device must contain, at a minimum, the console server device pathname specifying the server port to which the managed device console port has been connected. For example:

```
device=/dev/cua/1
```

> **Note:** To avoid connection problems, the "caller" device should always be used. All caller devices are in the `/dev/cua` subdirectory.

For a managed device to belong to a group, its configuration file must be located in the group subdirectory under `/opt/AURAc-mgr/config`. The subdirectory must contain a".`grp`" file with the same name as the subdirectory. The ".`grp`" file may be empty or contain parameters that will be applied to all devices in the group. The group file may not contain the device parameter.

You must create a configuration file for each managed device.

## Creating a Configuration File for a Managed Device

Perform the following procedure to create a minimal configuration file for a managed device:

***To create a configuration file***

***Log in as root (or use `su`):***

```
login: root
Password: <root_password>
```

3. cd to /opt/AURAcmgr/config:

   ```
   system# cd /opt/AURAcmgr/config
   ```

4. Using the text editor of your choice (`vi` is shown here), create a file having the name by which you wish this managed device to be known:

   ```
   system# vi <managed_device_name>.cfg
   ```

   The file must have a `.cfg` extension.

5. Insert into the file the line `device=/dev/cua/<`*port_number*`>` where <*port_number*> is the port to which this device has been attached. For example:

   ```
   device=/dev/cua/23
   ```

   for a managed device connected to serial port 23.

When you have created configuration files for all managed devices connected to the server, you are ready to start ControlTower.

## Configuration File Hierarchies and Precedence

Configuration file hierarchies are illustrated in Figure 4. The leaf nodes override anything above. For example, Group overrides `LOCAL`, and `LOCAL` overrides `DEFAULT`, but a configuration file for a managed device ovderrides all of these for that device.

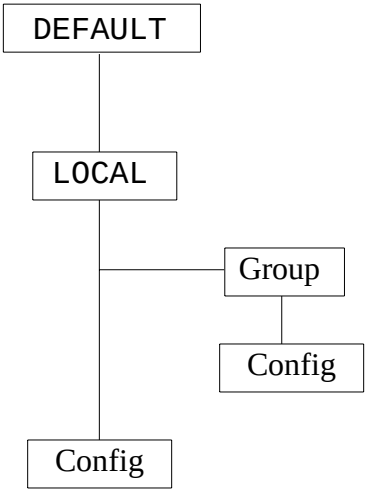


***FIGURE 4.*** *Configuration File Hierarchies*

### Configuring Groups

You can associate groups of managed devices using a subdirectory. Each subdirectory must have a file with the same name as the subdirectory and the extension `.grp`. This file contains the group configuration parameters. All devices that have configuration files within this directory will have the group configuration

file parameters applied to them unless these parameters are set in the devices' individual configuration files.

To avoid confusion, it is recommended that configuration file names be unique across groups. The configuration file name cannot be the same as the group file name.

## Creating Logins For Remote Users

You may want to perform one or more of the following tasks to set up the environment on the ControlTower server system:

If you intend to allow remote access to this ControlTower Host system, you must create user accounts for all users who are authorized to use ControlTower remotely. If you want separate logins for each user having access to the ControlTower Host system remotely, use the `authfile` parameter and create a separate login for each user listed in the `authfile` file. See "Username for Remote Access" on page 4 – 10. If you want to associate only one user account with any or all managed devices, set the `authuser` parameter to a user name and create a user account with the name specified. The default `authuser` name defined in the `DEFAULT` configuration file is "auracmgr". You can redefine this for all managed devices in the `LOCAL` file, or for an individual managed device in its <***managed_device_name***>.cfg file.

To create the "auracmgr" user:

```
system# useradd -s /usr/bin/true [-u <user_id>] auracmgr
system# passwd auracmgr
system# New Password: <auracmgr_password>
system# Re-enter New Password: <auracmgr_password>
```

## Starting the ControlTower Server Software

**Note:** Before starting the server, you must complete the intallation tasks described in Chapter 3.

### *To start the ControlTower Server*

To start the ControlTower server, type:

```
system# /opt/AURAcmgr/sbin/start
```

When the ControlTower server software starts up, it reads the managed device configuration files, starts up a process for each managed device, and connects to the console port of the managed device.

> **Note:** At start-up, the ControlTower server software stops existing ControlTower processes.

The ControlTower server is now in operation. You can use the Viewer Client to monitor and administer its managed devices.

### *To start ControlTower Server for an additional Managed Device*

If you have added a managed device, you can start a process to manage that system without restarting the ControlTower server software.

To start ControlTower server software for a single system, execute the `start` command and specify the new system name:

```
system# /opt/AURAcmgr/sbin/start <managed_device_name>
```

> **Note:** During installation, the ControlTower server start-up script is placed in `/etc/init.d` so that subsequent reboots of the system will automatically restart ControlTower.

## Stopping the ControlTower Server Software

### *To stop all ControlTower Server Software processes*

To stop the ControlTower server, execute the following command:

```
system# /opt/AURAcmgr/sbin/stop
```

### *To stop ControlTower Server Software for a Single Managed Device*

If you want to remove or rename a managed device, you can stop the process for that managed device without stopping the entire ControlTower server.

To stop the ControlTower server for a single managed device, execute the `stop` command and specify the managed device:

```
system# /opt/AURAcmgr/sbin/stop <managed_device_name>
```

## Configuration Parameters and Defaults

The following are the configuration parameters for ControlTower:

### exclusive

`exclusive` gives ControlTower sole access to a given port. The default is `true`. If this parameter is set to `false`, other programs can open this port. This is NOT recommended.

### uulock

`uulock` sets up a uucp-compatible lock file so that other programs do not use the port to send data to another system. The default is `true`.

### stty

`stty` controls serial port parameters. The default value is **`9600 cs8 -crtscts -cstopb -parenb -parext -parodd -ixoff -ixon istrip`**. Permissions for `stty` are set in the configuration file with `ttychanges`. See the `stty` man page for `stty` options and other information.

### ttychanges

`ttychanges` allows Viewer Client programs to change tty line parameters. The default is `true`.

### breakstring

`breakstring` allows you to configure what is sent instead of a break signal. If the `breakstring` parameter is not set, the break action will send a break signal to the managed device. If the `breakstring` parameter has been configured, the specified text will be sent. The default is null (`breakstring=`). If set, to unset this parameter in a configuration file at a lower level, set it to `*novalue*`. `breakstring` may contain backslash-escaped characters: \r \n \t \ooo (one or more octal digits) \xXX (two hex digits).

### logdir

`logdir` allows you to specify a directory to which log files will be written. The default is `/var/log/cmgrlog.` The value must be expressed as an absolute path. If the managed device is a member of a group, the device log file will be created in a subdirectory with the same name as the group.

### logfile

`logfile` allows you to explicitly specify the file to which log output will be written. `logfile` defaults to `logdir/<servername>`, but can be customized to a pathname for each server individually or all servers combined. If multiple server outputs are combined, it is recommended that you disable `logstamp`.

### lognameprepend

If enabled, `lognameprepend` prepends the server name to all logs made. Useful for combining several server log outputs to one file.

### loginput

If `loginput` is set to `true`, all text that is entered into the Viewer Client will be written to the log file, including passwords. The default is `false`.

### logmessages

`logmessages` controls whether messages generated by ControlTower are written to the log file. The default is `true`.

### logstamp

`logstamp` inserts a line containing a time stamp into the log file at regular intervals which you determine. Valid intervals are 10, 20, 30, or 60 minutes. A value of 0 means no logstamp is written. The default value is `60`.

### logstampformat

`logstampformat` contains the format of the time stamp that is inserted into the log file. See the `strftime(3C)` man page for valid format variables. The default is `**********%c**********`.

### loglinestamp

`loglinestamp` specifies that a time and date stamp will be written on each log line received from a managed device. If `loglinestamp` is null, no line-by-line timestamping will be performed. The default is `%c`. See the `strftime(3C)` man page for valid format variables.

### logmaxsize

`logmaxsize` is the maximum size of the log file in bytes. Every ten minutes, the log file size is checked, and, if it is over `log-`

`maxsize`, then the file is compressed and rotated. The default value is `50000`.

### logmaxfiles

`logmaxfiles` specifies the maximum number of log files. As each log file is compressed, it is assigned the suffix ".1" and all older log files have their suffixes incremented by one with the maximum suffix being `logmaxfiles`. The default value is `7`.

> **Note:** When setting this parameter, you will need to consider your overall file storage requirements for all of the log files created by ControlTower.

### logmode

`logmode` specifies the log file permissions mode. The default is `u=rw`, which means that only the owner has read/write access to the log files. The available values are `ugoa=rwx`. Different permissions can be set for different users (user, group, or other) by listing the different users and their permissions separated by a comma, for instance, `u=rwx,g=rw,o=r`. See the `chmod` (1) man page for more information.

### logowner

`logowner` specifies the owner of the log files. This would be the "u" in the description of `logmode`. The default value is `root`. Users are listed in `/etc/passwd`.

### loggroup

`loggroup` specifies the group to which the owner of the log files belongs. This would be the "g" in the description of `logmode`. The default value is `sys`. Defined groups are listed in `/etc/group`.

### logcompress

`logcompress` determines the compression utility used to compress log files. The default value is `/usr/bin/compress`.

### logcompressopt

`logcompressopt` specifies the command line options that will be used by the compression utility. The options must include one that will compress the log file regardless of whether any reduction in size will be realized. This is required so that a compressed file will have the extension of a compressed file, otherwise, the file will not be rotated. The default value is `-f`. See the `compress(1)` man page for command line options available for the default compression utility.

### logcompressext

`logcompressext` specifies the file extension given to a log file when it is compressed. It must be the extension that will be created by the compression utility or any options that specify an extension. The default is .Z.

### logfilter

`logfilter` specifies the name of a file that contains commands that drop or keep lines in the log file based on Regular Expressions. The name of the file may include an absolute or relative path. If relative, the path is relative to the directory in which the `logfilter` parameter is set. There is no default value. To unset `logfilter` in a configuration file at a lower level, set it to `*novalue*`. See the `regex(3)` man page for information on Regular Expressions.

The available commands that filter log file lines using Regular Expressions, are `keep` and `drop`.

The rules of log filtering are as follows:

- Each line of data from the managed device is tested in turn against each regular expression starting from the top of the list.
- When a match is found, processing stops. Therefore only the action of the first match is performed.
- If no match is found, the default action of `keep` occurs.

The following examples work collaboratively:

| Regular Expression | Application |
|---|---|
| keep /Mary had a little lamb/ | All lines containing the text "Mary had a little lamb" will be logged. |
| drop /lamb/ | All other lines containing the term "lamb" will be excluded from the log file. |
| # Comment | All text that begins with a # is a comment and is ignored. |

### Regular Expressions

Certain characters have special meaning in Regular Expressions. The most common are listed below along with their usage.

$ - the end of a string

^ - the beginning of a string, or NOT if it occurs at the beginning of (a) character(s) in square brackets

. - any single character other than a newline

+ - one or more occurrences of the preceeding character, e.g., a+

*- zero or more occurrences of the preceeding character, e.g., a*

() - delimits individual characters that form a string

[] - delimits a set of characters which must contain every character in the string for a match, '-' denotes a range of characters, e.g., [a-z]

\ - if before any of the special characters above, makes that character represent itself

Here are some examples:

```
^Mary[a-z ]*lamb$
```

matches any string with 'Mary' at the beginning, any number (including 0) of lower case letters and spaces, and 'lamb' at the end.

```
^[^0-9]+$
```

matches any string that doesn't have at least one digit.

```
(has).
```

matches any string with at least one occurrence of 'has' with at least one character after it.

It is important that `logfilter` files keep and drop the data they are expected to. To verify that they do, a syntax checker has been supplied that can run a `logfilter` file against sample input. The syntax checker is `filtertest` in `/opt/AURAcmgr/sbin`. The syntax is `/opt/AURAcmgr/sbin/filtertest` <***filterfile***> [<***inputfile***>].

If the <***filterfile***> contains a `drop` command and the first example of a Regular Expression (**`drop /^Mary[a-z ]*lamb$/`**) and the <***inputfile***> contains:

```
Mary had a little lamb.
Mary had a little lamb
Mary had 9 lambs
Mary has a little lamb
```

The output from `filtertest` will be:

```
KEEP: Mary had a little lamb.
DROP: Mary had a little lamb
KEEP: Mary had 9 lambs
DROP: Mary has a little lamb
SUMMARY: keep: 2, drop: 2
```

If the <***filterfile***> contains a `drop` command and the second example of a Regular Expression (**`drop /^[^0-9]+$/`**), the output from `filtertest` will be:

```
DROP: Mary had a little lamb.
DROP: Mary had a little lamb
KEEP: Mary had 9 lambs
DROP: Mary has a little lamb
SUMMARY: keep: 1, drop: 3
```

If the <***filterfile***> contains a `drop` command and the third example of a Regular Expression (**`drop /(has)./`**), the output from `filtertest` will be:

```
KEEP: Mary had a little lamb.
KEEP: Mary had a little lamb
KEEP: Mary had 9 lambs
DROP: Mary has a little lamb
SUMMARY: keep: 3, drop: 1
```

> **Note:** If you would prefer to use a delimiter other than '/', any character can be used as long as it begins and ends the Regular Expression.

### authuser

`authuser` specifies a user who is authorized to use a particular port. If `authuser` is used instead of `authfile`, there will only be one authorized user per port, so everyone who needs access to this port will use the same user name and password. The default is `auracmgr`.

> **Note:** Either `authuser` or `authfile` can be used for each device. Both cannot be used simultaneously.

### authfile

`authfile` is set to the name of a file that contains a comma-separated list of users and their permissions. The file name specified by `authfile` can include an absolute or relative path. If relative, the path is relative to the directory in which the `authfile` parameter is set. This parameter is unset by default. Once set, to unset `authfile` in a configuration file at a lower level, set it to `*novalue*`.

The permissions that can be assigned to users are listed below. Text in the parentheses are the parameters as seen in the `authfile` file.

- Attach (`attach`)--The user can acquire read/write permission for the managed device if there is currently no other user in read/write mode for that managed device. If another user is attached, and the user with `attach` permission tries to attach, the user will be attached in read-only mode.
- Force Attach (`fattach`)--A user who has this permission can acquire read/write permission to a device even if there is another user attached. Another user who is attached is forced into read-only mode.
- stty (`stty`)--The user has permission to set stty parameters for devices.
- Break (`break`)--the user has permission to send a break string to the managed device
- None (`none`)--The user has no authority to do anything.
- All (`all`)--the user has all of the above permissions.

Permissions can be combined with a plus (+) sign or subtracted with a minus (-) sign. The following is a sample `authfile` file with multiple users:

```
# This file contains users who have access
# to the devices in group
auracmgr attach+fattach+break
developer1 all-stty
# The following user will be allowed view-only
sessions
developer2 none
```

### tcpenable

`tcpenable` determines whether remote machines are allowed to connect to the server using TCP/IP over a network. The default value is `true`.

### tcpallow

`tcpallow` contains a list of machines that are allowed to connect to the server using TCP/IP over a network. If set, `tcpallow` will contain a comma-delimited list of IP addresses or host names, either of which can be followed by a mask. There is no default value. If set, to unset `tcpallow` in a configuration file at a lower level, set it to `*novalue*`.

### tcpdeny

`tcpdeny` contains a list of machines that are not allowed to connect to the server using TCP/IP over a network. The syntax is the same as for `tcpallow`. There is no default value. If set, to unset `tcpdeny` in a configuration file at a lower level, set it to `*novalue*`.

### defaultencrypt

`defaultencrypt` enables Twofish encryption over TCP/IP connections. The default is 0. Acceptable values are 0, 128 and 256. This only takes effect if the client on a local machine con-

nects to this server over TCP/IP using the CONSOLE_SERVERS environment variable.

### forceencrypt

`forceencrypt` causes all incoming TCP/IP connections to be dropped unless they accept the `defaultencrypt` or greater encryption level. This will also cause all v2.0 and v1.0 TCP client connections to be dropped.

### localenable (formerly UNIXenable)

`localenable` determines whether the command line viewer (`cmgr`) has access to the local ControlTower Server Host. The default is `true`.

### localauth (formerly UNIXauth)

`localauth` controls whether a password is required when using the command line viewer from the ControlTower Server Host. The default is `false`.

### disconnectidle

`disconnectidle` sets the maximum amount of time, in minutes, that the Viewer Client session is allowed to remain idle, regardless of whether it is in read-only or read/write mode. After this point, the Viewer Client is disconnected. If the parameter is set to 0, there will be no automatic disconnect. The default value is `0`.

### detachidle

`detachidle` sets the maximum amount of time, in minutes, that the Viewer Client is allowed to remain idle while in read/write mode. After this point, the Viewer Client is set to read-only mode. If the parameter is set to 0, there will be no forced shift into read-only mode. The default value is `0`. ❐

CHAPTER 6

# *Administering Managed Devices using the Command Line Interface*

## Administering Managed Devices using the Command Line Interface

This chapter tells how to administer and monitor managed devices using the Command Line Interface (CLI). You can administer and monitor managed devices through the Control-Tower Host and from remote Viewer Clients.

This chapter assumes knowledge of UNIX commands. If any listed commands are unknown or their usage is unclear, please see the man page for the command (`man` <***command***>).

## Setting the PATH Variable

To simplify entering ControlTower commands, you should add /opt/AURAcmgr/bin to your shell path as follows:

ksh or sh:

```
PATH=$PATH:/opt/AURAcmgr/bin
export PATH
MANPATH=$MANPATH:/opt/AURAcmgr/man
export MANPATH
```

csh or tcsh:

```
set path=($path /opt/AURAcmgr/bin)
setenv MANPATH $MANPATH:/opt/AURAcmgr/man
```

## Setting the CONSOLE_SERVERS Variable

Set the **CONSOLE_SERVERS** environment variable.

If you will be running ControlTower on multiple servers connected in a network and would like to use the Viewer Client to monitor devices managed by a different server, you may want to set **CONSOLE_SERVERS** to specify these servers. If **CONSOLE_SERVERS** contains a comma-separated list of ControlTower servers, the Viewer Client will have access to all of the listed servers .

Set **CONSOLE_SERVERS** as follows:

ksh or sh:

**CONSOLE_SERVERS=**<***server_system1***>,<***server_system2***>,...
**export CONSOLE_SERVERS**

csh or tcsh:

**setenv CONSOLE_SERVERS** <***server_system1***>, <***server_system2***>,...

The **CONSOLE_SERVERS** environment variable only exists for the server on which it was set. Each ControlTower server from which you wish to connect to devices on other other servers should have the **CONSOLE_SERVERS** environment variable set.

## About CLI Viewer Client

ControlTower CLI Viewer Client is a user interface to the ControlTower server software. After contacting the ControlTower server, the Viewer Client establishes an active session with the console port of *a single device* managed by that server. You must run one instance of Viewer Client software for each device you want to view.

You can run the ControlTower Viewer Client on the ControlTower Host, or on any Solaris for SPARC system connected to the host via network or modem.

There are two commands that start theViewer Client:

- `cmgr`, which runs in the current terminal window, or
- `xcmgr`, which opens and runs in a new `xterm`.

## CLI Viewer Client Operation

When you run the Viewer Client, you can specify in the command line the managed device you want to view, and the access mode.

### Specifying a Managed Device to View

To connect to a managed device, the Viewer Client needs to know:

- the name of the managed device, and
- the name of the ControlTower Host machine that manages that device, if it is on a remote server.

These are specified in the command line when you run the Viewer Client, as follows:

```
system# xcmgr <managed_device_name>[@<host_name>]
```

If you do not specify `@`<***host_name***>, the Viewer Client looks in the **`CONSOLE_SERVERS`** environment variable for a comma-

separated list of systems running ControlTower servers. If **CONSOLE_SERVERS** is not set, Viewer Client defaults to the local ControlTower server.

If you do not specify <***managed_device_name***>, the Viewer Client displays a list of devices accessible from the local server.

### Specifying the Access Mode

By default, the Viewer Client connects to the managed device in read-only mode. In read-only mode, you must enter an escape sequence to send input to the managed device console port.

You can, however, specify that the Viewer Client should `attach` to the managed device console port (i.e., read-write mode) with the ~a or ~A escape sequences. (For more information, see “Escape Sequences” on page 6 – 7.) In read/write mode, the Viewer Client window functions as a console terminal attached to the managed device. The users and permissions listed in the `authfile` file determine which escape sequences are available to which user. For more information on `authfile`, see “authfile” on page 5 – 15.

## Command Examples

These command examples show how to use the ControlTower Viewer Client to view managed devices. The **CONSOLE_SERVERS** environment variable determines how devices on local and remote servers are specified for viewing.

The following commands have the described effects when **CONSOLE_SERVERS** is not set.

- *List devices managed by the local server*

```
system# cmgr
cmgr: must have system name
hercules apollo ulysses agamemnon
[cmgr viewer exiting]
```

- *View a device managed by the local server in read-only mode:*

  `system# cmgr hercules`

  If security is being administered using `authuser`, the next line entered will be:

  `password:<`***authuser_password***`>`

  where <***authuser_password***> is the password for the user assigned to the authuser parameter. The default is 'auracmgr', but if authuser has been set to a different user, that user's password will be required. See the section "Setting the CONSOLE_SERVERS Variable" on page 6 – 2 for information on setting up this account.

  If security is being administered using `authfile`, the next lines entered will be:

  `username:<`***authfile_user***`>`

  `password:<`***authfile_user_password***`>`

  **Note:** The <***authfile_user***> specified will have the permissions assigned to them in the `authfile` file.

  The viewer is now attached to hercules in read-only mode. When you view a device in read-only mode, you cannot send input to that system. To send input, you must attach in read-write mode using an escape sequence. See "Escape Sequences" on page 6 – 7.

- *Attach in read-write mode to a managed device on the local server:*

  `system# cmgr -a hercules`

  Security will be as described above, however, if `authfile` is used for authorization, if the user who logs in doesn't have `attach` permission, the device will be attched in read-only mode.

  Similarly, if **-A** is used to force attach, if the user doesn't have `attach` and `fattach` permissions, any users already connected will not be diconnected and the device will be attached in read-only mode.

- *View a device managed by a remote server:*

  `system# cmgr <`***managed_device_name***`>@<`***remote_server_name***`>`

  If security is being administered using `authuser`, the next line entered will be:

  `password:<`***remote_server_authuser_password***`>`

where <***remote_server_authuser_password***> is the password for the user assigned to the `authuser` parameter on the remote server. The default is 'auracmgr', but if `authuser` has been set to a different user, that user's password will be required. See the section "Setting the CONSOLE_SERVERS Variable" on page 6 – 2 for information on setting up this account.

If security is being administered using `authfile`, the next lines entered will be:

`username:`<***remote_server_authfile_user***>

`password:`<***remote_server_authfile_user_password***>

**Note:** The <***remote_server_authfile_user***> specified will have the permissions assigned to them in the `authfile` file.

The following commands have the described effects when **`CONSOLE_SERVERS`** is set to both the local and remote servers. When this is the case, cmgr treats devices on a remote server in the same way it treats local devices.

- *List devices managed by the local server only:*

```
system# cmgr -l
cmgr: must have system name
hercules apollo ulysses agamemnon
[cmgr viewer exiting]
```

- *List devices managed by both local and remote servers:*

```
system# cmgr
cmgr: must have system name
agamemnon@server1 apollo@server1 dagwood@server2
dilbert@server2 hercules@server1 lucy@server2
ulysses@server1
[cmgr viewer exiting]
```

- *View a device managed by a remote server in read-only mode:*

`system# cmgr` <***managed_device_name***>

If security is being administered using `authuser`, the next line entered will be:

`password:`<***remote_server_authuser_password***>

where <***remote_server_authuser_password***> is the password for the user assigned to the `authuser` parameter on the remote server. The default is 'auracmgr', but if `authuser` has been set to a different user, that user's password will be required. See the section "Setting the CONSOLE_SERVERS Variable" on page 6 – 2 for information on setting up this account.

If security is being administered using `authfile`, the next lines entered will be:

`username:<`***remote_server_authfile_user***`>`

`password:<`***remote_server_authfile_user_password***`>`

**Note:** The <***remote_server_authfile_user***> specified will have the permissions assigned to them in the `authfile` file.

- *View either a local or remote device, specifying that output from the managed device to the terminal be 7 bits:*

  `system# cmgr -7 <`***managed_device_name***`>`

  Use of this option may be necessary if all 8 bits are processed by the server, but are not tolerated by the terminal.

- *View either a local or remote device, and specify a different escape character:*

  `system# cmgr -e % <`***managed_device_name***`>`

  This will cause all escape sequences to start with **%**.

- *View a remote device, using encrypted communications to the server:*

  `system# cmgr -f -128 <`***managed_device_name***`>`

## Escape Sequences

If the `authfile` parameter is set instead of `authuser`, the users and permissions listed in the authfile file determine which escape sequences can be used by which user. See “authfile” on page 5 – 15 for further information. All escape sequences begin with the tilde character ("~"), unless it was changed using the `-e` option in the command line or the escape setting of the `AURACMGR_OPTIONS` environment variable. The available escape sequences are as follows:

| | |
|---|---|
| ~. (tilde period) | Terminate the session. |
| ~CTRL/C | Terminate the session. |
| ~CTRL/Z | Suspend the `cmgr` program. The session is resumed with `fg`. |

| | |
|---|---|
| ~CTRL/L | Toggle local logging of the connection. |
| ~a | Attach: While in read-only mode, attach (read-write mode) to the managed device. Requires attach permission. |
| ~A | Force Attach: Force an attach (read-write mode) to the managed device. If someone else is attached (read-write), downgrade their connection to read-only. Requires attach and fattach permission. |
| ~d | Detach from the managed device, i.e., make the connection read-only. |
| ~q [ VWTA? ] | Query a server variable, as follows:<br>V Version<br>W who's connected to managed device<br>T tail of the log file<br>A all<br>? show options |
| ~s [ sT?] | Set a server variable as follows:<br>s set terminal stty parameters (Requires stty permission.)<br>T set tail length<br>? show options |
| ~# | Send a BREAK (if currently attached.) The user is prompted to confirm this action, which is aborted if not confirmed. Requires break permission. |
| ~? | Display help text on escape sequences |

❒

# APPENDIX A *Reference*

## User Commands cmgr(1)

## NAME

cmgr – Aurora ControlTower Console Manager viewer program

## SYNOPSIS

**cmgr** [ **-78aAlLPNv** ] [ **-d** *debuglevel* ] [ **-e** *c* ] [ **-f** *keylength* ] [ **-t** *taillen* ] [ **-o** *options* ] [ *system*[*@server*[*:port*]

**xcmgr** [ **-78aAlLPN** ] [ **-d** *debuglevel* ] [ **-e** *c* ] [ **-f** *keylength* ] [ **-p** *port* ] [ **-t** *taillen* ] [ **-o** *options* ] [ *system*[*@server*[*:port*] ]

## DESCRIPTION

**cmgr** establishes an interactive session with the console port of the named *system*. If no system name is specified, a list of possible systems will be printed.

If *@server* follows the system name, a TCP/IP connection will be made to the

named server, and the system name is used literally (no abbreviations accepted). The server name can be followed by a colon and a port number (or name from the **services**(4) file) to contact on the remote server.

If the **CONSOLE_SERVERS** environment variable contains a comma seperated list of servers (and the **–l** option is not given), the servers will be contacted in turn to retrieve the list of all possible system names. Each server name can optionally be followed by by a colon and a port number (or name from the **services**(4) file) to use to contact the remote server. If no system name is present on the command line, all the system names (and the server to which they are attached) will be sorted and printed. If a system name is present, it may be an unambiguous prefix. If the prefix is ambiguous, all matching system names will be printed.

If the **CONSOLE_SERVERS** environment variable is not set (or the **–l** option is given), the connection will be made locally, using Unix-domain sockets. This requires file access permission to the directory in which the sockets are located, and will not require a password to establish a connection. If no system name is present on the command line, all the system names will be sorted and printed. If a system name is present, it may be an unambiguous prefix. If the prefix is ambiguous, all matching system names will be printed.

**xcmgr** is a shell script which creates a new **xterm**(1) with a **cmgr**(1) program inside it.

## OPTIONS

**–7** Output only 7 bits of data to the terminal. This may be necessary if all 8 bits are being processed by the server, but are not tollerated by the user's terminal.

**–8** Neutralizes the effect of the **–7** option.

**–a** Attach to the system console as soon as the connection is established. By default sessions are view-only, and an escape sequence attach command (see below) must be typed to send input to the remote console port. If someone is already attached, a view-only connection will be established.

| | |
|---|---|
| **–A** | Force an attach to the system console as soon as the connection is established. If someone is already attached, their connection will be reduced to view-only. |
| **–v** | Neutralizes the effect of the **–a** and **–A** options. |
| **–d** *debuglevel* | Set program debug level. |
| **–e** *escapechar* | Set escape character. If *escapechar* is a single character it is used directly as the escape character. If *escapechar* is a multi-character sequence starting with a digit, it is interpreted according to **strtol**(3). If *escapechar* is **none** there is no escape character. Default escape character is tilde (~).. |
| **–f** *keylength* | Client encryption select parameter, where *keylength* is 0, 128. `-f 128` selects 128-bit encryption, `-f 0` disables encryption. Servers can be set to restrict TCP/IP connections to minimum key lengths through the ForceEncrypt=true configuration directive. |
| **–l** | Force a connection to be made locally, even if the **CONSOLE_SERVERS** environment variable is set. |
| **–L** | Neutralizes the effect of the **–l** option. |
| **–p** *port* | Changes the default TCP/IP port or service name used to contact remote location brokers. |
| **–P** | On TCP/IP connections, use location broker pass-thru feature. This only works with version 2.00 or later remote systems. When using this option, the only TCP connections made will be on the auracmgr/tcp port (364), which facilitates use across firewalls, Network Address Translation (NAT), and ssh, stunnel, or other port forwarding. The network connection is actually passed from the **locbrok**(8) process to the **conserv**(8) process, so there is no performance penalty. |
| **–N** | Neutralizes the effect of the **–P** option. |
| **–v** | Connect in view-only mode (neutralizes **–a** and **–A**). |
| **–o** *options* | Set **AURACMGR_OPTIONS** style options. |

## Escape Sequences

Lines that you type which start with the tilde character are \(lqescape sequences\(rq (the escape character can be changed using the -e option, see above).

| | |
|---|---|
| **~.** | Terminate the session. |
| **~CTRL/C** | Terminate the session. |
| **~CTRL/Z** | Suspend the **cmgr** program. |
| **~CTRL/L** | Toggle local logging of the connection. |
| **~a** | Attempt to attach (read-write) to the console. |
| **~A** | Force an attach (read-write) to the console; if anyone is currently attached, their connection will be downgraded to view-only. |
| **~d** | Detach from the remote console (make the connection view-only). |
| **~q** | Query server; a single character specifies the information to return: **A** (show All) **W** (show Who is connected to server), **T** (show Tail of log file) **V** (show Versions of server and console programs) **?** (show available options). |
| **~s** | Set a server variable (only if currently attached) a single character specifies the information to change; **s** (set stty parameters, a subset of the **stty**(1) command, options include: **crtscts -crtscts cstopb -cstopb parenb –parenb parext –parext parodd –parodd ixon –ixon ixoff –ixoff istrip -istrip cs5 cs6 cs7 cs8** or a baud rate), **T** (set distance back tail query will display in log file). **?** (show available options). |
| **~#** | Send a BREAK (only if currently attached). The user is prompted to confirm this action, which will be aborted if not confirmed. |
| **~?** | Display help on escape sequences. |

All other characters typed are sent to the remote system when attached. If not attached, the bell is rung for each character typed.

# ENVIRONMENT

### CONSOLE_SERVERS

see above.

### AURACMGR_OPTIONS

**AURACMGR_OPTIONS** Establish per-user defaults before checking command line options. **AURACMGR_OPTIONS** consists of a sequence of strings (and values) seperated by commas, one or more of;

| | |
|---|---|
| **attach** | see **–a** option. |
| **Attach** | see **–A** option. |
| **7bit** | see **–7** option. |
| **debug=***number* | see **–d** option. |
| **escape=***string* | see **–e** option. |
| **tail=***number* | see **–t** option. |
| **port=***string* | see **–p** option. |
| **local** | see **–l** option. |
| **nolocal** | see **–L** option. |
| **passthru** | see **–P** option. |
| **nopassthru** | see **–N** option. |

## SEE ALSO

**services**(4), **locbrok**(8)

## File Formats config(4)

## NAME

config – Aurora ControlTower Console Manager server configuration file

## SYNOPSIS

`/opt/AURAcmgr/config/DEFAULT`

`/opt/AURAcmgr/config/LOCAL`

`/opt/AURAcmgr/config/`*`group/group`*`.grp`

`/opt/AURAcmgr/config/`*`[group/]system`*`.cfg`

## DESCRIPTION

The Aurora ControlTower Console Manager **conserv** and **logcheck** programs read **DEFAULT**, site **LOCAL**, group, and per-system configuration files of the format described here. The **DEFAULT** file contains all global default values, and should not be edited. The **LOCAL** file is then read to allow avoid losing local changes to the **DEFAULT** file that might be lost in an upgrade. If the system config file is located in a subdirectory, that directory must contain a group configuration file with the same name as the directory, and the suffix **.cfg**"."

Finally, **the** *system***.cfg** file is read to supply values unique to a single system. The name of the configuration file determines the name of the managed system as known to the ControTower console management server, and need not be the official name of the server.

### File Format

Lines which start with a # character are treated as comments and ignored. Configuration lines are of the form *parameter=value*, where *parameter* is a

case-insensitive parameter name, and *value* is the parameter value.

### Parameter Syntax

Each parameter takes a value with one of the following syntaxes;

| | |
|---|---|
| *int* | Any integer value. A prefix of **0x** means value will be interpreted as base 16 (hex). A prefix of **0** means value will be interpreted as base 8 (octal). Otherwise the value is interpreted as base 10 (decimal). |
| *boolean* | A boolean value, one of: **1**/**t**/**true**/**y**/**yes** to enable a parameter, or on of **0**/**f**/**false**/**n**/**no** to disable a parameter. |
| *string* | An arbitrary string. |
| *mode* | File protection "mode", ether an octal constant (no leading digit required), or a symbolic value *[***ogua***]=[***rwxs***]+{,...}* (see **chmod**(1) man page). |
| *uid* | User id: decimal value or a user name from the **passwd**(4) file. |
| *gid* | Group id: decimal value or group name from the **group**(4) file. |
| *stty* | sequence of tokens/values (see **stty**(1) man page). Character size; **cs5**, **cs6**, **cs7**, **cs8**. Line speed (supported speeds depend on underlying hardware and operating system. custom speeds are not supported); **50**, **75**, **110**, **150**, **200**, **300**, **600**, **1200**, **2400**, **4800**, **9600**, **19200**, **38400**, **57600**, **76800**, **115200**, **153600**, **230400**, **307200**, **460800**. Flags (may be prefixed with '**–**' to disable); **crtscts**, **cstopb**, **parenb**, **parext**, **parodd**, **ixon**, **ixoff**, **istrip**. |

## Serial line parameters

| | |
|---|---|
| **device** | Syntax: *string*. **device** is the only parameter which must appear in the *system*.cfg file. This is the path of a tty device for the managed system. Call out devices **/dev/cua*** are typically used, to ignore changes in the state of the Data Carrier Detect (DCD) control line. |
| **stty** | Syntax: *stty.* Sets the initial terminal modes for the managed system serial connection. Any parameters missing from both the system and DEFAULT configuration will be left unmodified from system defaults (admintool terminal settings have no effect). |

**ttychanges** Syntax: *boolean*. Allow attached clients to change serial line parameters.

**uulock** Syntax: *boolean*. Honor and create uucp compatible lock files for the serial port.

**exclusive** Syntax: *boolean*. Set operating system 'exclusive access' flag on the serial port using the **TIOCEXCL** ioctl. Prevents non-superuser processes from opening the serial port.

**breakstring** Syntax: *string*. String to send to managed system instead of BREAK signaling. The following escape sequences are allowed: **\n** (newline), **\r** (return), **\t** (tab), *\OOO* (octal value), **\x***XX* (hex value).

## Log file parameters

**logdir** Syntax: *string*. Specify the absolute path to the directory for per-system log files. Log file names are by default the *system* name of the managed system, but can be explicitly specified using the **logfile** parameter. If the managed device is a member of a group, the device log file will be created in a subdirectory having the same name as the group.

**logfile** Syntax: *string*. Specify the absolute path to the file to which log output will be written. Defaults to `logdir/<servername>`, but can be customized to a pathname for each server individually or all servers combined.

**logfilter** Syntax: *string*. The path to an optional log file filter file. The path may be relative. Each line in the **logfilter** file starts with a filter type, one of **keep**, **drop**, or **alert**, followed by a delimited POSIX 1003.2 extended regular expression (see **regex**(7)), which may be followed by optional "tag" text. The delimiter character used to bracket the regular expression must not appear within the regular expression.

**loginput** Syntax: *boolean*. log user input (from attached user(s)). Will cause "double echo" of user input. No-echo input (passwords) will be logged!!

**loglinestamp** Syntax: *string*. Format string passed to **strftime**(3) to format timestamp on each line written to log file. If empty, lines are not timestamped.

**logmessages** Syntax: *boolean*. Log connect/disconnect/force messages (normally sent to connected users) to managed system log file (including parameter set messages).

**logmode** Syntax: *mode*. Protection for per-system log file. May be empty. Ignored if mode=0. Logfile mode is set on each (re)open for append.

**logowner** Syntax: *uid*. Owner for per-system log file. May be empty. The Log file owner is set on each (re)open for append.

**loggroup** Syntax: *gid*. Group for per-system log file. May be empty. Log file group is set on each (re)open for append.

**logstamp** Syntax: *integer*. Determines how often to timestamp the log file in minutes; one of: **10**, **20**, **30**, **60** or zero to disable periodic timestamps.

**logstampformat** Syntax: *string*. Format string passed to **strftime**(3) to format periodic logfile timestamps. If empty, periodic timestamps are not output.

**logmaxsize** Syntax: *int*. The maximum log file size (in bytes) before **logcheck**(8) rotates and compresses log files (zero to disable).

**logmaxfiles** Syntax: *int*. Maximum number of log files for logcheck to keep in rotation.

**logcompress** Syntax: *string*. Pathname of a program for **logcheck**(8) to use to compress old log files.

**logcompressopt** Syntax: *string*. Command line option(s) for **logcheck**(8) to pass to the program specified by the **logcompress** parameter. The compression program should always compress the input file, even if no savings will be realized.

**logcompressext** Syntax: *string*. The file extension (suffix) that the program specified by the **logcompress** parameter will add to the input file name.

## Local connection control parameters

**localenable** Syntax: *boolean*. True to allow local (unix domain) socket connections. This parameter was called **unixenable** in version 1.00 (which is still accepted as an alias).

**localauth** Syntax: *boolean*. True to force local (unix domain) socket connection users to be prompted for password (see **authuser** below). This parameter was called **unixauth** in version 1.00. (which is still accepted as an alias).

## Network access control parameters

**tcpenable** Syntax: *boolean*. True to allow TCP/IP connections.

**tcpallow** Syntax: *string*. If non-null, TCP connections will only be accepted if the

remote host matches a member of this list of comma seperated TCP hosts or networks to allow connections from. Hosts may be host names or IP addresses. Each may be followed with a forward slash (/) and an optional mask in dotted octet format, hex, or decimal network mask length. All of the following have the same effect: **/255.255.255.0**, **/0xffffff00**, **/24**. The mask determines which bits in the IP addresses will be examined: Any bit position with a zero mask bit will be ignored.

**tcpreject** Syntax *string*. If non null, TCP connections will be rejected if the remote host matches a member of this list (see **tcpallow** for syntax).

**authuser** Syntax: *string*. The name of a local user remote users must supply the password for when connecting. If not set (empty), or the user does not exist, no one can connect over the network (or locally if **unixauth** (see above) is set).

**authfile** Syntax: *string*. The path to an optional per-user authorization file, which contains a list of authorized users and their capabilities. The path may be relative. If the **authfile** parameter is not specified, all users must authenticate as the user specified by the **authuser** parameter. The authfile format is: *username* followed by one or more of the following; **connect** (may connect to server), **attach** (may attach in r/w mode), **fattach** (may force others off), **stty** (may change tty params), **break** (may send break), **all** (all of the above) seperated by plus (+) signs to add capabilities or minus (-) signs to subtract them.

### Idle time limit parameters

**detachidle** Syntax: *integer*. If non-zero, the maximum time in minutes before detaching (demoting to view-only) an idle attached viewer. If zero, no idle limit is enforced.

**disconnectidle** Syntax: *integer*. If non-zero, the maximum time in minutes before disconnecting an idle viewer regardless of whether viewer is attached or view-only. If zero, no idle limit is enforced.

## FILES

| | |
|---|---|
| `/opt/AURAcmgr/config/DEFAULT` | default values |
| `/opt/AURAcmgr/config/LOCAL` | site local default values |
| `/opt/AURAcmgr/config/group/group.grp` | group default values |

`/opt/AURAcmgr/config/[group/]system.cfg` per-system configuration

## SEE ALSO

**conserv**(8), **logcheck**(8).

## Maintenance Procedures conserv(8)

## NAME

conserv – Aurora ControlTower Console Manager server process

## SYNOPSIS

**conserv** [ **-d** *debuglevel* ] [ **-o** *parameter=value* ] *system*

## DESCRIPTION

Aurora ControlTower Console Manager server launches a **conserv** for each managed system. **conserv** reads the *system*.cfg file (see **config**(4)) and opens the serial port specified by the **device** parameter. **conserv** logs all managed system output in a file named *system* in the directory specified by the **logdir** configuration parameter. Users can connect to the **conserv** process using the **cmgr**(1) program. **conserv**(8) is normally launched by the **start**(8) script, but can be started by hand for debugging. Any number of **-o** options may be given, each with a *parameter=value* pair to override values in the *system*.cfg file. The **-d** option can be used to specify a debug level, which if non-zero keeps **conserv** from detaching from the terminal so that debug messages can be seen. Increasing debug levels increase the amount of debug output.

## FILES

| | |
|---|---|
| `/opt/AURAcmgr/config/system.cfg` | configuration file |
| `/opt/AURAcmgr/pids/system` | process id file |
| `/opt/AURAcmgr/sock/system` | unix-domain socket endpoint |

## SEE ALSO

**config**(4), **conserv**(8), **locbrok**(8), **start**(8).

## Maintenance Procedures convert(8)

## NAME

convert – Aurora ControlTower Console Manager config file converion tool

## SYNOPSIS

**convert** [ **-f** ] [ **-o** *outputdir* ] [ *inputfile ...* ]

## DESCRIPTION

**convert** reads input files (or the standard input if none are specified) that are tab or colon delimited and creates Aurora ControlTower Console Manager config files. The first column of the input is the managed system name, and the second is the serial device to which the managed system console is attached. Any remaining information is discarded. If a configuration file already exists, the entry will be skipped, unless the **-f** option is used, in which case the existing file will be saved as a .bak file. The **-o** option specifies the configuration file output directory.

## FILES

`/opt/AURAcmgr/config/system.cfg` system configuration files

## SEE ALSO

**config**(4), **conserv**(8).

## Maintenance Procedures Filtertest(8)

## NAME

filtertest – Aurora ControlTower Console Manager log filter test program

## SYNOPSIS

**filtertest** [ **-n** ] [ **-q** ] *filterfile* [ *inputfile* ]

## DESCRIPTION

**filtertest** reads an Aurora ControlTower Console Manager log filter file, checks the file for syntax, and reads an input file and applies the filters to each line of the input file. If no input file is specified, lines are read from the standard input stream. Each match is reported on the standard output, and a summary of each type of match (keep, drop, alert) is reported on standard errors when end of file is reached on standard input.

When the **-n** (no filter) option is specifed, *filtertest* will exit with zero (true) status after successfully parsing the filter file.

When the **-q** filter is specified, matches are not reported on standard output. A summary is still reported on standard error.

## SEE ALSO

**conserv**(8).

## Maintenance Procedures locbrok(8)

## NAME

locbrok – Aurora ControlTower Console Manager server Location Broker

## SYNOPSIS

**locbrok** [ **-d** *debuglevel* ]

## DESCRIPTION

The Aurora ControlTower Console Manager Location Broker reads and enforces the terms of the product licence file, and keeps a database of managed system names and the **TCP/IP** port the **conserv**(8) process for that managed system is available at. When a **cmgr**(1) is run remotely it first contacts one or more Location Brokers (on one or more servers) in order to find out what managed systems are available, what server they are attached to, and on which **TCP** port the **conserv (8)** process can be reached.

The **locbrok** process is normally launched by the **start**(8) script, and killed by the **stop**(8) script.

## FILES

| | |
|---|---|
| `/opt/AURAcmgr/config/license.dat` | license file |
| `/opt/AURAcmgr/pids/.locbrok` | process id file |
| `/opt/AURAcmgr/sock/.system/locbrok2` | unix-domain socket endpoint |

## SEE ALSO

**cmgr**(1), **conserv**(8), **start**(8), **stop**(8).

## Maintenance Procedures logcheck(8)

## NAME

logcheck – Aurora ControlTower Console Manager server logfile manager

## SYNOPSIS

**/opt/AURAcmgr/sbin/logcheck –i** *interval*

## DESCRIPTION

**logcheck** is run periodicly from the *root* user **crontab**(4) to manage logfiles for the Aurora ControlTower Console Manager server.

**logcheck**(8) honors the following configuration paramaters (see **config**(4) for descriptions); **logdir**, **logstamp**, **logmaxsize**, **logmaxfiles**, **logmode**, **logowner**, **loggroup**, **logcompress**, **logcompressopt**, and **logcompressext**.

## SEE ALSO

**config**(4), **logcheck**(8).

## Maintenance Procedures start(8)

## NAME

start – Aurora ControlTower Console Manager server start script

## SYNOPSIS

**/opt/AURAcmgr/sbin/start** [ *system group ...]*

## DESCRIPTION

The **start** script launches Aurora ControlTower Console Manager server processes.

If no arguments are given **start** first runs the **stop**(8) script to stop all server activity, then it launches a a location broker process (see **locbrok**(8)) and a **conserv**(8) process for each file ending in **.cfg** in the `/opt/AURAcmgr/config` directory.

If an argument list is given, **start** launches a **locbrok**(8) process if needed, and a **conserv**(8) for each system named in the argument list. If the argument specifies a group directory, a **conserv**(8) will be launched for each **.cfg** file in the named directory.

## FILES

| | |
|---|---|
| `/opt/AURAcmgr/config/*.cfg` | config files |

## SEE ALSO

config(5), conserv(8), locbrok(8), stop(8).

## Maintenance Procedures stop(8)

## NAME

stop – Aurora ControlTower Console Manager server stop script

## SYNOPSIS

**/opt/AURAcmgr/sbin/stop** [ *system group ...*]

## DESCRIPTION

The **stop** script stops Aurora ControlTower Console Manager server processes.

If no arguments are given **stop** kills all processes which have left pid files in the **/opt/AURAcmgr/pids** directory.

If an argument list is given, **stop** kills the **conserv**(8) process for each system named in the argument list. If the argument specifies a group directory, all running **conserv**(8) processes in the group will be killed.

## FILES

`/opt/AURAcmgr/pids/*` pid files

## SEE ALSO

config(5), conserv(8), locbrok(8), start(8).

❒

**APPENDIX B**

# *DEFAULT Configuration File*

The DEFAULT configuration file (`/opt/AURAcmgr/config/DEFAULT`) specifies the default configuration for devices managed by an Aurora ControlTower server. These configuration specifications apply to every managed device unless overriden in the LOCAL configuration file, the <*group_name*>/<*group_name*>.grp file, or the configuration file for that device (`/opt/AURAcmgr/config/`<*device_name*>.cfg.)

```
# Aurora ControlTower Console Manager DEFAULT configuration
#
# ******************* DO NOT EDIT THIS FILE *******************
#
# This file is read before the LOCAL file, <group>/<group>.grp files,
# and <system>.cfg files.  Any changes to these defaults should be
# made by adding lines to the LOCAL, or per-group configuration files.
#
######################################################################
# COPYRIGHT (c) 1998 - 2002 BY AURORA TECHNOLOGIES, INC., BROCKTON, MA.
#
# THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND
# COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH
# THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR
# ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE
# AVAILABLE TO ANY OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE
# PROGRAM IS HEREBY TRANSFERRED.
#
# THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
```

```
# NOTICE AND SHOULD NOT BE CONSIDERED AS A COMMITMENT BY AURORA
# TECHNOLOGIES, INC.
#
#
######################################################################
#                     **** NOTE WELL ****
######################################################################
#
# With very few exceptions, a value for each parameter MUST be
# specified.  NO default values are present in the code, so
# values must be specified here, or in a per-system config file.
# All parameters for which a default can be reasonably picked
# appear here.
#
######################################################################
# serial line parameters
#
# devicename for serial port attached to system console
# syntax: string
#device=
#
# set O_EXCL "exclusive open" bit on serial port open
# syntax: boolean
exclusive=true
#
# create (and honor) UUCP-compatible lock files for the serial port
# syntax: boolean
uulock=true
#
# tty mode.
# syntax: one or more tokens separated by commas or spaces
# tokens;
#              integer (speed/baud)
#              cs5 cs6 cs7 cs8
#              flag
#              -flag
# flags:
#              crtscts cstopb parenb parext parodd ixoff ixon istrip
#
# ALL flags/parameters should appear here in DEFAULT file.  subsequent
# stty configuration (in <system>.cfg or with -o on command line, or
# via console program "set" command) change ONLY the bits which are
# specified (all other remain the same).
stty=9600 cs8 -crtscts -cstopb -parenb -parext -parodd -ixoff -ixon istrip
#
# allow client programs to change serial line parameters
# syntax: boolean
ttychanges=true
#
```

```
# string to send instead of BREAK signal (optional)
# syntax: string
# the following escape sequences are allowed;
#              \r \n \t \ooo (1 or more octal digits) \xXX (two hex digits)
#breakstring=
#
######################################################################
#
# Logfile parameters
# All system console output is saved in a logfile.
#
# directory for all log files (must be absolute)
# syntax: string (path)
logdir=/var/log/cmgrlog
#
# log client (user) input in logfile (THIS INCLUDES PASSWORDS!!)
# all output (including echo) is always saved in the logfile
# syntax: boolean
loginput=false
#
# log messages sent to users (user connect/disconnects) in logfile
# (serial line change and break messages are always logged)
# syntax: boolean
logmessages=true
#
# owner for log files
# syntax: user name or uid
logowner=root
#
# group for log files
# syntax: group name or gid
loggroup=sys
#
# mode for log files
# syntax: octal mode (e.g.; 0600) or comma separated sequence
#              of symbolic absolute modes strings [uoga]=[rwxs]+
logmode=u=rw
#
# Optional: strftime(3) format used to timestamp lines in logfile
#              if empty, lines are not time-stamped
# syntax: string
loglinestamp=%c
#
#################
# Logfile parameters read by "logcheck" program run every
# 10 minutes from root crontab;
#
# Maximum logfile size in bytes before closing and "rotating";
# syntax: integer
```

```
logmaxsize=50000
#
# Number of old log files to compress and keep in "rotation";
# syntax: integer
logmaxfiles=7
#
# Optional: how often to timestamp logfile in minutes; one of: 10, 20, 30, 60
#              or zero to disable.
# syntax: integer
logstamp=60
#
# strftime(3) format used to output periodic logfile timestamps;
#              if empty, no periodic timestamps will be output
# syntax: string
logstampformat=**************** %c ***************
#
# Logfile compression program path
# syntax: string
logcompress=
#
# Logfile compression program options; compression program is expected
# to ALWAYS compress the log file even if this does not result in a
# space savings
# syntax: string
logcompressopt=-f
#
# Logfile compression program output extension (including DOT character)
# syntax: string
logcompressext=.Z
#
# Optional log filter file path.  If the path is relative (no leading
# slash), the pathname will be taken as relative to the directory in
# which the device .cfg file was found.
#
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
#
# syntax: string (path)
#logfilter=
#
######################################################################
#
# Authorization parameters
#
# The name of the user remote users must supply the password for.
# If authfile (below) is specified, authuser only applies to version 1
# viewers.
# syntax: string (user name)
authuser=auracmgr
```

```
#
# The path to an optional per-user authorization file which contains a
# list of authorized users and their capabilities.  The path may be
# relative (to group or config directory).  If not specified, all
# users must authenticate as "authuser", if set users will be prompted
# for a user name.
#
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
#
# syntax: string (path)
#authfile=
#
######################################################################
#
# Idle time parameters
#
# If non-zero, the maximum time in minutes before disconnecting an
# idle viewer regardless of whether view is attached/view-only.  If
# zero, no idle limit is enforced.
# syntax: integer (minutes)
disconnectidle=0
#
# If non-zero, the maximum time in minutes before detaching (demoting
# to view-only) an idle attached viewer.  If zero, no idle limit is
# enforced.
# syntax: integer (minutes)
detachidle=0
#
######################################################################
# local (Unix domain) socket parameters
#
# Allow local (Unix-domain) connections
# was called unixenable in Version 1.00
# (old name still accepted)
# syntax: boolean
localenable=true
#
# Require password on local (Unix-domain) connections;
# was called unixenable in Version 1.00
# (old name still accepted)
# syntax: boolean
localauth=false
#
######################################################################
# TCP socket parameters
#
# allow TCP/IP connections
# syntax: boolean
```

```
tcpenable=true
#
# The following have no default value, and may be left blank.
# syntax: comma separated list of host/mask pairs.
#
# The host may be a name (from /etc/hosts or DNS) or dotted decimal octets
# (nnn.nnn.nnn.nnn).  The mask is optional, and can be used to specify
# which bits of the host address are to be examined.
#
# 1) a single decimal number (/24) signifying the number of high-order
#              bits set in the mask
# 2) four dotted decimal octets (/255.255.255.0)
# 3) a single hexadecimal value (/0xffffff00)
#
# if no mask is supplied, all bits in the host address are examined,
# so "host/32" is the same as "host"
#
# If set a host must match an entry in "tcpallow" in order to be accepted.
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
# syntax: string (acl)
#tcpallow=
#
# If set a host must NOT match an entry in "tcpdeny" in order to be accepted
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
# syntax: string (acl)
#tcpdeny=
```

# APPENDIX C *Glossary*

**Attach** - see Read/Write mode.

**AURAcmgr** - The ControlTower command line Viewer Client software package. Required for installation of AURAcmgrs package.

**AURAcmgrs** - The ControlTower Server Software package. Requires that the AURAcmgr package be installed. AURAcmgrs is required for installation of the AURAjcmgr package.

**Break Signal** - An RS-232 signal that for some managed devices is interpreted as a device reset command.

**Breakout Box** - Hardware used to connect RS-232 serial devices to multiport serial cards. Also referred to as a Connection Box.

**Character Oriented Viewer Clien**t - Software supplied by Aurora Technologies that provides access to the managed devices' console serial port through a character-oriented window. Also known as "CLI Viewer Client" and "Command Line Viewer Client".

**CLI Viewer Client** - see Character Oriented Viewer Client.

**cmgr** - Aurora Technologies supplied software program running the Character Oriented Viewer Client functionality in the existing terminal window.

**Command Line Viewer Client** - see Character Oriented Viewer Client.

**Connect** - The act of connecting to a managed device.

**Connection Box** - see Breakout Box.

**Console Management** - see Console Management Services.

**Console Management Services** - Logging and real time viewing of output from managed devices, and control of managed devices.

**Console Serial Port** - The serial port on the managed device whereby commands can be sent and data received. Also known as “Serial Console Port” and “Console Port”.

**Control** - See Read/Write mode.

**ControlTower Host**- see ControlTower Host Server System

**ControlTower Host Server System** - The computer on which the ControlTower Server Software has been installed, regardless of whether any Aurora Hardware is installed.

**ControlTower Host System** - The computer system including: Breakout Boxes, Multiport Serial Cards, Expansion Chassis, and ControlTower Software. Also known as “Host”, “Host System”

**ControlTower Server Software** - Software supplied by Aurora Technologies that provides console management services, see the entries for AURAcmgrs, AURAjcmgr.

**Force Control** - See Force Read/Write mode.

**Force Read/Write Mode** - The ability of the Viewer Client to take Read/Write mode if there is already another user that has Read/Write mode.

**Host** - See ControlTower Host Server System.

**Host System** - See ControlTower Host Server System.

**Local Viewer Client** - A Character Oriented Viewer Client that is run directly on the ControlTower Host, without the CONSOLE_SERVER environment variable defined.

**Log** - see Log File.

**Log File** - Output from a managed device that is stored locally on the ControlTower Host.

**Managed Device** - A computer or other system that accepts basic management commands over an RS-232 serial interface; see Console Serial Port.

**Monitor** - See Read-Only mode.

**Network Client** - GUI or CLI connection to ControlTower Host using TCP/IP.

**Package** - A Solaris software package that is installed on a computer system using the Solaris system command, pkgadd. Package removal is done with the Solaris system command, pkgrm.

**Read-Only mode** - The ability of the Viewer Client to monitor output from the managed device. A Viewer Client connection that allows the user to view all managed device output as it happens, but not to send any keystrokes to the managed device. Requires the “connect” capability in the managed system authfile.

**Read/Write mode** - The ability of the Viewer Client to interact with the managed device. A Viewer Client connection that allows the user to see all managed device output as it happens, and to send keystrokes to the managed device. The act of entering read/write mode is called “attaching”, and requires the “attach” capability in the managed system authfile. If another user is currently attached (in read/write mode), you can forcibly take control away from them (this requires both the “attach” and “fattach” capabilities in the managed system authfile).

**View** - see Read-Only mode.

**Viewer Client** - Aurora Technologies supplied software that provides ability to issue commands to a managed device’s console serial port, view log files and interact with the ControlTower Server Software.

**xcmgr** - Aurora Technologies supplied software script program running the Character Oriented Viewer Client functionality in an xterm.❐

# APPENDIX D *An Example Configuration*

This is an example configuration with devices at various levels and in groups and with parameters changed from their defaults at various levels. This configuration has been set up and run in a lab.

Devices have been configured at the top level in the `/opt/AURAcmgr/config` directory and under a group directory. The files and directories under `/opt/AURAcmgr/config` are:

```
DEFAULT
LOCAL
LOCALauthfile
device2.cfg
device3.cfg
group/
license.dat
```

Under the `group` directory are the files:

```
device0*
device0.1.test*
device0.cfg
device1.cfg
```

```
group.grp
groupauthfile
```

Modifications to global parameters have been made in the LOCAL file instead of the DEFAULT file. The LOCAL file is:

```
# Don't allow client programs to change serial line parameters
ttychanges=false
#
# So no one can send a 'break' to a device, set the
# breakstring to the text 'NO!'.
breakstring=\116\117\041
#
# Everyone can read the log files.
logmode=u=rw,g=r,o=r
#
# Set loglinestamp to NULL so that lines are not time-stamped.
loglinestamp=
#
# Since there are no timestamps per line, there will be timestamps every 10
# min.
logstamp=10
#
# Add the week # to the default periodic time stamp.
logstampformat=**************** %c Week#: %U ****************
#
# Logfile compression program path
# syntax: string
# Change compression utility to gzip for better compression.
logcompress=/usr/bin/gzip
#
# This is the same as in the DEFAULT file but is left here for clarity.
logcompressopt=-f
#
# This is the extension that is appended by gzip.
logcompressext=.gz
#
# Put logfilter file with logfiles it affects. This file will filter input to
all log files,
# including ones under a group
logfilter=/var/log/cmgrlog/LOCALlogfilterfile
#
# "Unset" the authuser parameter
```

```
authuser=
#
# This file currently contains 'auracmgr all' so that only people with the
# auracmgr
# password have access to devices.
authfile=LOCALauthfile
#
# Set so that a viewer will disconnect if it's been idle for 5 minutes.
disconnectidle=5
#
# Set so that a viewer will detach if it's been idle for 10 minutes.
detachidle=10
#
# Require authorization for even local connections to devices.
localauth=true
#
# Keep out the people in Marketing
tcpdeny=100.100.100.100/8
```

Notice that `authfile` is set to `LOCALauthfile` with no specified path. This is why a `LOCALauthfile` file has been created in the `/opt/AURAcmgr/config` directory. The contents of this file are:

```
auracmgr all
```

The contents of the two device configuration files in the `/opt/AURAcmgr/config` directory are:

`device2.cfg:`

```
device=/dev/cua/2
```

`device3.cfg:`

```
device=/dev/cua/3
```

The `group.grp` group configuration file under the group directory contains:

```
# The devices in this group belong to development so we will be
# giving more people access to these devices.
authfile=groupauthfile
```

This is why a `groupauthfile` file has been created under the group directory. Its contents are:

```
# As developers are hired, add them to this file.
auracmgr all
developer1 all-break
developer2 attach+fattach+stty
```

The configuration file for device0 under the group directory contains:

```
# device0 has been having problems so the settings are being
# changed temporarily to facilitate diagnosis.
#
# devicename for serial port attached to system console
device=/dev/cua/0
#
# allow client programs to change serial line parameters
ttychanges=true
#
# In case of problems, we want to be able to send a real break.
breakstring=
#
# This is the directory containing configuration files. The log files will be
# sent here for easy access.
logdir=/opt/AURAcmgr/config
#
# This will write everything typed into a cmgr session, including PASSWORDS!
loginput=true
#
# logmessages=true in the DEFAULT file.
#
# mode for log files
# Results in -rwxrw-r--
logmode=764
#
# To make debugging easier, line-by-line time stamping is being turned back
# on with the default value.
loglinestamp=%c
#
# This will keep larger log files and more of them.
logmaxsize=500000
#
logmaxfiles=20
#
# Since line time-stamping is back, we don't need as many periodic time
# stamps.
logstamp=60
#
```

```
# Return the format of the periodic time stamp to the default.
logstampformat=*************** %c ***************
#
# Logfile compression program path
# Use compress so that log files can be copied to, and uncompressed on,
# machines that don't have gzip.
logcompress=/usr/bin/compress
#
# This still hasn't changed but is left for clarity.
logcompressopt=-f
#
# Logfile compression program output extension (including DOT character)
# The compress extension is .Z
logcompressext=.Z
#
# Turn off log filtering.
logfilter=*novalue*
#
# Set authuser to the user whose password everyone has.
authuser=auracmgr
#
# Turn off per-user authorization.
authfile=*novalue*
#
# Turn off idle disconnect.
disconnectidle=0
#
# Turn off idle detach.
detachidle=0
#
# Don't require authorization for local connects.
localauth=false
#
# Turn off tcpdeny so that everyone can look at managed devices.
tcpdeny=*novalue*
#
# Turn on errlog so that error messages will be written to somewhere
# other than syslog.
errlog=device0errlog
#
# Turn on debugging for as much info as possible.
debug=1
```

The settings in this file have been chosen to maximize the information written to log files, including the additional log file, `device0errlog`, which is written to the directory in which the ControlTower server was started, in this case, `/opt/AURAcmgr/config`. Notice that the value of `logdir` has been changed to `/opt/AURAcmgr/config`. This is why there are `device0` log files, `device0` and `device0.1.Z`, in the `/opt/AURAcmgr/config` directory listing above. These files are listed with a star after the name indicating that these are executable files because logmode was set to 764 (*u=rwx, g=rw, o=r*).
Notice also that `authuser` is set to `auracmgr` for this device.

The default log directory of `/var/log/cmgrlog`, contains files and directories:

```
device2
device2.1.gz
device2.2.gz
device3
device3.1.gz
group/
```

Under the group directory is:

```
device1
device1.1.gz
device1.2.gz
```

`device1` is the device under the group directory that has no parameter changes of its own. ❒

# APPENDIX E *Warranty*

## Warranty Information

### Hardware

All Aurora Technologies, Inc. hardware products are warranted against defects for two years from the date of delivery. Buyer agrees that if this product proves defective, Aurora Technologies, Inc. is obligated only to repair, replace or refund the purchase of this product at Aurora Technologies, Inc.'s discretion. The warranty is void if the product has been subjected to alteration, neglect, misuse or abuse; if any repairs have been attempted by anyone other than Aurora Technologies, Inc.; or if failure is caused by accident, acts of God, or other causes beyond the control of Aurora Technologies, Inc.

Aurora Technologies, Inc. reserves the right to make changes or improvements in any product without incurring any obligation to similarly alter products previously purchased. In no event shall Aurora technologies, Inc. be liable for any direct, indirect, incidental or consequential damages arising out of or in connection with the performance or use of the product or information provided. Aurora Technologies, Inc.'s liability shall in no event exceed the purchase price of the product purchased hereunder.

The foregoing limitation of liability shall be equally applicable to any service provided by Aurora Technologies, Inc.

## Application and Protocol Software

**Limited Software Warranty.** Aurora Technologies, Inc. does not warrant that the functions contained in its Software products will meet your requirements or that the operation of the Software will be uninterrupted or error free. However, Aurora warrants the physical media on which the Software is furnished will be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of shipment.

Except as provided above, the software and its written materials are provided "As-Is," without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the software is with the licensee. Should the software prove to be defective, licensee assumes the entire cost of necessary servicing, repair, or correction.

## Return Policy

Products returned for repair must be accompanied by a Return Material Authorization (RMA) number, obtained from Aurora Technologies, Inc. prior to return. Freight on all returned items must be prepaid by the customer, and the customer is responsible for any loss or damage caused by common carrier in transit. Items will be returned to Aurora Technologies, Inc. via a Customs cleared carrier (for example, Federal Express, UPS, DHL), unless prior arrangements are made by the customer for an alternative shipping method. Each product that is returned for repair must include a failure report and must have the RMA number clearly marked on the outside packaging.

Return Address:
Attn.: RMA Department
Aurora Technologies, Inc.
646 Summer Street
Brockton, MA 02302
USA

## 90 Day Technical Support

Products must be registered with Aurora Technologies, Inc.'s Customer Service and Support (CSS) organization to receive the 90 Day Technical Support. You must fill out and mail or FAX the warranty card that is included with the product before receiving technical assistance.

### What you get during the 90 Day Technical Support

90 Day Technical Support is provided by e-mail, FAX, or by telephone. Customers calling in for technical support on current Aurora technologies, Inc. products will receive a response within four (4) business hours. Customer's e-mails or faxed requests will receive a response within twenty-four (24) business hours.

The Technical Support hours in Massachusetts are

> 8:30 a.m. - 6:00 p.m. Eastern Time,
> Monday through Friday, excluding holidays.

Services provided under the 90 Day Technical Support Plan are:

- Help on installation and configuration
- Help diagnosing problems with Aurora hardware and standard released Aurora device drivers.
- Help navigating and locating existing Aurora documentation
- Acceptance of bug reports and providing status updates on any applicable bug fixes.

Policies and pricing are subject to change without notice.

For extended support, please refer to the Aurora Technologies, Inc. web site at www.auroratech.com or call your Sales representative for details.

❒

# Software License Agreement

THIS LEGAL DOCUMENT IS AN AGREEMENT BETWEEN YOU, THE END USER OR "LICENSEE", AND AURORA TECHNOLOGIES, INC. ("AURORA" OR "LICENSOR"). BY OPENING THIS DISTRIBUTION MEDIA PACKAGE, YOU ARE AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT, WHICH INCLUDES THE SOFTWARE LICENSE AND SOFTWARE WARRANTIES (COLLECTIVELY THE "AGREEMENT").

THIS AGREEMENT CONSTITUTES THE COMPLETE AGREEMENT BETWEEN YOU, THE LICENSEE AND AURORA. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT OPEN THE DISTRIBUTION MEDIA PACKAGE. PROMPTLY RETURN THE UNOPENED PACKAGE, HARDWARE (IF ANY) AND THE OTHER ITEMS, INCLUDING WRITTEN MATERIALS, BINDERS AND OTHER CONTAINERS, THAT ARE PART OF THIS PRODUCT TO THE PLACE OF PURCHASE.

## Aurora Technologies Software License

1. GRANT OF LICENSE:. In consideration of payment of the License fee, which is part of the price paid for this product, Aurora, as LICENSOR, grants to you, the LICENSEE, a non-exclusive right to use and display this copy of an Aurora Software program, (the "Software") on a single computer at a single location. If the single computer on which you use the Software is a multi-user system, this License covers all users of that system. Aurora reserves all rights not expressly granted to the LICENSEE.

2. DELIVERY, INSTALLATION, ACCEPTANCE AND RISK OF LOSS: Aurora shall deliver the Software to a common carrier, FOB Aurora's Facilities. LICENSEE shall be solely responsible for installation of the Software on the computer. LICENSEE agrees the acceptance shall occur upon delivery of the Licensed Software to a common carrier. LICENSEE assumes all risk of loss or damage upon delivery of the Software by Aurora to LICENSEE or common carrier.

3. PAYMENTS: In consideration of the License and rights in the Software granted by Aurora and in consideration of Aurora's performance of its obligations hereunder, LICENSEE agrees to pay Aurora Technologies, Inc. the License fee according to the terms as invoiced. Failure to remit complete payment according to the specified terms will result in revocation of the License upon thirty (30) days' written notice.

4. OWNERSHIP OF SOFTWARE: As the LICENSEE, you own the magnetic or other physical media on which the Software is originally or subsequently recorded or fixed, but Aurora in no way transfers title or ownership of the Software recorded on the original distribution media and all subsequent copies of the Software, regardless of the form or media in or on which the original and other copies may exist. This License is not a sale of the original Software or any copy.

5. COPY RESTRICTIONS: This Software and the accompanying written materials are copyrighted. Unauthorized copying of this Software, including Software that has been modified, merged or included with other Software, or of the written materials is expressly forbidden. LICENSEE may be held legally responsible for any copyright infringement that is caused or encouraged by LICENSEE's failure to abide by the terms of this License. Subject to these restrictions, and if the Software is not copy protected, LICENSEE may make one (1) copy of the Software solely for backup purposes. LICENSEE must reproduce and include any copyright notice on the backup copy.

6. USE RESTRICTION: LICENSEE may physically transfer the Software from one computer to another provided that the Software is used on only one computer at a time. LICENSEE may not distribute copies of the Software or accompanying written materials to others. LICENSEE may not translate, decompile, disassemble or otherwise reverse engineer, or modify, adapt or create derivative works based on the Software. LICENSEE may not modify, adapt, translate or create derivative works based on the written materials without prior written consent of Aurora.

7. TRANSFER: LICENSEE may sell the License rights in the Software to another party only if that party also agrees to the terms and conditions of this Agreement. In accordance with such sale, the LICENSEE must simultaneously transfer any and all written materials and the backup copy, or destroy the backup copy.

8. TERM AND TERMINATION: The License is effective until terminated. This License will terminate automatically without notice from Aurora if LICENSEE fails to comply with any provision of this License. Upon termination you shall destroy the written materials and all copies of the Software, including modified copies, if any.

9. UPDATE POLICY: Aurora may create, from time to time, updated versions of the Software. At its option, Aurora may make such updates available to the LICENSEE and transferees who have purchased an Extended Support Plan from Aurora.

10. EXPORT RESTRICTIONS: The LICENSEE understands that the United States export control laws may govern the export and re-export of products to be licensed under this Agreement and that individual validated export licenses may be required from the U.S. Department of Commerce prior to the export of products. The LICENSEE agrees to assist the LICENSOR to obtain any required License by supplying appropriate documentation requested by the seller. The LICENSEE agrees to comply with the U.S. Export Administration Regulations in effect from time to time and will not re-export any products without first obtaining approval from the LICENSOR and the U.S. Department of Commerce as required. The re-export of LICENSOR's source code, whether modified or not, is prohibited without prior written approval from the LICENSOR. If the LICENSOR grants the LICENSEE approval to re-export source code, LICENSEE will obtain all required export approvals from the U.S. Department of Commerce prior to sale and shipment. The LICENSEE agrees to indemnify and hold harmless the LICENSOR from all costs and expenses incurred by the LICENSOR as a result of the LICENSEE's breach of this section.

11. RIGHT TO GRANT A LICENSE: Aurora hereby warrants that it has the right to grant a License to use the Software to the LICENSEE and that it has the right and power to enter into this License.

12. LIABILITIES: In no event will Aurora be liable for any lost revenues or profits or other special, indirect or consequential damages, even if Aurora has been advised of the possibility of such damages. Aurora's maximum liability for damage shall be limited to the License fees paid by the LICENSEE under this License for the particular Software which caused the damages.

13. GENERAL: LICENSEE may not sublicense, assign or transfer the License on the Software except as expressly provided in this Agreement. Any attempt otherwise to sublicense, assign or transfer any of the rights, duties or obligations hereunder is void. This Agreement will be governed by the laws of the Commonwealth of Massachusetts in the United States of America. ❐

# Index