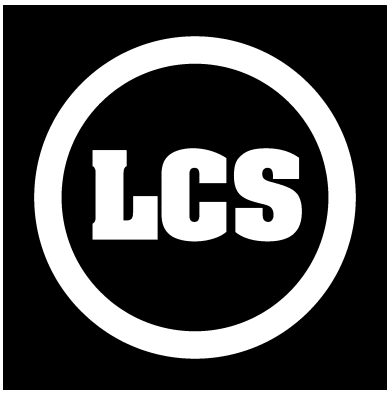
Level Control Systems

CueStation v3.1 Users Guide

Quick Contents

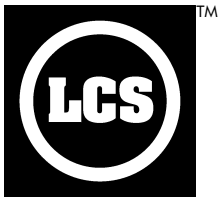
1 Welcome to CueStation
2 Using BeOS
3 Understanding CueStation
4 Tutorials
5 Common Tasks
6 Commands Reference
7 Hardware Reference
8 Appendices



CueStation™ v3.1
Users Guide

Level Control Systems™, Inc.

Bringing You the Future
a Little Ahead of Schedule℠



This manual was prepared using

- BeOS v5.0.3
  (tested with & without BeOS Network Extension BoNE 7a)
- LX-DSP EPROM . . . . . v1.6.1 & v1.4.3)
- CueStation v3.1.1
- CueLibServer v1.1.1
- Externals editor v1.1.2
- NetMeters v1.2
- Subcue Librarian v1.1.1
- Templates v020115
- VRAS editor v1.2.1
- Wild Tracks editor v1.2.1
- LX main firmware v1.1.2
- EtherTracks firmware v1.1.8

Please record your version numbers

Contents

Preface vi
Acknowledgments viii
User Guide Conventions x

1 Welcome to CueStation
What's New in CS Version 3.0 1-1
About the Help Guides 1-2
About Level Control Systems 1-4

2 Using BeOS
BeOS For New Users 2-1
Customizing BeOS 2-5
Mounting a Device 2-8
Accessing Files 2-9

3 Understanding CueStation
CueStation and the Matrix^3 3-1
Automation Techniques 3-8
Matrix Mixing 3-15
SpaceMaps 3-21
Project Management 3-31

4 Tutorials
SpaceMap Tutorial 4-2

5 Common Tasks
Basic Interface 5-1
Basic Faders 5-3
Basic Cues 5-4
Basic Transport — Running a Show 5-7
Basic EQ, Delays, and Compression 5-9
Basic SpaceMap 5-12
Basic Wild Tracks 5-19
Basic Externals 5-24
Using Boot Cues 5-28
Configuring the Matrix^3 5-29
Setting Up the Network 5-48
Configuring Ethernet 5-50
Configuring CobraNet 5-51
Directories Quick Reference 5-55
Accessing Files in an Archive 5-56

6 Commands Reference
Commands Listed by Window 6-1
Most Windows 6-1

Subcue Librarian *6-2*
Transport *6-2*
Cue List *6-3*
Cue Library *6-3*
Subcue Library *6-4*
Externals Library *6-4*
SpaceMap *6-5*
Console *6-6*
Bus Levels *6-6*
Matrix *6-6*
Output Masters *6-7*
Aux Masters *6-7*
Virtual Groups *6-7*
Input EQ *6-7*
Output EQ *6-7*
Aux Output EQ *6-8*
Log *6-8*

Commands Listed by Action 6-9
Application Control *6-9*
Automation Control *6-9*
Cue Control *6-10*
Cue Edit *6-10*
Cue List Edit *6-10*
EQ Edit *6-10*
Externals Edit *6-10*
Hardware Control *6-11*
Hardware Setup *6-11*
Information *6-11*
Log *6-11*
Matrix Edit *6-11*
Project Control *6-12*
Project Edit *6-12*
Software Control *6-12*
SpaceMap Edit *6-12*
SubCue Control *6-13*
Subcue Edit *6-13*
User Interface *6-14*

7 Hardware Reference
Frame Specifications 7-1
Facility Power Requirements 7-2
Component Specifications 7-3

8 Appendices
Templates 8-1
Glossary 8-8
Guide Questionnaire 8-11

Preface

The *CueStation v3.0 Users Guide* is a supplement to the CueStation classroom training course. CueStation, coupled with the Matrix^3, is a complex and extremely powerful sound automation system. The combination of classroom training and this Users Guide will have you up and running quickly and effectively.

How This Guide Is Organized

Chapter 1, ***Welcome to CueStation,*** introduces Level Control Systems (manufacturers of the Matrix^3 audio automation system and CueStation programming software), describes the conventions used in this Users Guide, and provides details on your documentation and support options.

Chapter 2, ***Using BeOS,*** deals with the differences between BeOS and the operating systems you are most likely to be familiar with. You will learn how to customize some of the system settings, and how to access storage devices and files. Additional tasks — those that deal with CueStation in some manner — are addressed in the tasks reference section of the Users Guide.

Chapter 3, ***Understanding CueStation,*** contains a series of articles on the functions, structure, and use of CueStation and the Matrix^3. It provides excellent pre-training reading material, and is a useful post-training reference as you become an adept automation programmer.

Chapter 5, ***Common Tasks,*** is directed toward the beginning programmer.

Chapter 6, ***Command Reference,*** documents every command and keyboard shortcut available in CueStation. These are listed in two formats: by application, and by function.

Chapter 7, ***Hardware Reference,*** provides some basic information regarding the expansion cards available for the Matrix^3 hardware, and pin-out diagrams. The information may be of use when troubleshooting.

The ***Appendices*** contain information about templates, and the glossary and index.

Acknowledgments

The *CueStation Users Guide* was developed by Level Control Systems, in collaboration with David Priest (WritersBlock Technical Communications).

Thanks go to Richard Zvonar (primary author, Ch. 3; editor), Richard Bugg (technical reviews; editor), Steve Ellison, Conrad Krieg, Brent Hoffman, Steven Noftal, John McMahon, Michael Dascenzo, and other LCS staff.

CueStation version 3.*x* was written by Jeremy Friesner, with a little help from Brian Tietz, Steve Ellison, and Jeff Koftinoff. Internal testing was performed by Richard Bugg, Conrad Krieg, Richard Zvonar, and David Priest. Special thanks to our beta team including Francois Bergeron, Kelly Prince, David Starck, and Garth Hemphill. Steve Ellison held the baton. James McCartney built the foundation.

We must also thank our users: your feedback and feature requests have driven our product development, and we are thrilled to provide you the world's best hardware and software sound automation system.

User Guide Conventions

It is important that you understand and remember the conventions used in this documentation.

You may find that BeOS has a “look and feel” that is different than that of the operating systems you have previously used. You may need to adjust some of your keyboard and mouse skills. Please refer to *Using BeOS,* chapter 2, for details.

Mouse Conventions

When you see this...	Do this...
Click	Click the left mouse button, once.
Double-click	Click the left mouse button, twice in rapid succession, without moving the mouse.
Drag	Press the left mouse button, move the mouse, and then release the mouse button.
Right-click	Click the right mouse button, once.
Right-drag	Press the right mouse button, move the mouse, and then release the mouse button.
Select	Point the mouse cursor at the object, and then click the left mouse button, once.

Keyboard Conventions

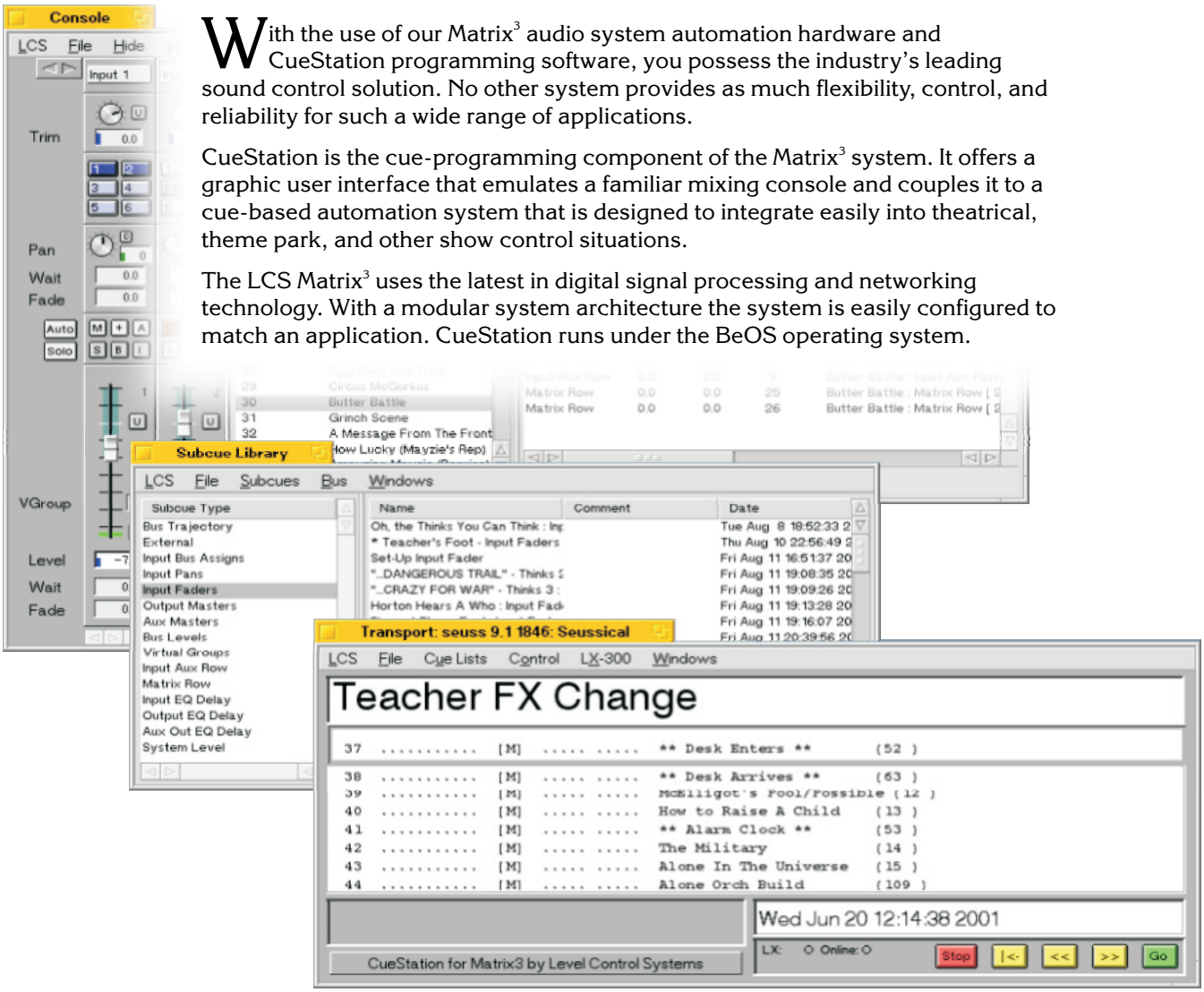
When you see this...	Do this...
Press F1	Press (and release) the “F1” key on your keyboard.
CTRL+SHIFT	Hold the “Ctrl” key down, and press the “Shift” key.
Type “`ping 127.0.0.1`”	Type the text that is surrounded by quotation marks(“ ”). Do not type the quotation marks.

Typographic Conventions

When you see this...	It means this...
Open the **File** menu Click the **Done** button ...in the **duration** box.	Bold text indicates commands that are issued as soon as you choose them. It also indicates labels for user interface objects, like data entry boxes.
Press ENTER Press ALT+G	Small-caps text indicates keyboard commands that are issued as soon as you press the keys that are described.
Type "`ping 127.0.0.1`"	"Typewriter" text indicates a command or value that you are to type into a value box or command line.
See *User Guide Conventions*	Italicized text indicates a cross-reference to information in this user guide or another manual.
 **Tip**	Provides information that can help you work faster or better.
 **Note**	Provides an explanation or further detail that needs to be brought to your attention.
 **Warning**	Provides information that can make or break your project.

Welcome to CueStation

1



With the use of our Matrix[3] audio system automation hardware and CueStation programming software, you possess the industry's leading sound control solution. No other system provides as much flexibility, control, and reliability for such a wide range of applications.

CueStation is the cue-programming component of the Matrix[3] system. It offers a graphic user interface that emulates a familiar mixing console and couples it to a cue-based automation system that is designed to integrate easily into theatrical, theme park, and other show control situations.

The LCS Matrix[3] uses the latest in digital signal processing and networking technology. With a modular system architecture the system is easily configured to match an application. CueStation runs under the BeOS operating system.

What's New in CS Version 3.0

Perhaps the most obvious improvement to CueStation, from a user perspective, has been to the user interface. Innumerable changes have been made in an effort to provide an intuitive, full-featured, and efficient automation programming system. A new system configuration interface makes system setup quite intuitive. Capturing and updating subcues and cues has become easier, and you can now sort the lists as desired.

CueStation v3 now fully supports Matrix^3 hardware. A new "Configurator" window makes system setup easy: you choose a frame and select the cards which are installed in it. The configurator will only allow valid module combinations and provides an estimation of dsp usage, so that you can avoid over-loading any one frame.

There is a new window, System Status, that provides real-time monitoring of each frame and dsp. Frame temperature, link status, power supply voltage, control latency, and dsp usage are displayed for each frame.

The Generate Report command has been expanded. It provides a text file that reports the mixer configuration, cue contents, a listing of externals and bus trajectory comments, and the thumbnails fields of all subcues. Thumbnails are automatically generated when a subcue is created and provide a shorthand description of the control point settings involved. These expansions to the report serve not only to aid in documenting the project file, but also to assist in troubleshooting and design review.

Request Mixer Settings is a new command that allows you to verify that the hardware control points match those displayed in CueStation.

Our innovative SpaceMap spatial panning software has been significantly improved: you can now edit a trajectory path, adjust the time offsets for each node, and create subcues more easily.

The Equalizers now support configuration with a variable number of bands for each set of eight inputs or outputs. Maximum delay times can be set for each set of modules.

Other improvements include on-line help, ToolTips help, automatic project backups, password protection, and window management improvements.

LCS remains committed to the continuing development and evolution of our software and hardware. We will always listen to suggestions from users.

About the Help Guides

CueStation is a comprehensive and exceptionally flexible cue programming system, and must cope with a virtually unlimited number of Matrix[3] configurations. To ensure that our customers have the best possible experience with our products, we provide classroom-based training for new users, and support via e-mail and telephone. On-site service is an available option.

This Users Guide does not replace the training program. Rather, it is your pre-course reading material, classroom textbook, and post-course reference. Various sections of the document address the intricacies of CueStation through different approaches: specific task outlines, conceptual overviews and analyses, and command references.

CueStation documentation includes:

- The **User Guide.** There are articles that will help you understand how CueStation works and how to use it effectively, a task reference that sketches-out the basic use of the software, and a Commands Reference that provides detailed information for each command.
- **Online Help:** adapted from the user guide, our online help provides you immediate access to descriptions for every command available in CueStation.
- **ToolTips:** these are pop-up help boxes that appear when you “hover” the mouse over buttons, data entry boxes, and controls. They provide abbreviated help — perhaps just enough to help you remember their function without resorting to this Users Guide.

Tip

In our continuing efforts to create the most powerful and useful software control system for the Matrix[3] system, we look for new and better ways to document our products. If you have suggestions for improvements to the documentation, please let us know.

Level Control Systems
Documentation Department
130 East Montecito Ave #236
Sierra Madre, CA 91024
USA
Tel: +1 (626) 836-0446
Fax: +1 (626) 836-4883
http://www.LCSaudio.com
support@LCSaudio.com

User Guide Conventions

Style and language conventions for this guide are provided in the front matter, page *x*.

Using Online Help

This user guide is available in online format. Open the **LCS** menu and choose **Help.** You may be offered a choice of languages and/or chapters. The help document will be shown using your default web browser; we suggest using the Opera browser (http://www.opera.com) for the best results.

Using ToolTips

If you have enabled ToolTips help, simply point the mouse cursor at the button, data entry box, or other control that you're curious about, and wait a moment. If ToolTips help is available for the object, it will be displayed in a pop-up box.

To enable or disable ToolTips help, open the **LCS** menu and choose **ToolTips Enabled.** This toggles the ToolTips setting. If ToolTips are already enabled, a checkmark will be shown beside the command, and choosing the menu item will disable ToolTips help. If there is no checkmark, choosing the menu item will enable ToolTips help.

About Level Control Systems

Level Control Systems is recognized across the world as the leading manufacturer of high-end sound control hardware and software for theater, themed entertainment, and major theme parks worldwide. The company was founded in 1992, and LCS automation quickly became the preferred system in Las Vegas, Broadway, and major theme parks.

LCS Canada is the principle location for hardware design and manufacturing and is located in Vernon, British Columbia, Canada. Software development, sales, and support is primarily handled by the LCS Office in Sierra Madre, California, USA.

Worldwide sales of LCS products are handled through our network of dealers and regional distributors. Many of our key dealers are accomplished sound designers in their own right, and with hundreds of installations throughout the world, pre-sales demonstrations may be arranged.

Worldwide Sales Headquarters

LEVEL CONTROL SYSTEMS
130 East Montecito Ave #236
Sierra Madre, California
91024, USA
Tel: +1 (626) 836-0446
Fax: +1 (626) 836-4883
http://www.LCSaudio.com
info@LCSaudio.com

LCS Service and Support

Level Control Systems is absolutely committed to providing our customers with the highest-quality technical support. Most simple questions can be answered quickly via e-mail or telephone support. Extended on-site service and support is available. Please contact LCS for current rates and availability.

E-mail Support

LCS provides e-mail support services at support@LCSaudio.com. We are committed to responding to e-mail support incidents within one business day.

Telephone support is available through the LCS office at +1 (626) 836-0446. We are open Monday–Friday, standard business hours (Pacific Standard Time).

Web Site Support

The web address for LCS products on the Internet is http://www.LCSaudio.com. At this location, you can download the latest versions of software and obtain documentation in PDF format.

Training Program

Purchase of CueStation entitles you to attend a training program. LCS conducts training classes at its office in Sierra Madre, California, a short distance from Los Angeles. The standard course is three days long. If you are interested in training please contact us by telephone at +1 (626) 836-0445 or by e-mail, support@LCSaudio.com.

Before Calling Support

Before calling LCS Technical Support, please have the noted information available. This will help us help you more quickly and effectively. If you're dealing with a system crash, be prepared to send some debugging files to us, as detailed in *Common Tasks: Collecting Files for Troubleshooting Assistance,* page 5-54.

Before calling Support, please have the following information available:

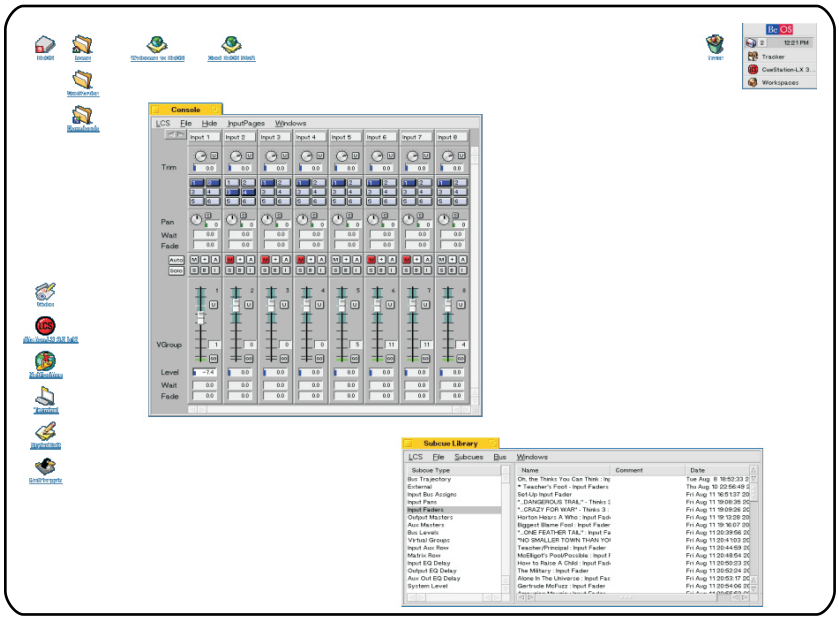
- A brief description of the problem.
- The components used in your Matrix^3 system.
- The firmware version used in your Matrix^3 system.
- If the problem is with a CueStation crash:
  - The last three CueStation log files.
  - Your CueStation project file.
  - The software versions of your installation of CueStation and Helper Apps.

Using BeOS 2

The Be Operating System is a modern operating system, designed to provide superlative real-time multimedia support. LCS has chosen to use it for its stability, raw speed, and support for high-quality multitasking. CueStation requires BeOS version 5.0.3.

Note

Visit Be's website for general information, patches, support, and a comprehensive list of BeOS- compatible hardware.
http://www.be.com



BeOS For New Users

If you are running a standard BeOS release 5 installation, there is an excellent users guide already on your system.

To access this documentation:

1. Click the BeOS button on the Deskbar.

2. Select the **Applications** submenu, and choose **NetPositive.**
3. Type file:///boot/beos/documentation/User's Guide/index.htm in the **address** box.

An alternative address, if you have access to the Internet, is http://www.be.com/documentation/User's Guide/

If you cannot locate the BeOS User's Guide, contact us: we'll help you install it, or send you an electronic copy.

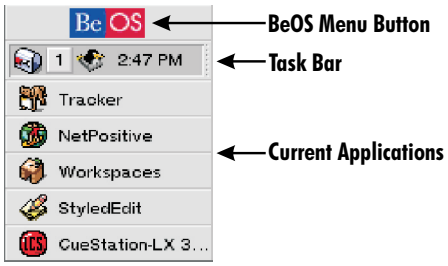
Unique BeOS Features

The BeOS graphic interface is similar to those used by other operating systems. If you have previously used Microsoft Windows, Mac OS, or one of the Unix GUIs, you'll find that BeOS is generally familiar.

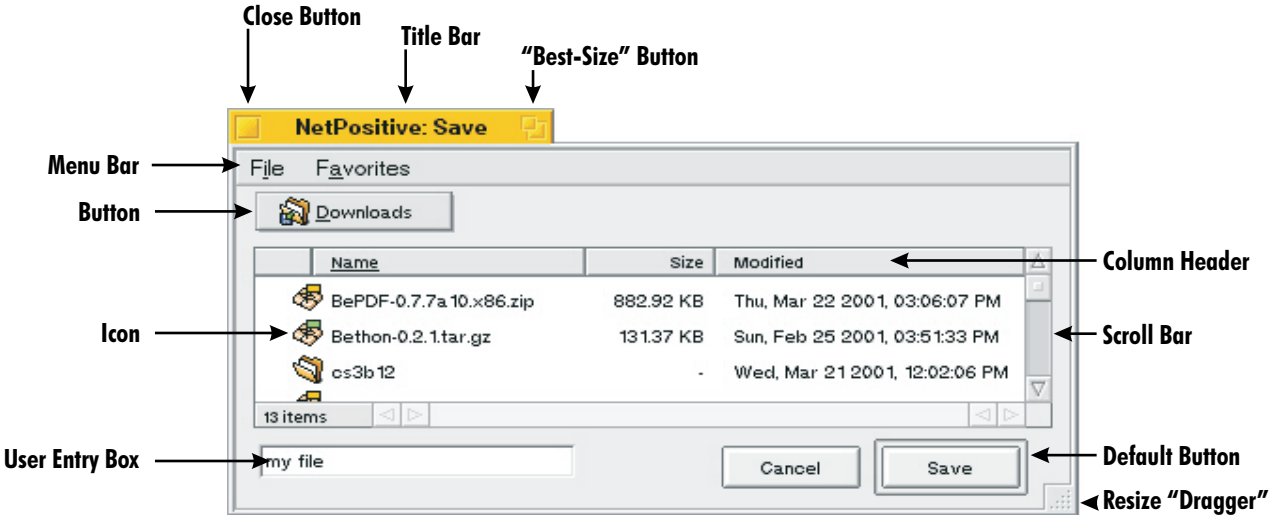
The following subsections provide information for the parts of BeOS that may be new, unusual or unfamiliar to you:

The Graphical Interface

The BeOS Deskbar has a BeOS button that opens a menu; a status bar that typically contains some Add-On shortcuts and the time; and a list of open applications. Right-clicking an application name opens a menu that provides some window commands. Clicking the dotted line beside the task bar will shrink or expand the Deskbar.



A typical BeOS window will contain the following controls. Most of the controls work as expected. It's worth noting that you can reposition the title bar by holding the SHIFT key while dragging it, and that double-clicking the title bar will hide the window.



Workspaces

BeOS provides the virtual equivalent of nine separate computer monitors (displays).

With any graphic interface, there is a limited amount of screen "real estate" available. For a complex application like CueStation, some twenty-odd windows can be displayed at one time — far more than can fit into a single screen.

One solution would be to have multiple computer displays. However, such a solution requires a lot of space and is prohibitively expensive.

BeOS provides a solution by using *workspaces*. Workspaces are "virtual displays" — basically, a series of individual computer displays, all shown through your one computer monitor. You choose which workspace to view at any one time, and can move windows from one workspace to another.

There are graphic and keyboard controls for the workspaces.

To use the graphic interface, use the Workspaces tool. Click the BeOS button on the Deskbar, select the **Preferences** menu and choose the **WorkSpaces** command. A small window will be displayed, similar to the one shown to the left. Click on the "panes" to change to that workspace; or drag the miniature windows from one workspace to another.

Alternatively, you can use ALT+F1 through ALT+F9 to change views among the nine default workspaces. To move a window from one workspace to another, simply point at the window title bar, hold the left mouse button down, and press the ALT+F*n* key for the workspace you want to move it to.

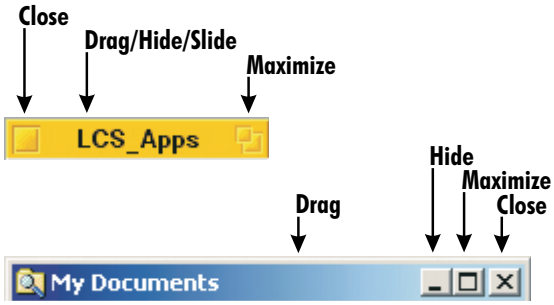
Differences from Microsoft Windows

The single biggest hurdle for the experienced Windows user is to use the ALT key for most shortcuts, instead of the CTRL key. There are other differences, too, as listed below:

GUI Differences

The Deskbar provides the equivalent of the Windows task bar, with the BeOS button acting similarly to the Windows Start button.

The Windows and BeOS window title bars are similar, mainly differing in their placement of window controls.



Keyboard Differences

CTRL versus ALT

BeOS uses the ALT key for the majority of keyboard shortcuts, whereas Windows uses the CTRL key. For instance, you use CTRL+C to copy a selection in Windows, but ALT+C in BeOS.

If you don't like using the ALT key for shortcuts, you can change it to CTRL. Refer to *Alt versus Ctrl,* page 2-5, for details.

Note

This Users Guide maintains the BeOS convention of using ALT as the shortcut key. If you choose to use the CTRL key, you will need to mentally reverse the two as you read our instructions.

ENTER, TAB and SPACE in dialog boxes.

BeOS and Windows behave very differently when using keyboard shortcuts in dialog boxes. In Windows, the ENTER key and the SPACEBAR do the same thing: they “click” the *highlighted* button. You can use the TAB key to move the highlight from button to button.

But in BeOS, the ENTER key is *always* equivalent to clicking the *default* button: it doesn’t matter if you move the highlight. The SPACEBAR is used to press the highlighted button. The ALT+TAB key combination moves the highlight (shown as a faint underline).

Unlike Windows, if you move the highlight from the default **OK** button to the **Cancel** button, and then press ENTER, you have just chosen to “ok” the operation instead of cancelling it! Pressing the SPACEBAR would have been the correct action.

File Management Differences

In BeOS, storage devices must be *mounted* before they can be used. The boot hard drive is automatically mounted; for other drives — secondary hard drives, floppies and CD-ROMs — you must mount them. See *Mounting Devices,* page 2-8, for details.

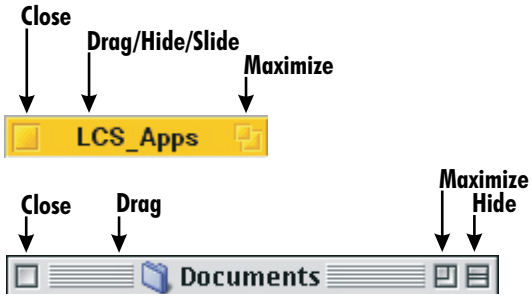
Differences from Mac OS

Experienced Mac users who are used to the single-button mouse will need to adjust to using a two-button mouse. There are other differences, too:

GUI Differences

The Deskbar is a bit like the new Mac OS X Dock, but with more functionality. The BeOS button acts similarly to the Apple menu.

The Mac and BeOS window title bars are markedly similar.



Mouse Differences

BeOS makes extensive use of the two-button mouse. In this Users Guide, whenever the right-hand button is used, it is explicitly identified. If a left/right distinction has not been made, use the left-hand button.

Keyboard Differences

OPTION versus ALT

BeOS uses the ALT key for the majority of keyboard shortcuts, just as Mac OS uses the CMD key. In both cases, the key is immediately beside the SPACEBAR, so you probably won't have to re-learn your habits.

ENTER, TAB and SPACE in dialog boxes.

BeOS and MacOS behave very differently when using keyboard shortcuts in dialog boxes. In Mac OS, pressing the ENTER key is the equivalent of clicking the *highlighted* button. You can use the TAB key to move the highlight from button to button.

But in BeOS, the ENTER key is *always* equivalent to clicking the *default* button: it doesn't matter if you move the highlight. The SPACEBAR is used to press the highlighted button. The ALT+TAB key combination moves the highlight (shown as a faint underline).

Unlike MacOS, if you move the highlight from the default **OK** button to the **Cancel** button, and then press ENTER, you have just chosen to "ok" the operation instead of canceling it! Pressing the SPACEBAR would have been the correct action.

ALT+*Arrow* in windows

BeOS uses the arrow keys extensively, both with and without the ALT key. You can frequently use the arrow keys to adjust controls, choose from menus, and perform other actions. Experimentation is the best way to discover what can be done.

Customizing BeOS

There are several things you can do to make BeOS easier to work with. Perhaps the most important are:

- To make the keyboard work the way you expect it to.
- Increase your screen resolution so that you have more working space.

Accessing Preferences

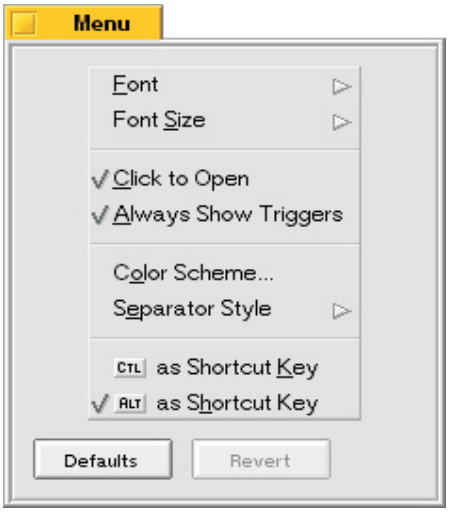
BeOS preferences are most easily accessed by clicking the **BeOS** button on the Deskbar, and then choosing from the **Preferences** submenu.

Alt versus Ctrl

BeOS uses the ALT key as its main shortcut key. While this makes sense for Macintosh users (who are used to using a shortcut key that is immediately beside the Spacebar), it can be difficult for Microsoft Windows users, who are more familiar with using the CTRL key.

Fortunately, BeOS allows you to choose which key to use as a shortcut key:

1. Click the **BeOS** button on the Deskbar, open the **Applications** submenu and choose **Menu.**
2. Choose your shortcut key preference.



Increasing Your Screen Resolution

You can often use a higher screen resolution than BeOS assigns by default. However, it is worth noting (a) that your maximum settings are dependent on the capabilities of your video card and monitor, and (b) that the maximum setting may not be optimal.

At the some resolution settings, you may find that the display becomes fuzzy, flickers, or is difficult to view. Because it can be difficult to use the system when these conditions occur, BeOS uses a time-out when you change resolutions: if you don't agree to use the new setting, the old setting will be re-applied. The time-out is approximately fifteen seconds.

If through some mistake a setting is used that is incompatible with the display, you can boot BeOS in "safe mode" by pressing the spacebar during start-up and following the on-screen prompts. This will provide you the opportunity to correct the problem.

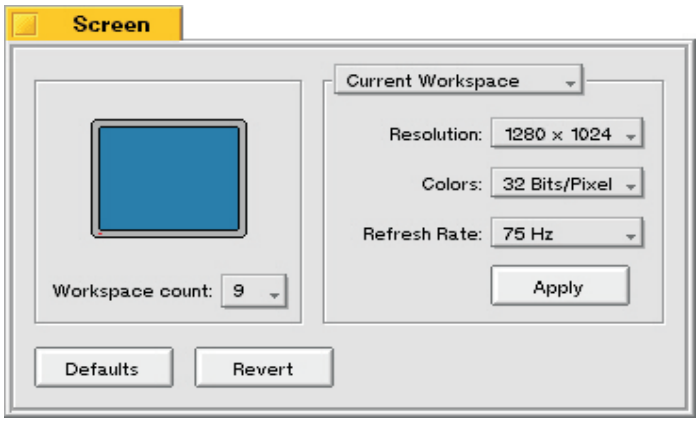
Warning

Most modern monitors will protect themselves when settings are too high. Older monitors, however, can be damaged by video signals outside their working range.

To change the display resolution:

1. Click the **BeOS** button on the Deskbar, open the **Applications** submenu and choose **Screen.**
2. Click the **Current workspace** button on the top right and choose **All workspaces.**
3. Choose the screen resolution you'd like to use. If you have a 15" monitor, we recommend using 800×600 resolution; if you have a 17" monitor, try the 1152×864 resolution; and if you are using a 19" monitor, you may find 1280×1024 works well. Note that CueStation requires a minimum 800×600 resolution.
4. Choose the color support: we suggest 16- or 32-bits/pixel, for the best color support.
5. Choose a refresh frequency between 60Hz and 90Hz. If the available values are below 60Hz, choose the next lowest resolution and try again. The lower the refresh frequency, the more likely it is to flicker and cause eyestrain.
6. When you click **Apply**, BeOS may recommend that you check your monitor user's manual to ensure that the settings you have chosen will not damage the monitor. **Please heed this advice!**

   You are then presented a "Revert"/"Keep" dialog box. If you do not choose "Keep" within fifteen seconds, your previous settings will be re-applied.



Changing the Desktop Color

It's easy to become lost when moving between workspaces: they all tend to look alike. (See page 2-2 to learn how to use workspaces.)

To help alleviate this, you might want to change the color of the desktop for each workspace:

1. Click the **BeOS** button on the Deskbar, open the **Applications** submenu and choose **Backgrounds.**
2. Select a unique background color for the workspace.
3. Point at the **Backgrounds** title bar, press the left mouse button, and use one of the ALT+F1 through ALT+F9 commands to change another workspace's color.

Mounting a Device

Your file storage systems — hard drives, floppy drives, CD-ROMs and such — are called *volumes* in BeOS. By default, BeOS only shows its own hard drive volume on the Desktop (normally shown as a hard drive icon located at the top left of the screen, labeled “BeOS”).

To gain access to other volumes, you have to *mount* them. When you mount a volume, its icon will appear on the Desktop, where you can double-click it and use the BeOS Tracker to manage its files. It will also become accessible through **Open** and **Save As** dialogs, so that you can transfer files to and from CueStation.

To mount a volume:

1. Right-click the Desktop. Open the **Mount** submenu and choose the volume you wish to mount. Previously mounted volumes have a checkmark beside their name.
2. A few seconds after mounting the volume, its icon will appear on the Desktop. You now have access to it.

Warning

The BeOS Tracker does not automatically detect a floppy disk change. If you eject a disk and insert a new one, you will need to remount the floppy volume by unmounting and mounting it. The automount setting doesn’t help with this, unfortunately.

To unmount a volume:

You can remove access to a volume by “unmounting” it: simply right-click the volume icon and choose unmount (ALT+U). You cannot unmount the main BeOS volume, as that would crash the system.

To use automatic mounting:

You can choose to have volumes automatically mounted when the system starts up, by opening the Mount menu and choosing **Settings.**

Tip

We suggest using pre-formatted IBM-style floppy disks. While BeOS can format a floppy disk, the disk will not be readable on Microsoft Windows machines. The IBM-style disks are readable by Windows, Mac and BeOS systems.

Accessing Files

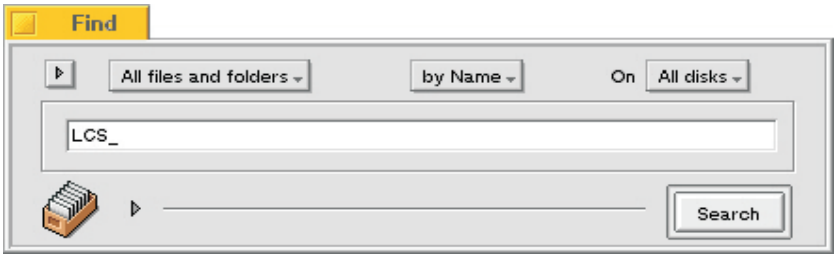
BeOS uses a “file folder” model for its filing system. Conceptually, your volumes (hard drives, floppy disks and CD-ROMs) are treated as a manila file folder, of the variety used in filing cabinets. A folder can contain more folders, as well as files, and this pattern can be repeated indefinitely.

You’ll access your files through the Find command, the File dialog, the BeOS Tracker, or the Desktop.

Using the Find Command

The quickest and easiest way to find a file is to use the File Finder.

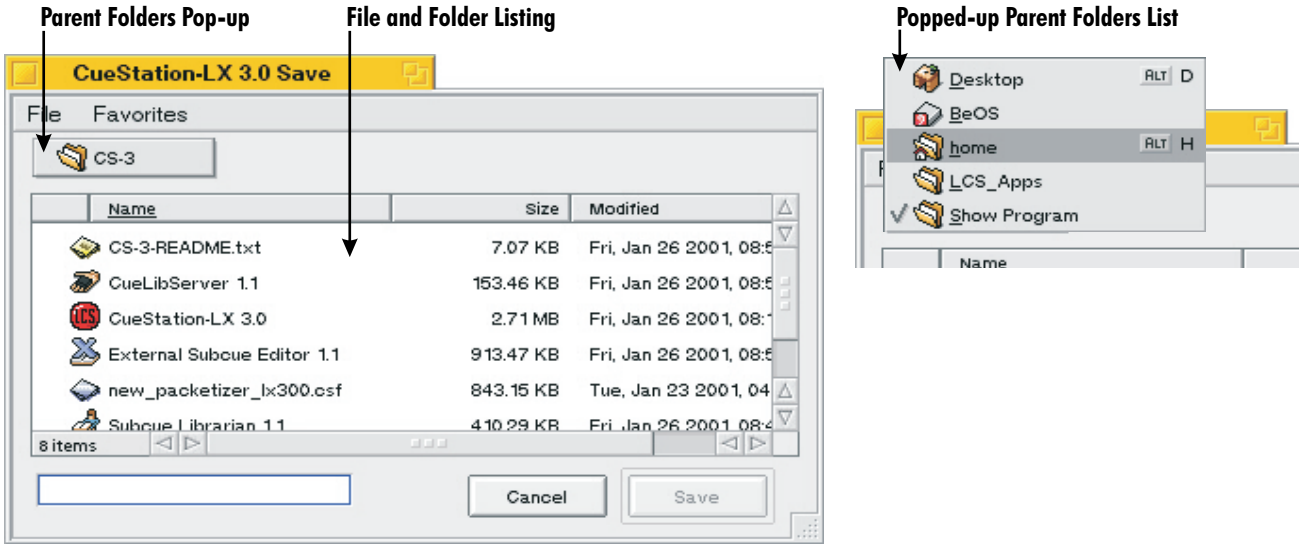
1. Click the **BeOS** button on the Deskbar and choose **Find.**
2. Enter your search term. The Find command is quite powerful. It is worth spending some time experimenting with its various options.
3. The Tracker will show a list of files that match the search terms you used. You may double-click files to open them, or right-drag them to copy or move them. See *Using the BeOS Tracker* section for more information.



Using the File Dialog

If you already have an application open, the easiest way to access the files it creates is to use the **Open** or **Save As...** commands. These commands will show you the file dialog.

The largest part of the dialog is given over to the file and folder listing.



When you click the **parent folder** button, it displays a hierarchical list of folders. You can move directly to any of these folders, and then use the file and folder listing to "drill down" to the folders and files contained inside it.

If you need to access a different volume (hard drive, floppy drive or CD-ROM), choose Desktop (at the top of the list) and then choose the volume. Volumes must, of course, be mounted before they can be used. Refer to *Mounting Volumes,* page 2-8.

Double-clicking a folder will open the folder, showing a listing of that folder's contents (and placing the folder name on the pop-up hierarchical menu).

Tip

As you can see in the pop-up folder list, the ALT+H and ALT+D shortcuts take you directly to the Home directory and Desktop directories. ALT+ARROW moves up and down the list.

Opening Files

Double-clicking a file will open the file in the current application. If the application doesn't know how to deal with that type of file, it will report an error.

Saving Files in the File Dialog

Double-clicking a file listed in the file listing will save a new file using the existing file's name: *this will overwrite the existing file,* though you will be given an opportunity to cancel the command.

To create a new file, type the file name in the file name entry box below the file and folders listing. If this name is unique — not listed in the file listing — then a new file will be created.

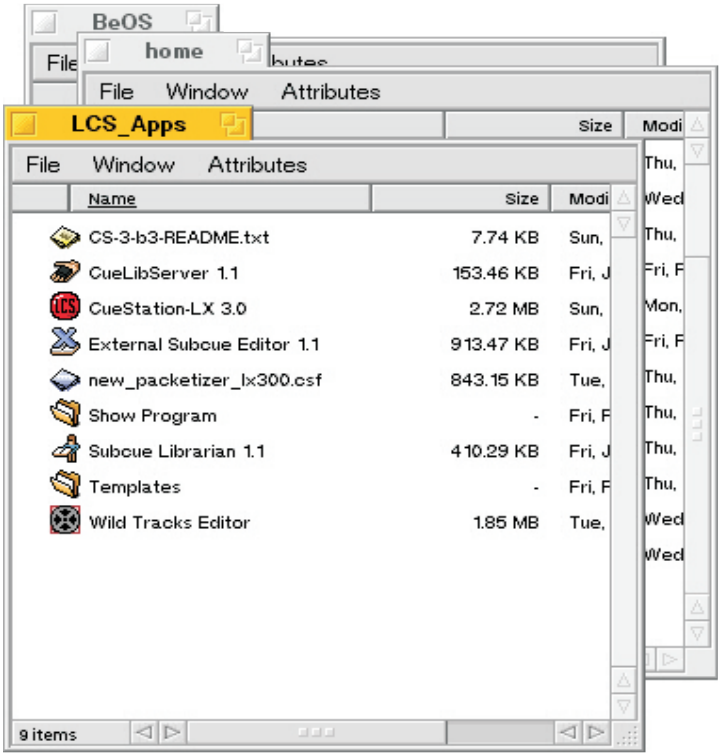
Using the Tracker

The Tracker provides a view of the folders and files. You can single-click a file or folder to select it; right-click a file or folder to access a pop-up menu of commands; or double-click the file or folder to open it. When you open a folder, a new Tracker window will appear, showing the contents of the folder. When you open a file, BeOS will start the application that can handle that type of file.

You can “drill down” to a folder or file by opening a volume or folder that’s shown on your Desktop, and repeatedly opening the folders that are listed, until you reach to your destination folder or file.

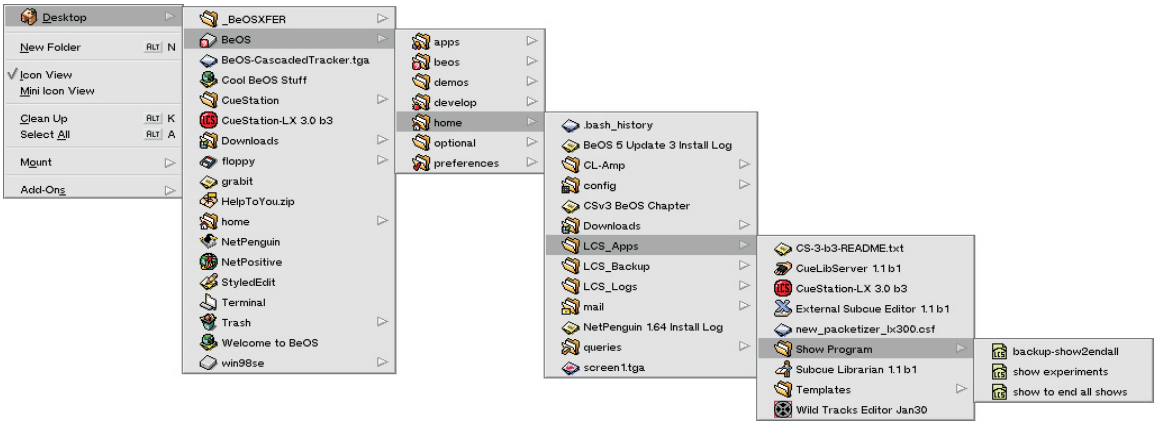
Opening a File

When you double-click a file that is displayed in the Tracker, it will be opened by an application that can use it. If BeOS doesn’t know which application to use, it will ask you to select one.



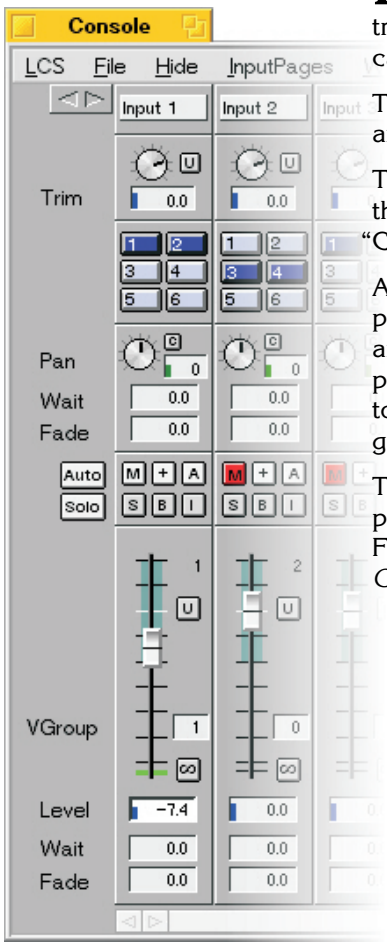
Using the Desktop

Alternatively, you can right-click a volume or folder on the Desktop, and use the menus that appear beside its name in the pop-up menu. The folder or file you choose through the submenu will be opened directly.



Understanding CueStation

3



In the LCS Matrix[3] Training Course, we find that the most common challenge to new CueStation users is in expanding their thinking beyond the limitations of traditional physical consoles. While CueStation presents a familiar interface, it is capable of far more complex sound control than has been previously possible.

The following articles will introduce you to the rationale of CueStation's design and structure.

The first few articles are particularly valuable to those users who are about to take the CueStation training course: reading them will help you get the most from "CueSchool."

All the articles will prove valuable as you start your first CueStation automation programs, from overall show design to actual Cue programming. Refer to the articles while you design the show, so that you can use the system to its full potential. While you are programming the show, refer to them whenever you seem to be doing things the hard way, or can't figure out how to accomplish a design goal.

The material is directed toward presenting ideas, rather than step-by-step procedures, so that you can apply the knowledge to your own unique situations. For step-by-step procedural instructions, refer to the *Common Tasks* and *Command Reference* chapters.

CueStation and the Matrix³

In this article, you will learn how CueStation uses the familiar signal path of a traditional mixing console as the basis for a complete sound control system with time-based dynamic mixing and cue list automation. We will examine the signal path and its control points, and the hierarchical organization of CueStation's cue list automation system. You will learn about the relationship between CueStation and its family of Helper Applications, and will see how CueStation communicates with and controls the functions of the Matrix³ hardware.

The Software Interface

CueStation uses a graphic representation of a physical mixing console to present a familiar and user-friendly interface to the powerful digital signal processing inside the Matrix³ hardware. When using CueStation, you will work with familiar controls: faders and knobs, bus assignment switches, and value entry boxes. You will also have the option of using more powerful controls that have no physical console equivalent: SpaceMap control of moving sounds, cue list automation, Wild Tracks hard disk playback, and VRAS room simulation.

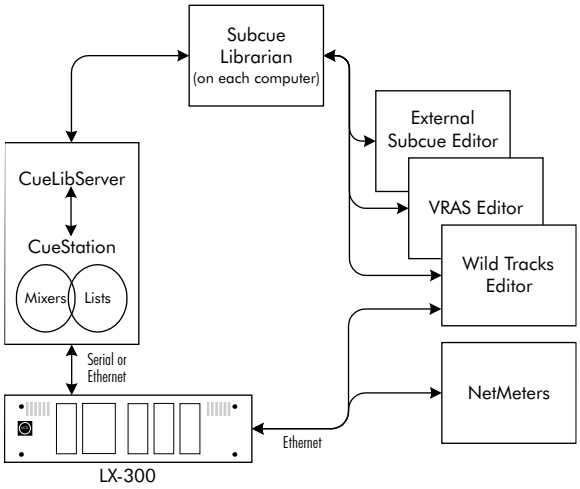
CueStation provides the presents the image of a single, monolithic mixing and processing engine. In reality, the Matrix³ system contains from one to thirty-two LX-300 processor "frames." Each frame in a system can have some combination of inputs and outputs plus audio playback and processing options, and each frame holds a digital signal processor (DSP) that handles the processing of audio for that frame's functions. These independent frames are digitally interconnected so that audio and control data can pass freely among them, and this permits CueStation to treat the entire system as a single unit.

In most installations, the Matrix³ hardware will be installed in a location remote from the computer that runs the CueStation software.

CueStation Software Architecture

CueStation v3.0 represents a transitional stage between two software architectures. The earlier version was similar to many legacy applications in that all functions were embodied in a single, large program. Future versions of CueStation will be completely modular, consisting of a collection of smaller, specialized applications that interact in a client-server relationship and can operate over a network to allow several sound programmers to work together.

The computer running CueStation connects directly to the LX-300. This can be via serial line or by Ethernet. The Helper Applications connect to the CueStation application via CueLibServer. CueLibServer is a specialized application that runs in the background and handles the exchange of data between CueStation and the Helper Apps. Subcue Librarian provides the user interface to CueLibServer.



When using a network connection (ie. Ethernet), the Helper Apps (External Subcue Editor, Wild Tracks Editor, and VRAS Editor) do not need to run on the same computer as CueStation and CueLibServer. However, each computer running a Helper App must also be running its own copy of Subcue Librarian.

NetMeters is a separate application that communicates directly with the LX-300 over Ethernet. This requires an LX-ELC EtherTracks card.

The EtherTracks card also has a built in web page that provides status information. Any web browser can connect via EtherNet to this card.

CueStation proper divides its functions into two main areas: a mixer programming section with a graphic user interface that looks like a conventional mixer control surface, and a list-based automation section that supports the creation and organization of cues and cue lists. The various control surfaces and automation lists are available through separate windows, so that you can mix-and-match your "workspace" to suit the task at hand. In addition to the CueStation windows, you have access to the Subcue Librarian and other editors, as well as system status and log windows.

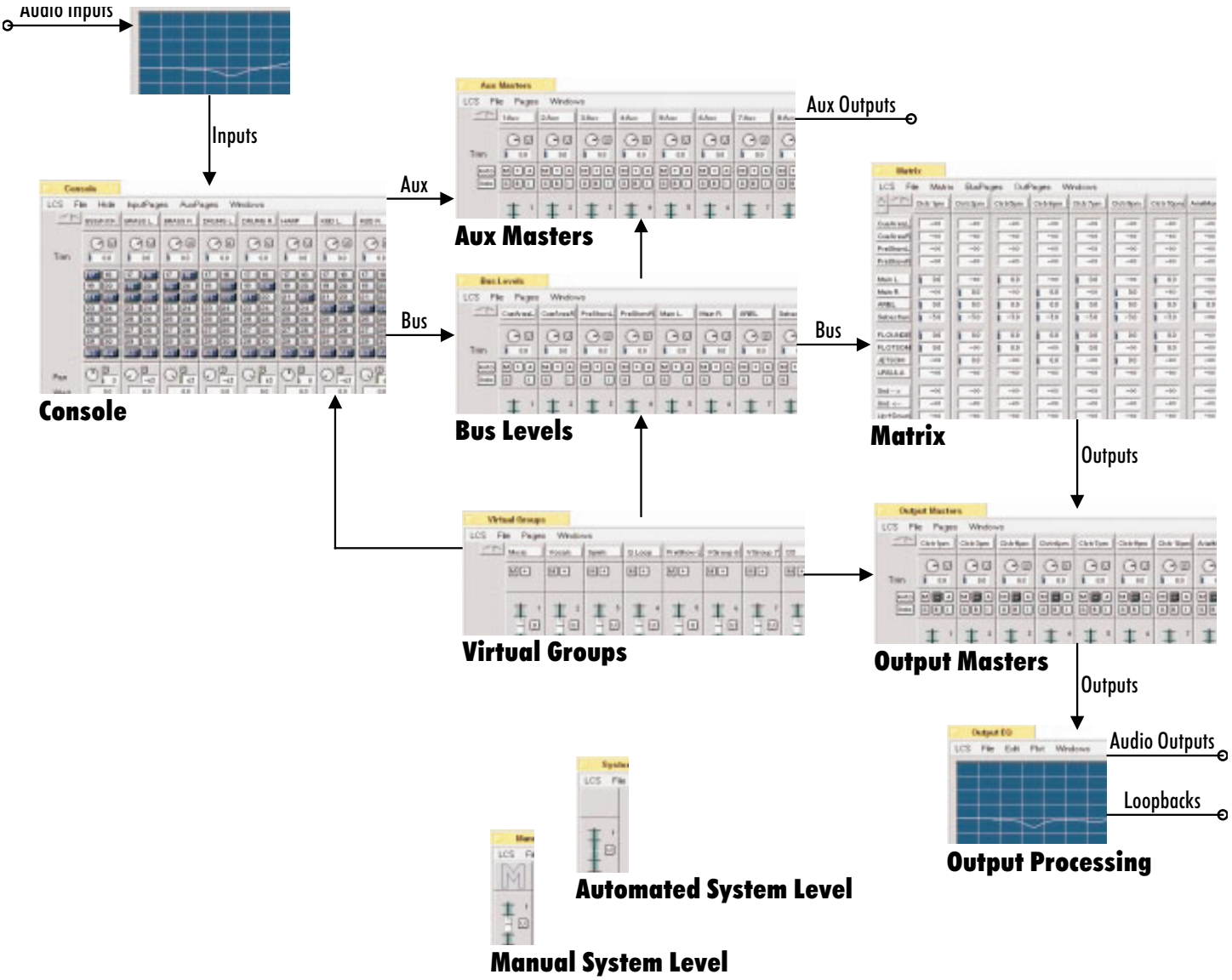
Signal Flow through CueStation

From your viewpoint — that of a CueStation programmer — the Matrix³ looks like a configurable mixer with inputs and outputs, equalization and delay processing, and (optionally) Wild Tracks hard disk playback and VRAS reverberation subsystems. The inputs can be divided between external inputs (analog, ADAT lightpipe, CobraNet), Wild Tracks playback channels, and VRAS returns. The outputs can be divided between regular outputs (connected to the amplifiers and loudspeakers or the main sound system) and auxiliary sends (connected to effects processors and/or monitor systems).

Between the inputs and outputs is a flexible, matrix-based, signal-routing system with multiple control points for dynamic mixing and distribution of the signals. There are two alternative signal paths through this system:

- The Matrix mix signal path passes through the Input fader, Input Trim, Pan, Bus Level, Bus Level Trim, Matrix, Output Master, Output Master Trim, System Level, and Manual System Level.
- The Auxiliary mix signal path passes through the Input fader, Input Trim (if the Aux is in Post-fader mode), Aux Level, and Aux Master.

Both paths provide a full complement of control points, as described in the next section.



Control Points in CueStation

A “control point” in CueStation is defined in a general sense as any parameter that can be adjusted by the user, either manually or by the automation system. Some control points address signal processing parameters, such as equalization and delay, that are seldom changed throughout a show. Other control points, such as Automate and Isolate, affect the operation of the automation system itself.

Most of the automated control points are associated with fader functions in the user interface, and these are always grouped in threes: Level, Wait, and Fade. The “Level” parameter is the target value for gain scaling. “Wait” is the time delay between the moment a cue is recalled and the fader starts to change value, and “Fade” is the length of time it takes for the fader value to change continuously from its current value to the target value set by the Level parameter.

For signal paths that pass through the Matrix, there are five automated level control points: Input Fader, Bus Level, Matrix, Output Master, and System Level. If you are using the auxiliary send signal path, there are two or three control points: Input Fader (in Post- but not Pre-fader mode), Aux Send, and Aux Master.

The values of arbitrary groups of fader controls (Input, Bus Level, Output Master, or Aux Output Master) can be additionally scaled by Virtual Group Faders. These "meta" controllers reside in the Virtual Groups window. VGroups can be used like the VCA Groups in a conventional console, but VGroups are much more flexible because a single VFader can control several types of control point.

Besides the automated level controls, there are manual trims available at Input, Bus Level, Output Masters, Aux Masters, and System Level. These controls are used to adjust the relative levels of the automated mix during a performance, without interfering with the overall contour of the automation. There are, of course, no wait or fade times associated with a manual control.

The remaining two automated control points in a primary signal path are Bus Assignments, which have an on/off state for each bus, and Pans, which have Level, Wait, and Fade parameters. The virtual pan defined in CueStation performs similarly to a physical pan pot with its pair of potentiometers scaling the two legs of one split signal. It can be thought of as another gain stage in the virtual signal path: Input Fader, Input Trim, Pan, Bus Level, Bus Level Trim, Matrix, Output Master, Output Master Trim, System Level, and Manual System Level.

In addition to level control and signal routing, EQ and delay processing are available for each external input and output. Separate EQ editing windows contain control points for setting the filter Type, Gain, Frequency, and Q for each EQ passband, and for setting delay time.

Six additional types of control points affect either the continuity of the signal path or the behavior of the automation control at various points: Mute, Polarity, Automation Solo (all three automated), Solo, EQ Bypass, and Isolate (all three non-automated). Mute and Solo function as they would in a conventional console: Mute shuts off the signal on any channel where it is selected, Solo shuts off the signal on any channels where it is not selected. Polarity inverts the signal waveform. Bypass turns off the EQ for an input or output channel. Automation Solo enables or disables automation for a channel; it follows similar logic to an audio solo function (if Automate is not selected on any channel, then all channels are automated; but if it is selected on any group of channels, then only the selected channels will be automated.) Isolate removes a channel from automation control.

Cue-Based Automation

The Automation section of CueStation is where all the control point settings are stored and organized for recall during performance, using a cue-based snapshot scheme. It has six windows that allow the creation and control of cue lists containing control data for all mixer functions and for control of external devices. The design of the cue structure is hierarchical, and is designed to be flexible and easily tailored to different programming styles. It is in many ways akin to cue control systems in use throughout the entertainment industry for lighting and device control, and is commonly interfaced with such systems in theatrical, theme park, and similar venues.

Related sets of control point data (such as Input Faders or Matrix Rows) are saved as data structures called "subcues," and collections of subcues are in turn grouped into "cues." A cue thus represents the behavior of all or part of the

Matrix³'s control point parameters at a particular moment. A sequence of cues is organized as a "cue list.".

CueStation stores discrete states, target values, and time intervals, and it calculates the continuous transformations "on the fly." Such efficiency is essential for large configurations. The flexibility of cue automation is also critical in live performance, where timing and order can be indeterminate.

The CueStation data hierarchy, from the top down, is as follows:

- Project file: All setup data defining the mixer and user interface configuration, together with one or more cue lists plus libraries of cues and subcues containing real-time control data for the mixer and any external devices in the show control system.
- Cue Lists: Ordered collections of cues allowing either manual or automated stepping through of a pre-programmed sequence of events.
- Cues: A collection of control data designed to be triggered as a group. May contain data of one particular type or of many different types.
- Subcues: A collection of control data related to specific subsystems of the mixer or external devices.
- Control points, commands and related data structures: Individual control messages, most commonly addressing the parameters of external devices, and residing inside External subcues.

Cues and cue lists may be created in two ways: "top-down" and "bottom-up." In the top-down approach, the current state of the entire system can be captured into a new cue at any moment with a single keystroke. This is a quick way to "rough out" a mix during a rehearsal, and during subsequent iterations the settings can be as quickly adjusted and updated. The bottom-up approach lends itself to an off-line working situation; it consists of the creation of individual subcues and the assembly of these into cues and cue lists.

Simulation and Synchronization

During normal operation, the combination of CueStation software and the Matrix³ hardware will perform seamlessly, with the appearance of full integration among its component parts. The hardware will respond instantly to commands issued from the CueStation computer, a CueConsole or CueMixer control surface, or from an external show control system. CueStation's graphic user interface will display all dynamic changes in the state of mix architecture as they happen, and the Log window will promptly print pertinent information about the recalling of cues and other processes.

In reality this is an illusion, though a necessary and benevolent one. Because a Matrix³ system can have a great many dynamic processes acting simultaneously within its signal processing architecture, it is essential that the automation control be embedded in the LX-300 hardware itself. This ensures efficiency and robustness, but it requires a partial "decoupling" of the DSP processes from the visible user interface on the CueStation computer.

What is shown in CueStation during the running of a show is in fact a simulation of what is happening in the hardware. The embedded automation in the hardware and the animated user interface on the computer both work independently, with re-synchronization happening only when cues are triggered from the computer.

Most of the time the communication between the computer and LX-300 Frames will be intermittent, taking place only when a cue is fired from CueStation or when

the Matrix³ sends a status message. The motions of graphical controls are a simulation of what is going on inside the Matrix³'s mix engine. Practically speaking, this means that two copies of the current Project are always operating in tandem: one inside the Matrix³ and the other on the computer.

Every time a cue is fired, both systems move through the dynamic processes described in the cue data. The on-screen faders will move at the same time, and with the same rate, that signal levels are varied in the LX-300. Time code will be generated in the LX-300 and cues will be triggered, while CueStation displays the changing time code values and steps from cue to cue in its Transport and Library windows.

This simulation is necessary because the serial connection between the computer and the Matrix³ has an insufficient communications bandwidth to transmit all the real time processes as they happen. The simulation is dependable because both the BeOS and the LX-300's embedded automation system have been designed with multiple, concurrent real time processes in mind.

Having said that, it is important to add that "dependable" does not mean "100% perfect" and occasionally there may be a loss of synchronization. Such a situation might occur due to a momentary communications dropout or data corruption due to noise in the serial connection, or it could result from a power outage that affects part of the system. It might also result from corruption in the Project file itself, or even in some cases during programming when some editing changes seem not to "take" properly.

Loss of synchronization becomes apparent when the on-screen display doesn't match the sound output that you're hearing. It's easily resolved by re-sending the project or re-sending the settings.

Synchronizing the Matrix³

You can re-send the Mixer Configuration, Project data, and the current state of all mixer settings manually by selecting commands from the LCS menu. This ability to "refresh" or "re-synchronize" the state of the LX-300 hardware can be very useful when synchronization has been lost by going "offline."

Resetting the Matrix³ to a known state by a simple two-step procedure:

1. Resend Mixer Configuration: Open the **LCS** menu, choose **Send Mixer Configuration and Settings.**
2. Resend Project: Open the **LCS** menu, choose **Send Project.**

In each case a window with a progress bar will appear, showing the state of the data transmission as it proceeds. The first stage should be almost instantaneous, but Send Project may take a while if your project file is large.

Checking the state of the Matrix³ is also simple:

1. Open the **LCS** menu and choose **Request Mixer Setting.**

All windows will be updated to show the current settings of all controls.

Configuring the Hardware

An essential part of the start-up procedure for the Matrix³ is to load the proper System Configuration and Project data into the LX-300 hardware. This is done either through transmitting the data from CueStation to the hardware via the serial

connection, or by programming the Matrix³ to load this information from its own Flash memory or connected hard disk drive. The System Configuration and Project data are both contained in the Project file and are normally sent automatically right after the file is opened. Should there be any loss of synchronization between CueStation and Matrix³, it is generally a reliable solution to re-open the Project file to allow CueStation to automatically re-send the data.

Reading a Project from the LX-300

When a Project file is sent to the LX-300 there is a data conversion that occurs to optimize the speed of execution and reduce its memory use. Once the project has been sent to the frame, it is not possible for CueStation to retrieve the project.

For Matrix³ systems that have been set to run a project from frame flash memory, it is important to keep backups of the original project!

Automation Techniques

This article provides a few details regarding common cue automation techniques.

Overview

Mixer automation in CueStation is based on control settings and specifying the time needed to arrive at that setting.

This differs from simple “snapshot” automation because it provides control that changes dynamically with time.

CueStation automation specifies control values that occur at points in time that can be arbitrarily executed by the user. The specification includes the value of the control point, the length of time that the system should take to arrive at that point, and the length of time that the system should wait before starting to move the control to that value. As we will see this is a compact yet powerful way of defining moving automation.

A “control point” in CueStation is any parameter value that can be adjusted by the user, either manually or by the automation system. Mixer subcues contain two categories of control point values: parameter settings, such as input fader levels or bus assignments, and the time intervals regulating the transitions between them. Other control points address signal processing parameters, such as equalization and delay, that are seldom changed throughout a show. A third category, including automate and isolate, affects the operation of the automation system itself.

A second type of subcue, “Externals,” consists of commands to be sent to devices external to the mixer architecture. Just as a mixer subcue contains control points for faders or bus assigns, so an External subcue contains commands that address non-mixer functions (e.g., MIDI machine control of external devices, time code generation, MIDI program change messages, Wild Tracks playback).

The Nature of Subcues

Subcues contain the values for the settings of specific subsystems. There are fifteen categories of subcues: ten for various level control points and signal routing, three for EQ and delay, and special-purpose bus trajectory and external subcues. Each mixer subcue contains the data for all the control points for its type. For example, if a Matrix^3 system is set up with sixty-four inputs, then each input fader subcue will hold the values for all sixty-four faders. In contrast, an external subcue can contain an arbitrary collection of commands to address non-mixer functions such as control of external playback devices.

Mixer subcues are created by “capturing” the current control point settings for particular parts of the mix structure, such as Input Faders and Matrix Rows. In technical terms this means that a data structure of a particular type is created and given a name, and that the control point values are then written into its data registers.

In the Subcue Library this is done by selecting the desired subcue type from the list at the left of the window and then invoking the **New Subcue** command from the **Subcues** menu. An existing subcue can also be “updated” with a revised set of data by highlighting it in the list and invoking the **Update Subcue** command.

Most of the time, though, you'll want to create a cue that uses the subcue settings that you're capturing. In the Cue List and Cue Library windows, you can use any of several commands in the Cues menu to create a new cue, populate it with subcues, and capture the control point values into those subcues.

New Cue & Cue Entry & Capture All and **New Cue & Capture All** are two of these all-at-once commands, with the former also placing the new cue into the current cue list. In both cases you will name the new cue (and its subcues) and select the types of subcues that will be captured.

There are also times when you'll want to re-use an existing cue, but with different subcues — perhaps the cue has already been used in a list, but a small change to the sound design has been made. Again, you are provided commands to simplify the update: **Capture All** replaces the existing subcues in a selected cue with new subcues containing the current control point data, while **Capture Differences** replaces subcues in a selected cue with new subcues only if its control point values have been changed. In either case, the original subcues, and the cues that use them, are left unchanged: only the cue you have selected is affected.

Finally, there are commands which update the subcues themselves, either by changing the subcue control points directly. These commands offer the opportunity to alter all instances of a subcue in all cues which contain it — and the danger of unintended side effects, should it affect cues you hadn't planned on altering!

Update All Subcues in the **Cues** menu and **Update Subcue (Recapture)** in the **Subcues** menu are the commands which recapture the control point values for all selected subcues, overwriting all previous saved values. This change is irrevocable: the only way to undo the change is to set the control points back to their original values and update the subcues again.

Tip:

Four function (F*n*) keys provide cue-capturing shortcuts:

F1 **Update Cue**

F2 **Capture Differences**

F3 **New Cue & Capture All** (places a new cue in the Cue Library only.)

F4 **New Cue & Cue Entry & Capture All** (places a new cue in the Cue Library and also at the end of the current Cue List.)

Time Relationships

The structure of cue lists, and the internal structure of the cues themselves, provides several levels of control for time-sequencing of events. For manual cue triggering, the order of the cues in the cue list is predetermined but the time intervals between them are not. When cues are triggered from time code, both cue order and the moment of triggering are predetermined, according to specific time values programmed relative to the stream of time code.

We've called this automation scheme "snapshot" automation, but this is a slight oversimplification. Cues and subcues have considerable internal flexibility for the timing of events, so rather than being a snapshot, it's more like a small movie. In the case of our Input Faders subcue, each individual fader has its own Wait and Fade times that allow it to perform with a measure of independence from all the other faders in the subcue.

Granted, this is a relative sort of independence: a Wait time offsets the fader movement from the time when the entire subcue is recalled. It allows each channel to start its fade at an independent time, and the individual Fade times similarly allow the faders to reach their target levels at their own pre-programmed rates.

This type of relative time-independence exists throughout CueStation. Just as a subcue contains multiple control points with individual time parameters, a cue contains multiple subcues, each with its own wait time. Cues are in turn organized into cue lists, where the time intervals can either be pre-programmed or left open for manual triggering.

This nesting of timed control points offers a great deal of flexibility for creating complex level functions, but of course it also offers a vast opportunity for chaos. Thinking of these options as several alternatives is perhaps better, rather than to try to use them all at once.

Automation Solo

Automation Solo allows one or more mixer channel strips to be individually programmed without being linked to all the other channels.

Each channel strip has an automation toggle button that enables or disables automation (green when enabled) for that channel. The switching logic is similar to that used in audio solo functions:

- if none of the Automation Solo buttons are selected, then automation is enabled for all channels and all fader moves will be saved together in one subcue.
- if the automation buttons for any group of channels are checked, then only those channels will be automated. Any mixer subcues in a cue that contains a fader with Automation Solo–enabled channels will store and recall data for just those channel strips.

As of CueStation v3.0, Automation Solo and Isolate from Automation affect Mute, Polarity, Fader Level, Pan, and Bus Assign. They do not affect Channel EQ and Aux Send.

As of CueStation v3.1, Automation Solo and Isolate from Automation do affect Input Channel EQ.

When to use Automation Solo

The subsystem-wide grouping of control points provided by CueStation subcues may be adequate for the design of most mixes, but there are times when you must have independent control of just one channel or group of channels. For example, an elaborate soundscape of ambient sound effects may form the backdrop for a pair of actors who are moving spontaneously across the stage. As each actor hits each of her marks, the voice signal needs to be properly distributed to a set of delay speakers, for realistic imaging. This has to be done individually for each actor, so a single sequence of cues simply can't handle it.

In these situations, Automation Solo provides a solution. It allows individual channels to be automated independently of each other, or in arbitrarily defined groups.

Wild Tracks, too, provides opportunities for effective use of Automation Solo.

Using Automation Solo

To create a subcue using Automation Solo, toggle the **A** buttons for the desired channel strips to the on state (green), set the control point values, and save the subcue. When that subcue is recalled, only the automation-enabled channels will be affected by automation control. All other channels will continue to do whatever they were doing, without being affected by the new subcue. If they were in mid-fade they will continue their fades; if they were at rest they will remain at rest.

Subcues using Automation Solo can belong to successive cues in a cue list, and their fader movements can even overlap. For example, cue 1 might contain an input fader subcue with automation enabled for input 1 only, while cue 2 has automation enabled for input 2 only. If cue 2 is recalled while fader 1 is still performing a fade, that fade will continue even as fader 2 starts its own motion. This behavior can even take place within a single cue, just by incorporating the two input fader subcues into one cue and setting the fader wait times.

Automation Follow Subcue Settings

When you capture a subcue from the Console or Wild Tracks windows, several types of subcue are created. The Automation Solo setting is stored in the Input Fader and Wild Tracks Fader (respectively) subcue types.

Capturing in other windows creates only one type of subcue containing all the settings for that window, including Automation Solo.

Autofollow

Autofollow is a method for chaining several cues together so that they can be triggered by a single manual command. It combines the flexibility of manual triggering with the precision of automated timing. Autofollow cues can be set up so that all the linked cues will fire at essentially the same moment, or they can fire in an ordered sequence, with predetermined time intervals between them.

Autofollow delays are cumulative, because Autofollow cues are organized in a sequential list. The time between the triggering of the first cue in a sequence and the last will be the sum of all the Autofollow delays.

When to use Autofollow

Autofollow can be used whenever a sequence of automation events has been assembled from individual cues. The technique is a useful way to consolidate the manual control of a cue list as a show begins to “take shape” during rehearsals or during a run, by eliminating the need to trigger a cue manually when it always follows another by the same time interval. It can also be useful if you prefer to build up automation sequences in functional “chunks” by defining some cues as Console only, others as Matrix only, and so on, and especially for External subcues such as a sequence of MIDI Machine Control Locate followed by Play commands.

Using Autofollow.

The Autofollow relationship is set up in the cue list by setting two parameters: the identity of the cue to be automatically recalled, and the time delay before this happens. The automatically-triggered cue can be *Next,* which allows the creation of linear sequences of cues; or it can point to an arbitrary cue in the list, which allows a variety of looping and branching strategies.

A single cue, or a sequence of cues, can be made to loop indefinitely by pointing the last cue in a sequence back to the first cue (or back to itself in the minimal case). This can be useful for a “vamping” sort of structure, where a sequence of events must repeat for an indeterminate period. It is always possible to break out of such a loop by manually triggering the next cue following the loopback cue.

In another situation, a branching structure could be created by pointing the last cue in a sequence to another part of the cue list. This might allow an operator the ability to advance to an alternate sequence of cues should certain conditions exist (by manually advancing the cue list before the delay time elapses), or to jump to another cue sequence automatically if they do not (by allowing the delay time to elapse).

Wait and Fade Times

Besides the time relationships that can be programmed in the cue list, there are several nested levels of time control within the cues themselves. Each subcue in a cue has a Wait time, from 0.0 to 999.9 seconds, that determines when it will be recalled relative to the time cue was triggered. By default, all subcues have 0.0 wait time, such that all subcues are recalled at the same time the cue is triggered.

Subcue wait times are independent of each other and act to delay each subcue individually from the triggering of the cue. The wait time of any one subcue is independent of the other subcue wait times: there is no cumulative effect.

In addition to the Wait times, certain types of subcues have Fade times. Fades have the same time range and resolution as the Wait times, but they default to one second. This parameter is available only for Input Aux Row and Matrix Row subcues. The Fade time is the length of time it takes for all matrix crosspoint values to change from their initial values to the values programmed in the newly recalled subcue.

Fader and Pan subcues offer Wait and Fade times on a per-channel basis. This allows you to program fader and pan moves with different times and rates within the same subcue.

When to use Wait and Fade Times

Wait and Fade are controls for fine-tuning a mix, and there are probably few CueStation projects in which they are not useful, and not used. Fade times will be essential in almost every cue. Wait times may be more commonly used when mixing multitrack playback material, where entrances and exits of individual sounds happen at fixed times. Both Wait and Fade could be effectively used in shaping vocal and orchestral balances over time.

Using Wait and Fade Times

The basic idea is simple: after a cue or subcue is fired a control does nothing until its assigned Wait time has elapsed, then it begins its fade, arriving at its programmed value at the end of its specified Fade time. For example, the Input Faders for channels 1 and 2 might be programmed with zero Wait times and 5.0 second fades, while Faders 3 and 4 might have 2.0 second Wait times and 3.0 second fades. When the subcue is recalled, the first pair of faders will begin to move immediately. After the two-second wait, the second pair will move as well, but at a faster rate. At the end of the five-second period, all four faders will arrive at their target values simultaneously.

Multiple-Subcue Sequences

The evolution of a CueStation mix over time automation is normally accomplished by a succession of cues, with each cue containing a collection of unique subcue types. However, nothing prevents the coexistence of several subcues of the same type within a single cue, and if they are organized into a timed sequence using Wait times they can be useful.

When to use Multiple Subcue Sequences

This technique is most useful as a way to consolidate a cue sequence that is well worked-out and has fixed time intervals between successive automation moves. You may wish to "lock down" a timed sequence of fader and matrix movements, perhaps as part of a sound effects cue that also includes a Wild Tracks Play command. Each successive fader or matrix row subcue would have its individual set of level settings and wait/fade times, resulting in a precisely repeatable "signature" effect.

Using Multiple Subcue Sequences

Multiple subcue entries are created by dragging the subcues individually from the Subcue Library, into the right-hand pane of the Cue Library. This allows you to create subcue sequences that could not otherwise be captured, such as complex fades (using several fader subcues and setting wait times between them.)

It is important to realize that you cannot easily create these multiple subcues by using **Capture Differences**. The Capture Differences command can't identify which specific subcue should be replaced in a cue, and thus will replace all the subcues of the same type with a new entry. You will then have to drag your original subcues from the Subcue Library back into the cue.

Update versus Capture Differences

Once a set of cues and subcues has been created, revising the programming in various ways is possible. After recalling the old version of a cue, you could make whatever changes are needed and then create a new cue to replace the original. This has the advantage of keeping the unaltered original as a backup in the Cue Library, but it can result in an accumulation of unused cues. (You can remove unwanted cues and subcues using the optimize and purge commands in the **Cue Lists** menu in the **CueList** window.)

A better alternative to using **Capture All** just to make changes is to use **Capture Differences** to update the cue. New subcues will be created for any parameters that have been changed from their previous values in the cue. The old subcues will remain in the Subcue Library. This is a non-destructive way of editing a cue and you can always go back to previous settings by restoring the original subcue.

These new subcues will be named after the originals by automatically adding a numeric suffix, thus maintaining a visible record of their history. You can also rename them. Cues that use the original subcues will not be affected by the update.

If you want to make a permanent global change so that any cue that uses a specific subcue will be updated, you can use the **Update** command. This will overwrite the old subcues with new values, while retaining the original name. This is an action that will affect all cues that use the same subcues.

Since this can not be undone, you should use the **Duplicate Subcue** command if you think you may want to revert to the previous settings.

Automatic Subcue Naming

Whenever subcues are created by means of a **Capture All**, CueStation automatically names them after the cue that contains them. For example, a cue named “Cue 0” will have subcues named “Cue 0 : System Level,” “Cue 0 : Input Bus Assign,” and so on.

Suppose you want to change the name of the cue after the fact? This is easily done with the **Rename Cue**, but only the cue itself will be renamed. The subcues will still retain their original name. In order to rename the subcues as well it is necessary to issue a **Capture All**. This will force the naming of the subcues to conform to the name of the parent cue. And, because the Capture Subcues window gives you the option of capturing only certain classes of subcues, you have the freedom to create a new set of names for just a selected type or types of subcues.

Matrix Mixing

This article compares manual and automated console mixing, and provides an overview of the advantages and limitations of automated matrix mixing. We will present examples of how automated mixing can be used to perform powerful sound control effects, and you will learn to identify opportunities to use matrix mixing in your sound design projects.

Manual versus Automated Matrices

Most sound engineers who have experience with larger mixing consoles are familiar with the basics of matrix mixing: audio buses deliver signals to an array of potentiometers, which in turn are used to scale the amplitudes of the signals before distributing them to multiple outputs.

Take as a specific example a small matrix, having eight inputs and eight outputs (8×8). Each matrix input is split eight ways. Each split goes to a gain control and then is combined with the signals from the gain control of the seven other inputs to feed an output.

Inputs are on rows, outputs on columns. The gain control is at each junction, or "crosspoint," of a row and column. This allows each output to be a different mix of the 8 inputs. In this case, with eight inputs being mixed and distributed to eight outputs there are a total of 64 crosspoints.

The usual hardware control surface for a matrix mixer is the familiar "wall o' knobs" with many rows and columns of potentiometers. Stage-monitor consoles are common examples of such a configuration, and large front-of-house (FOH) consoles typically have a matrix section as an output option. Even in consoles that lack an output matrix *per se* there is usually some matrix functionality in the guise of auxiliary sends or monitor ("foldback") sends on the input channel strips.

The layout of the matrix window in CueStation is similar to the layout of knobs in a hardware matrix, but instead of crosspoint knobs there are number boxes. The visual appearance is similar to a spreadsheet, with the rows representing the input buses and the columns representing the outputs. At each crosspoint is a number box containing a scaling factor in decibels operating over a range from +10.0 dB to –∞. Unity gain is represented by 0.0 dB and –∞ equals full off.

To help clarify the relationship between the way the user interface looks and the way the signals are controlled, let's examine some simple cases, using our example of an 8×8 matrix:

- If all 64 of the number boxes are set to –∞ this means that all signals are scaled to full off and no signal will appear at the outputs.
- If all eight crosspoints in the first row (bus 1) are set to zero (unity), all eight outputs will receive the signal from bus 1 at an equal amount, at unity level.
- If all eight crosspoints in the first column (output 1) are set to zero, output 1 will receive a mix of the signal from all eight buses, each at unity level (this may overdrive your outputs!)
- If the diagonal crosspoint values are set to zero (where bus number = output number: bus 1 to output 1; bus 2 to output 2; etc.), and all other values are set to –∞, then each output will receive only the unity-level signal that comes from the same-numbered bus.

Consider this last case: any time the matrix is set so that the crosspoints form a unity-gain diagonal from upper left to lower right, the bus assignments in the Input Console window are equivalent to direct assignments to system outputs: if an input is assigned to bus 1 it will be routed to output 1, and so on.

CueStation programmers often use this special case as a starting point, or as part of a basic system test procedure. With the 256 buses available on the LX-300 experienced programmers will often map the first series of buses at 1:1 to provide a quick path for a direct output from an input. Even on a system with a large number of outputs there will still be plenty of buses for matrix mixing and Space Map moves.

Advantages of Automated Matrix Mixing

The matrix sections of a few sound reinforcement consoles have some automation capabilities, but the flexibility of their routing is constrained by their limiting the number of matrix inputs (buses) to be equal or less than the number of hardware outputs.

The software approach taken in CueStation has significant advantages, a principal one being that the dimensions of the matrix can be set arbitrarily within very generous limits. With any given Matrix^3 configuration there can be up to 256 matrix inputs. The outputs can include not only feed to physical modules, but also to “loop-back” channels, allowing complex processing like inter-matrix EQ and Delay.

This number of buses allows plenty of flexibility for multiple, alternative mixing strategies, allowing you to reserve certain subsets of buses for specific signal distributions (e.g., buses 1–16 assigned directly to outputs 1–16, buses 17–32 distributed to surround speakers, buses 32–40 controlled by SpaceMap trajectories. etc.)

It is also possible to set up the Matrix^3 system so that matrix functions are available directly from the signal inputs, by using auxiliary sends. Any physical output of an LX-300 can be defined as either a normal output or an auxiliary output. Normal outputs show up in the user interface as columns in the Matrix window. Auxiliary outputs show up as rows of Aux Sends in the Console window. An input matrix made up of aux sends has the ergonomic advantage of being in the same window as the input faders, and because each aux crosspoint can be either Post- or Pre-fader the signal path can be much simpler. Of course, a simpler signal path can also be a limitation, and a matrix made out of aux buses can’t be controlled by the SpaceMap function. Such tradeoffs are both a challenge and an opportunity!

Challenges of Matrix Mixing

Matrix mixing offers a great deal of flexibility to the CueStation programmer, but it also has some inherent challenges that are the result of trade-offs in the software design. The Matrix window packs a large number of control points into a small space. To keep things manageable, each matrix subcue stores only the level scaling values for the crosspoints for a single row. The Wait and Fade times are then applied per matrix row subcue as part of a cue definition.

Another challenge makes it difficult to program smooth, equal-power panning effects using matrix crossfades. Because the gain versus time function of faders and matrix crosspoints has been optimized for natural sounding fade-in and

fade-out effects, there is a significant power drop of 18 dB at the midpoint of a crossfade between two outputs. The solution is to use Space Maps. This will allow a natural sounding crossfade, and the rate can be edited as part of the trajectory.

But perhaps the greatest challenge is the very lack of limitations that the matrix imposes, combined with the “big array of numbers” it presents. Getting a feel for the matrix can be difficult if you are used to looking at the position of a knob. Using number boxes instead of graphic faders places more programming control in the space than would be possible with graphic knobs, but the tradeoff is a diminished sense of control surface. Using a Space Map to set values will help in translating the array of numbers and give them a more obvious meaning.

The flexibility of the Configure Mixer functions also makes it easy to overdo it when defining the number of buses in a system. Each bus you define corresponds to a new row in the matrix window and a separate Matrix Row subcue in each cue. This can become unwieldy if you define far more buses than are actually needed. For the sake of structural clarity, think through your needs in advance and add only a few extra buses.

When to Use the Matrix

Matrix mixing is commonly used for:

- Signal routing of inputs to output.
- Crossfades between unequally scaled signals.
- Creating sophisticated timed pans.
- Alternative signal routing schemes.

Matrix mixing is useful for signal routing through the system. It’s also used when a signal must be distributed proportionally to several outputs in a way that is beyond the limits of the two-way input pan control in the Console window.

The input pan controls are flexible, within the limitations of their design: they can be programmed with individual wait and fade times per channel, and the panning law that governs the signal distribution ensures that there will be no signal drop midway through the pan. However, the signal distribution is simple and only twofold, with the signal being distributed in equal amounts to a set of assigned buses at one end of the pan, and fading to a second equal distribution to the buses at the other end of the pan.

Any situation that calls for a crossfade between two groups of unequally scaled signals (such as 0.0, -6.0, +1.2 dB) to another (such as -3.5, 0.0, -0.8 dB) is a suitable application for matrix mixing. Programming such a crossfade is accomplished simply by saving each of the two groups of settings to a separate matrix subcue and then including these in a pair of cues. The crossfade time between them is set in the second of the two cues.

Another application of matrix mixing might be to create the simulation of a moving sound source that travels through many speaker locations. Here the crosspoint values would first be programmed for significant points along the path, such as those moments when the sound is coming from a single speaker. Then the crossfade times might be set and tested, and adjustments made. You may need to program some intervening points along the path, to create a smooth effect. It would be an iterative (and potentially time-consuming) process. In all but the simplest cases it may be an easier and more elegant solution to use CueStation’s SpaceMap feature, as described in a separate article in the Users Guide.

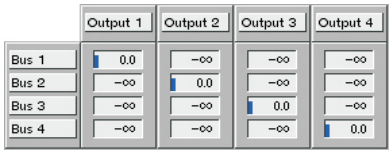
The mixer topology offered by CueStation may not cover every need. Sometimes the matrix can be used to design alternative signal paths, such as an external loopback of outputs to inputs as a way to implement crosspoint EQs and delays, or an auxiliary send that taps a signal from a bus rather than an input. Such novel strategies are often born of the moment, and we trust that the matrix functions will provide a sufficiently general tool kit to realize your programming inspiration.

Matrix Applications

The matrix is a flexible tool that can be used in various ways. Different users will approach matrix programming in different ways, and a single user may use different programming approaches for different projects. You might find these few strategies useful:

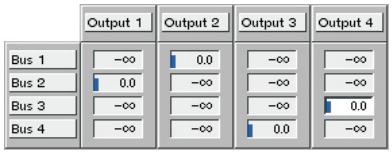
One-to-one assignment of buses to outputs

To route input signals directly to individual outputs by using the Console window bus assignment switches, the matrix must become “invisible” by passing the signals straight through to the outputs. The crosspoint levels are set to unity for each case where the bus number equals the output number. The resulting configuration looks like a diagonal of unity settings running from upper left to lower right in the Matrix window.



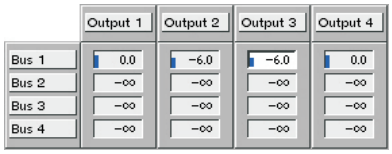
Remapping of buses to different-numbered outputs

The matrix can act as a simple crosspoint switcher to reassign buses to individual outputs. This may be useful when you want to select output assignments from the Console window, but the physical outputs aren’t connected to speaker channels in a simple or logical manner. The crosspoints acting as “on” switches will be set to unity (0.0 dB) and the rest of the crosspoints will be set to off (−∞). Note that a matrix row could also serve as a static signal splitter or “mult” by assigning a bus to more than one output.

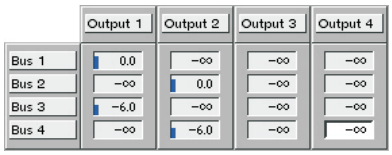


Proportional signal distribution to multiple speakers

Besides simple signal splitting and routing, the matrix can be used to distribute a signal proportionally to several outputs by setting the crosspoints to different values. This is useful when multiple speakers must provide coverage over a wide area, especially when level, EQ, and delays are used to balance the resulting sound field. The same signal might be sent to a main channel of the proscenium cluster and to a set of fill speakers serving the under-balcony seating. Although the program material might be the same at the source, each set of destination speakers needs to be fed from a different channel and at a different level. By controlling this in the matrix, and by taking advantage of the LX-300’s own EQ and delay, there is no need to use external splitters or processing devices.



Creation of overlapping zones



The matrix can be used to mix the signals from multiple buses to a single output. A simple application of this is to derive a mixed signal from a stereo pair for the center channel in an L-C-R speaker array, or to derive a monaural fill mix from a multichannel main house mix. More elaborate applications of this technique might serve the needs of an outdoor theme park installation, where are large number of speakers is used to create a continuous soundscape from a number of discrete playback sources.

Dynamic panning from one group of speakers to another

Any of the preceding routing and distribution techniques can form the basis for dynamic panning by creating a sequence of cues and using suitable fade times to transform one mix continuously to another. For example, the effect of a sound source that changes apparent size or stereo width (such as a flock of birds approaching and then fanning out to surround the audience) could be achieved by initially routing several buses to a single output, and then progressively fading up additional crosspoints to move the sound into additional speaker zones.

Auxiliary (effects) sends from grouped signals

On occasion it may be useful have an auxiliary send that “branches off” from a mix bus further along in the signal path, rather than using the auxiliary sends on several input channels to feed signals to an auxiliary output. This could be carried out by using a normal output (rather than an auxiliary output) as an effect send and by controlling its level from a matrix crosspoint. This send could be kept independent from other matrix functions by giving it its own bus assignment. This could prove useful for global reverberation effects, or for simple control of a low-frequency effects (LFE) channel.

Controlling individual crosspoints

Matrix Row subcues are constructed so that all the crosspoints in a given row are subject to a common set of Wait and Fade times, programmed in the containing cue. This limitation can be bypassed to some extent by using a separate bus for each group of crosspoints that requires individual control.

To treat two sets of crosspoints as independent subgroups (e.g., one set feeding outputs 1–4 and the other feeding 5–8), first assign the input signal to two buses. Then program the Wait and Fade times for crosspoints 1–4 on the first bus only and the level changes for crosspoints 5–8 on the second bus only. Be sure to keep the unused crosspoints for each bus set to full off (−∞). You can apply this process even to the control of a single crosspoint, though this means dedicating an entire bus to that function.

This technique is one way to subdivide the matrix into several independent matrices, potentially useful in larger compound installations such as a theme park environment. For example, guests in a ride attraction might be transported from one environment to another, and each local soundscape may have a similar set of sounds but with its own particular dynamic mix.

SpaceMap panning

Complex panning can often best be achieved with CueStation’s SpaceMap feature — a graphical programming interface that provides a higher-level method of controlling the crosspoint values of the matrix as a continuous real-time process.

A SpaceMap is a graphic representation of a multiple-speaker configuration, usually custom-designed to correspond visually to the physical layout of the

speakers. The map is composed of triangular panning areas (called “trisets”) and uses an algorithm that ensures a smooth, equal-power distribution of signals as the “trajectory” of a sound moves around the SpaceMap. Multiple trajectories can be recorded and then incorporated into cues for later playback.

This technique simplifies the creation of highly complex dynamic panning moves that would be extremely difficult and time-consuming to create using the simple binary pan control in the input Console window or by a piecemeal “cut and try” method in the matrix.

SpaceMaps

Ordinary left–right panning no longer satisfies the needs of today's sophisticated audience. With surround-sound now the standard, everyone expects you to envelop them in a rich audio environment.

SpaceMaps are your best tool for meeting their expectations. They provide multidimensional control of sound, enabling you to move sound through space, and do so through an intuitive, easy-to-use interface.

This article will introduce the concepts underlying SpaceMaps, and will help you design and implement an immersive sound environment.

Introduction

The SpaceMap is a graphic programming interface for the Matrix section of CueStation. It appears visually as a diagram of the sound field defined by multiple loudspeaker positions. This diagram can be simple and concrete, showing the physical locations of the speakers, or it can be more abstract, allowing representation of mixing functions as fadeout zones or auxiliary sends. Designing a two-dimensional SpaceMap that will control panning in three-dimensional space is possible, though such elaborate mappings are best tackled after mastering the basics of simple planar panning.

Note that traditional pan controls are one-dimensional: they shift sound along a line between two points. So-called "stereo" sound is, in fact, one-dimensional sound: there is left and right, but no up and down, or front and back. A SpaceMap, on the other hand, is two-dimensional and can even be used for three-dimensional work. It can position sound on any plane and, by panning between planes, can position sound anywhere in space.

A SpaceMap is constructed from two basic elements: nodes and trisets. Nodes commonly represent the positions of loudspeakers or groups of speakers, and they may be of several types. A triset is a triangular panning surface defined by three nodes: it is the graphic representation of a panning algorithm that distributes an incoming signal to three outputs while maintaining an overall unity gain. Like the one-dimensional panning law used by a conventional pan pot, the triset ensures a smooth pan without a drop in the signal at any point, but in two dimensions.

You might wonder, "Why a triangle?" The answer is simple: just as two points are the minimum required to define a line (traditional, one-dimensional panning), three points are the minimum required to define a plane (the SpaceMap's two-dimensional panning).

The primary use of a SpaceMap is as a control surface for dynamic panning of moving sounds. Think of it as the "stage" upon which spatial pan controls can "perform." The path along which a particular pan control moves is called a trajectory. The trajectory may be recorded in real time, edited and reshaped, and then recalled as part of a SpaceMap subcue. Several of these bus trajectories can be active at once, one per bus, and there can be an arbitrary number of SpaceMaps and Trajectories in a Project file.

Although trajectories are normally created for a specific SpaceMap, the two entities are actually independent until specifically linked together in a SpaceMap subcue. This allows you to use one map with several different trajectories, or vice-versa.

The Parts of a SpaceMap

The simplest way to think of a SpaceMap is to visualize it as a picture of the physical loudspeaker layout in a system. The most elementary configuration is three speakers set up in a triangle and represented by a single triset in the SpaceMap window. The physical proportions of the triangle is not critical, because the power-preserving panning law is proportional rather than absolute, being based on the relative distance between the spatial pan control and each of the three surrounding nodes. If you move the pan control around within the triset, the associated bus signal will be distributed proportionally to the three outputs, with more signal going to closer nodes and less going to more distant nodes.

This basic principle is applied in every triset that makes up a SpaceMap. If the system has four speakers instead of three, the map can be constructed as a rectangle subdivided by a diagonal line to form two adjoining trisets.

Speaker nodes

Speaker nodes represent physical outputs of the Matrix^3 hardware. Each node is assigned to a single output, and the output numbers appear next to their corresponding nodes in the SpaceMap window. Nodes may be defined for any outputs you chose, whether these are connected to speakers, effects processors, or other devices. You may even assign multiple nodes to the same output, should that be useful.

In our previous example, a SpaceMap of four speaker nodes divided into two trisets, we have flawless panning within each triset, but suffer an awkward transition between the two trisets. The solution is found in the use of “virtual” nodes.

Virtual nodes

A virtual node can represent multiple speaker nodes. That is, signal sent to a virtual node is distributed proportionally to several speaker nodes associated with it in a relationship called a “Link.” By default the signal will be divided equally among all the linked speaker nodes, but it is also possible to define proportional weights to change the balance of the distribution.

Our simple four-loudspeaker example can be refined by placing a virtual node at the center of the four speaker nodes and by defining four trisets that share the virtual node among them.

The virtual node is linked to the four speaker nodes so that any signal that is assigned to it will be equally distributed among the surrounding speakers. Thus a trajectory traveling around the perimeter of the map will pan linearly from one speaker to the next, but a trajectory that moves toward the virtual node at the center of the map will cause the signal to spread out to all four speakers gradually. This proportional distribution method creates a convincing phantom image throughout the panning area.

This equal distribution was not possible using just two trisets without a virtual node: a sound moving diagonally across both trisets would have started at one speaker, then spread equally to the three in the first triset, then dropped out of its starting speaker. On crossing to the second triset it would reverse this process. At no point would it appear in all four speakers, nor in both the starting and ending speakers at the same time.

However, despite improving the multichannel panning with a Virtual node, our simple model still suffers a flaw: any panning path that extends beyond the trisets

causes the sound to be abruptly stopped. Fading would be preferable, and the solution is to use “silent” nodes.

Silent nodes

A Silent node can take part in a triset in place of a speaker node, but it is not associated with an output. Signal panned toward a silent node simply disappears, and this provides an easy way to create fade-in and fade-out effects.

This feature also solves the signal dropout problem that exists when a trajectory strays outside the panning area defined by a triset of speaker nodes. Since the area outside a SpaceMap is undefined, the signal abruptly drops out. By surrounding a SpaceMap with silent nodes you can create a perimeter of “fadeout” trisets and guard against unintended loss of signal.

Derived nodes

The Derived node provides a way to send a signal to a secondary output whenever that signal is being sent to a primary output through the action of speaker nodes in trisets. Like a Virtual node, it is linked to one or more speaker nodes but the logic of the signal routing is reversed. Whereas a Virtual node divides a signal proportionally among its linked Speaker nodes, the Derived node receives a sum of the signals from its linked Speaker nodes. The Derived node will generally not be part of a triset.

Derived nodes are commonly used for subwoofer sends, fill mixes, balconies, and other cases where a secondary mixdown of a multichannel mix is required. For example, a Derived node feeding a single subwoofer would be linked to all the main channels, so that even if the source signal is being panned the subwoofer will receive a constant feed.

Virtual Weights

By default, Virtual nodes distribute the signal to their linked Speaker nodes, and Derived nodes receive the signal from their linked nodes in equal proportion, but this proportion can be modified by using Virtual Weights. You can edit these parameters by choosing a Virtual or Derived node and then selecting “Edit Virtual Weights” from the Space menu. Each node will have one weight value for each Speaker node it is linked to. These values are not absolute; they represent ratios between the amounts of signal related to each Speaker node. You can represent a ratio of 2:1 by weights of 2.0 and 1.0, 1.0 and 0.5, 30.0 to 15.0, or any other numeric equivalent. You’ll recall that signal levels are measured in decibels, and that +3 dB is equivalent to a 2:1 ratio between two signals (just as -3 dB represents a 1:2 ratio). Similarly, a 6 dB difference can be achieved with a 4:1 or 1:4 ratio between weights.

Take for example an LCR SpaceMap that is set up so that a hard pan Left or Right causes the signal to be 3 dB down in the Center speaker, and a pan to the Center causes the signal to be 6 dB down in each of the Left and Right speakers (the combined L+R signal is 3 dB below the Center, so the overall level is the same). This is achieved by using Virtual nodes for the Left, Center, and Right positions. The weights for the Left and Right Virtual speakers are (1.0, 0.5) so that the Center receives half the power (-3 dB) of either side speaker. For the Center the ratio is doubled to (2.0, 0.5, 0.5) to achieve an additional 3 dB difference between the Center and each side speaker.

Virtual speakers above and below provide continuity, and Silence nodes surround the entire map. Note that the speakers themselves do not need to be in any triset!

Trajectories

A trajectory is a path along which the spatial pan control will travel, when fired as part of a cue. The position of the pan control interacts dynamically with the SpaceMap's trisets as it traverses them and controls the matrix crosspoint values for the trajectory's assigned bus. This can create the illusion of a moving sound, it can be used to fade sounds in and out, or it can control a variety of other effects depending on the design of the SpaceMap.

You can use the SpaceMap's "Test" mode to try a trajectory out before recording it, or you can even perform real-time manual panning, but normally you'll want to create a fixed path that can be triggered at will.

Our description of the SpaceMap as the stage on which a trajectory performs is especially apt when we consider that each trajectory is an entirely independent entity, with no absolute relationship to any one SpaceMap. Moreover, a trajectory can be played back with real-time modifications, such as number of repetitions, rate, orientation, scaling and offset in both X and Y dimensions. Several copies of the same trajectory could be assigned to different buses and performed simultaneously with different modifiers on each. This might mean that two sounds could move in a mirror image to each other, using the same trajectory in its original form on one bus and in inverted form on another. Similarly, several sounds could be made to travel in concentric circles by using the same circular trajectory with a different scaling factor for each.

General Design Guidelines

Unlike surround-sound pan controls optimized for cinema sound in the 5.1, 6.1, or other "standard" formats, the SpaceMap interface has been designed to be extremely general and adaptable to any conceivable speaker configuration. This means that there is no absolutely "correct" way to design a SpaceMap, though there are some "incorrect" or at least "ill-advised" methods and also some "recommended" procedures.

SpaceMap "Do"s

- Analyze both your venue/system configuration and your sound design needs before embarking on an elaborate SpaceMap design.
- Create "sub-maps" of portions of a large configuration rather than trying to represent all in one. You'll gain more detailed control of trajectories if you have more room to draw them.
- Take advantage of the "snap-to-grid" feature to constrain your node positions to an orderly arrangement. A tidy SpaceMap is a well-behaved SpaceMap.
- Surround your SpaceMap with a perimeter of "fade in/out" trisets, using Silent nodes, to guard against sudden dropouts.
- Take advantage of BeOS workspaces and the full size of your monitor to create large SpaceMaps with plenty of navigation room and precision.

SpaceMap "Don't"s

- Don't overlap trisets. This will cause unpredictable and confusing results.
- Don't leave any parts of a SpaceMap undefined. This can result in signal dropouts. It is possible for a triangular area to be surrounded by trisets, giving it the false appearance of being a triset itself.
- Don't neglect to define links for all Virtual and Derived nodes.
- Don't get hung up on designing a SpaceMap to be a literal representation of your physical speaker layout: sometimes the most creative applications will be more abstract.

Design Guidelines for Simple Systems

Although we've just told you not to get hung up on using a SpaceMap as a literal representation, this is still by far the easiest way to get started. With simpler speaker configurations it may be all you really need to do.

The SpaceMap window even gives you a set of "training wheels" in the guise of "Set Background Picture..." in the Plot menu. This allows you to import a graphic file as a template for each SpaceMap. It might be a drawing of your venue or it might be a more generalized diagram against which to compose your SpaceMaps and trajectories. Just be aware that this picture will hide the window's grid: you cannot use both the imported picture and the grid as visual guides.

The first step in creating a simple SpaceMap is to place a set of speaker nodes in a configuration that represents the physical layout of the loudspeakers. This arrangement doesn't necessarily need to be exactly to scale. After all, your physical system configuration may have to be designed for a room that doesn't have the same proportions as the virtual space you are trying to simulate. How you accommodate to this situation will depend on too many particulars to lay out any general rules. Try to keep in mind that you'll always be striving for a compromise between physical space and perceptual space.

Learn by doing: experiment with the system, and you'll learn quickly.

Similarly, no hard rules can determine your speaker layout, though typical configurations serve particular types of venues and shows. One common design is for the creation of a soundscape framed by the proscenium, as in a crowded street scene. Some speakers may be placed at the level of the stage and others suspended higher up, forming a panorama that has both width and height. A SpaceMap for such a design would be drawn as a front view of the stage, from the audience's perspective. Other configurations might employ surround speakers encircling the audience, possibly even including some overhead speakers. Here the SpaceMap would represent an overhead view, looking down on the audience.

Once the speaker nodes have been laid out it is necessary to connect them into trisets. Most of the time you will also need to create some virtual nodes to "tie together" the speaker nodes. As a rule of thumb, it is common to place a virtual node at the center of a group of four speaker nodes, which allows you to create additional trisets and ensure a smooth panning transition from any location to any other. Another case might be when speakers are laid out in a linear array, so that they don't enclose a space in which to form a triset. A virtual node can provide the essential third vertex of the triangle.

Once you have a set of speaker nodes and virtual nodes connected neatly by trisets, you'll want to create a "buffer zone" around the perimeter. This will usually be done with silent nodes. The easiest way to do this is to place silent nodes near the edge of the window, positioned to form trisets between each pair of outer-edge

speakers or virtual nodes. This will insure that any trajectory that slips outside the main trisets will not drop out abruptly. It also provides a way to fade signals in and out. Another common effect is to include speaker nodes functioning as reverb sends. This can allow a sound to move from the inner “dry” region of the map, out to a “distant” reverberant zone, and finally to silence.

Your finished SpaceMap should be neatly laid out and should represent your venue and speaker configuration in a clear and intuitive fashion. All areas of the map should be defined by trisets and there should be no boundaries where signal drops off abruptly from one side to another. Check your work in “Test” mode by moving the pan controller around inside and between trisets, while monitoring the smoothness of the pan. Test the edges of the map to be sure that the signal fades smoothly in and out.

If possible, do all your testing in the venue itself, from the audience’s perspective, and with loudspeakers on. If this isn’t possible, then you can at least check the SpaceMap’s basic functionality by placing the Matrix window in the same workspace as the SpaceMap window, and watching the behavior of the crosspoint values as you move the pan control.

Design Guidelines for Complex Systems

Most systems that are large enough to invite the use of SpaceMaps are also complicated enough to require SpaceMap design that goes beyond simple mirroring of speaker locations. The first stage of abstraction is to subdivide the speaker system into zones, each with its own map or set of maps. Such subdivisions may be obvious (such as creating one map for surround sound environmental effects and another for a virtual pit orchestra) or subtle (such as maps that cover the same physical region but incorporate different groups of speakers).

When starting to venture beyond simple physical mapping, it is important to fully absorb the ideas that:

- A SpaceMap is not one comprehensive representation of a sound system configuration,
- Multiple SpaceMaps can operate simultaneously,
- SpaceMaps can function together, can overlap to some extent, or can be entirely independent.

One reason to use multiple SpaceMaps is to reduce complexity: if a given trajectory will use only a few speakers, there is no need to map all the speakers; they just add complexity and unnecessary design overhead that cannot benefit you. With SpaceMap design, the K.I.S.S. principle is your best ally!

Another reason for subdivision can be found in larger surround configurations, where a trajectory moves from one part of a room to another. In such cases you might build a chained sequence of SpaceMap Subcues, each using a different but connected SpaceMap and trajectory. For example, a sound will be flown across the stage and the to the right wall, across the back, up the left wall, and back to the stage. One can set up a separate SpaceMap for each surface, reducing the complexity of the maps. When using this technique, particular care must be taken to “smooth the joints” where a trajectory crosses from one zone to the next.

Non-theater such as theme park attractions and outdoor installations may consist of multiple independent spaces and will require completely independent SpaceMaps. Again, there can be a great deal of variation in the design. A set of adjoining rooms that are acoustically isolated would have entirely independent

sound fields and therefore distinct SpaceMaps. On the other hand, an outdoor soundscape is more likely to have “bleed” from one area to another, requiring adjoining SpaceMaps to share their edge nodes.

Design Guidelines for Trajectories

As always, it’s a good idea to plan ahead before recording your trajectories. It helps to imagine your sound environment, both aurally and visually, before designing the SpaceMap. If possible, sit in the venue and picture how your sounds will move around the space. Consider the audience’s perspective, and do this from various seating areas. Make sketches. Think about timing. Take a stopwatch to rehearsals and time how long certain stage actions, music cues, and scene changes take so you can plan your sound cues objectively. Do this as early as possible: you may find that part of your spatial design must be done before you even approach SpaceMap and trajectory creation.

One you have a clear idea of how your sounds need to move through space you can design an appropriate SpaceMap, or set of SpaceMaps. It may also help to spend some time listening analytically to moving sounds in the real world. It is often surprising how many wrong assumptions we make about our own perceptions.

For instance, the timbral quality of a distant sound such as an airplane or train may vary as it moves, either due to its interaction with its surroundings or due to its changing orientation toward the listener. This may lead you to crossfade between two or more sounds (on separate buses) as they travel along the same trajectory. Another consideration may be the apparent size of a sounding body. Again, this will vary with distance, and again an approaching train can serve to illustrate: When the sound is first detected in the distance it is a point source, and it may be heard to gradually increase in loudness but not so much in direction. As it comes closer, movement and direction become more perceptible, as do changes in the timbre and detail of the sound. Finally the train is “in your face” and it clearly has a sonic presence of orchestral dimensions, spanning the proscenium and filling the entire frequency spectrum. Here, as many as four separate recordings may be required, each controlled by a different trajectory, with gradual crossfades and divergence as the sound evolves.

The more one knows about the dynamics of moving bodies, the acoustic behavior of sound, and the psychoacoustics of perception, the more realistic one’s sonic illusions can be. According to Newtonian mechanics, the velocity of a moving object is defined by both speed and direction, and a change in velocity is acceleration or deceleration. Our senses are finely tuned to these changes and any attempt to fool the senses should take this into account. In designing sound trajectories, keep in mind that acceleration effects are more evident at those points of inflection where a sound may “whip” around a corner. These portions of a trajectory should be designed in more detail, with more data points and particular attention to the rate of acceleration and deceleration. If you use a simple SpaceMap (e.g., a rectangle with a speaker node at each corner) coupled with a simple trajectory (e.g., that same rectangle defined by just four trajectory points) the resulting illusion will fool no one.

Be sure to take advantage of the trajectory editing features. While most of us can draw a fairly smooth freehand curve with a mouse, or can point and click with accuracy, navigating an exact path while maintaining an “organic” feel is difficult. If a trajectory must “nail” certain hit points along its path, it is easier to create it piecewise and then smooth it out later. On the other hand, if a trajectory calls out for the more “organic” feel of a real time performance, doing this with a free hand

may be best and without worrying too much about precision. The path can be reshaped later.

Details regarding the creation and editing of trajectories are found in the Common Tasks section of this Users Guide.

Advanced Design: The Z-Axis

The simple examples we've looked at are immediately useful as the basis for spatial panning in a planar array of speakers, but what about installations with width, depth and height? The Z-axis may be the orphan child of cinematic surround sound, but in many larger theatrical and theme park installations the perception of height is crucial to convincing sonic illusions. Matrix[3] can easily handle this but the current SpaceMap window uses a strictly two-dimensional interface which tends to limit one's ability to think in three dimensions.

The trick to getting out of this box is to think about the box in various and novel ways. The simplest technique is to deconstruct your three-dimensional speaker array into its component planes. If we expand our previous four-speaker example and think about a cubic configuration with eight speakers we find that we can visualize this three-dimensional arrangement as six two-dimensional quad arrays, each representing a face of the cube.

Sound can be navigated across any one of these surfaces by using a rectangular SpaceMap similar to the one we designed previously. If we want to move the sound from one face to another, we can change from one SpaceMap to another at the point where the trajectory crosses from one face of the cube to another, using two SpaceMap subcues in succession.

An alternative to this could be to create one SpaceMap that contains both panning surfaces, so that we could use just one continuous trajectory for the entire sound path. This in fact is where the process gets really fun (or really confusing), because there really is no end to the topological games you can play with a SpaceMap. If you think of SpaceMap design in the light of cartography, you will see that you are faced with similar problems of deconstruction and unfolding that map-makers face when they want to represent our spherical Earth on the flat pages of an atlas.

The core problem is also akin to those faced in mechanical drawing and architecture. You can view any three dimensional object in many ways: front, side, or overhead; in perspective; exploded component view; unfolded box; whatever you can imagine. It's a bit of a mental stretch perhaps, but we've seen many ingenious solutions to the flat earth problem in mathematics, engineering, visual art, and literature (recommended readings: Flatland, books on topology, M.C. Esher, origami, cartography, engineering drawing, cubism…)

Using Degrees of Abstraction

It is possible to design a SpaceMap whose trisets are entirely composed of Virtual nodes. One practical application of this is in the type of large environmental sound installation that uses repeating zones or "cells" of loudspeakers to cover a wide area with surround sound effects.

The Fremont Street Experience in Las Vegas and the annual New Year's Eve celebration in Times Square are examples. In both cases a single rectangular SpaceMap controls multiple quad arrays of speakers (at Fremont Street a total of

52 channels!). The corners of the rectangle are formed by virtual nodes linked to corresponding speakers in the various cells (the stereo orientation alternates from one to the next). This SpaceMap is used with an alternative map made up of individual Speaker nodes. This provides for “fly-though” effects that span all the individual zones.

Such a strategy can be applied to other complex installations where many speaker channels are individually controllable but must be re-zoned dynamically. In such cases you can duplicate a single SpaceMap made up of virtual nodes, and then redefine the linked speaker nodes for each copy of the map. This saves on drawing many similar maps and it presents intriguing possibilities for dynamic sound transformations that may not be imagined by one working with a more conventional one-to-one representation.

Using SpaceMap Subcues

Before a trajectory can be used for playback in a cue or cue list, it must be incorporated in a SpaceMap Subcue. This can be created either in the SpaceMap window (“Capture Trajectory Subcue”) or in the Subcue Library window. The SpaceMap Subcue editing window allows the binding of a trajectory to a pair of SpaceMaps, with several modifier parameters to reshape the path of the trajectory and govern its performance. The trajectory can be made to repeat a fixed number of times, or indefinitely. It may be rotated, scaled and/or offset in X and Y dimensions, and panned between two SpaceMaps. Its playback rate and its signal level can be scaled.

A parameter called “Divergence” causes a certain amount of the signal to be spread evenly throughout all the speaker nodes in the map. This is used to counterbalance the “sweet spot” limitations that plague surround sound for larger audiences. It may reduce the strength of the panning effect, but it keeps the signal from being lost to some listeners as the sound pans away from them.

These modifiers can be used in various ways, ranging from simple adjustment of trajectory performance to an efficient way to “multiply” a single trajectory and to use it in several guises. For instance, a single “figure-8” trajectory could be used in several simultaneous SpaceMap subcues, with a variety of rotation settings, to create a precisely choreographed swarming of sounds. Or, a trajectory could be “reflected” about the Y-axis by giving it an X scaling factor of -1.0, thus allowing two sounds to move together in mirror image. Several sounds could be made to chase each other along a single path just by putting time offsets between their subcues.

Besides spatializing such “multiple mono” signals, it is sometimes useful to fly multichannel sources by carefully controlling the space and time relationships between the trajectory subcues. In such cases it is important to keep in mind the objective reality your spatialization is meant to suggest. Does a physically large sound maintain the same apparent size, or does perspective play an important role in the audience’s perception? Does the source always maintain the same orientation to the audience or does it rotate in some manner? Is the sound source a rigid object, such as a steamship or a cargo helicopter, or does it change shape and size, such as a thunderstorm or a swarm of bees?

Subcue parameter details are found in the *Basic SpaceMaps* section of the *Common Tasks* chapter, page 5-12.

External trajectory control

Besides the usual cue playback of prerecorded trajectories, it is possible to "perform" a sound trajectory by remote control by sending serial messages representing the X-Y coordinates of the spatial pan control in the SpaceMap window. These messages are in CASL format (CueStation's proprietary control data format). This mode resembles test mode, where the pan control can be moved manually by the mouse, but it has the advantage of allowing high precision real time performance control of multiple simultaneous trajectories. An obvious application for this might be in an interactive virtual reality environment, such as planetarium shows or interactive games, or one might generate trajectories mathematically on a second computer and then record these in the SpaceMap window for later, fixed playback.

See Also

The following books contain information on spatial hearing:

Begault, D., *3-D Sound for Virtual Reality and Multimedia*. A clear and comprehensive presentation of 3-D audio principles and current technology. Unfortunately, this book is now out-of-print. (You may find it at http://human-factors.arc.nasa.gov/ihh/spatial/papers/pdfs_db/Begault_2000_3d_Sound_Multimedia.pdf) ISBN 0120847353.

Blauert, J., *Spatial Hearing:The Psychophysics of Human Sound Localization*. The standard reference on the psychophysics of three-dimensional hearing. (See Mills (1972) for a shorter overview.) ISBN 0262024136.

Bregman, A. S., *Auditory Scene Analysis:The Perpetual Organization of Sound*. A massive description of experiments by the author and his students on the factors that influence the formation and segregation of sound streams. The first and last chapters are readable by nonspecialists, but see Handel (1989) or Yost (1991) for an easier introduction. ISBN 0262521954.

Carlile, Simon, Ed., *Virtual Auditory Space: Generation and Applications*. An excellent survey of the psychophysics of spatial hearing and the generation for spatial audio. Highly recommended. ISBN 1570593418.

Handel, S., *Listening:An Introduction to the Perception of Auditory Events,* A recommended general introduction to the psychology of hearing; includes a good summary chapter on neurophysiology. ISBN 0262081792.

Stern, R. M., Jr., *An overview of models of binaural perception*. A useful survey paper directed at models that attempt to explain all known psychoacoustic phenomena. Available in *Proc. 1988 National Research Council CHABA Symposium*.

Yost, W. A., and G. Gourevitch, Eds., *Directional Hearing,* The most important collection of research contributions since Blauert's book. Includes results of direct measurements of head-related transfer functions. ISBN 0387964932.

Project Management

When you approach any sizeable sound programming project, the greatest risk of failure comes from poor planning. As malleable, uncertain, and artistic as a sound design project may be, there is nothing to be gained-and a lot to be lost-in taking a “hack and patch” approach to your work.

This section provides project management ideas that can guide you toward a controlled, but flexible, system for succeeding in your projects. We recommend you further your learning: read the books we list at the end of this article.

It begins by explaining how data is organized in a CueStation Project, as this may affect your decision in choosing a project design approach.

How Cue/Subcue Data is Organized

The organization of CueStation automation data is hierarchical: a cue list contains cues, a cue contains subcues, and a subcue contains control point or individual commands. However, like many things in the computer “metaverse,” this is just the surface level of the user interface as the CueStation programmer sees it. The deeper reality, though less literal, is nevertheless easy to comprehend and to use to your advantage!

In computer programming there is a concept called “indirection” which is a way to efficiently and flexibly organize data by providing pointers to data storage locations, rather than directly providing the data itself. In this way, multiple processes that operate on the same sets of data don’t have to keep individual copies at all times — they just retrieve the data they need when they need it.

That is how the cue structure of CueStation is organized. All the cues in a Project file are kept in a Cue Library and all the subcues are kept in a Subcue Library. The relationships between cue lists, cues, and subcues are really just sets of pointers, or links, between the higher-level data structures and the lower-level ones. That is, a cue list contains pointers to a set of cues in the Cue Library, and a cue contains pointers to a set of subcues in the Subcue Library. If a cue list contains multiple instances of one cue, it actually contains multiple pointers to the same cue; it doesn’t need to “contain” several duplicates of the cue itself.

This data storage and retrieval scheme has real benefits. When a new cue is created and the current state of the mixer is saved into subcues, CueStation does not have to create an entirely new subcue of each type. If a group of settings such as bus assignments had not been changed since the creation of a prior cue, then a pointer to the existing subcue will be incorporated into the new cue. This is both efficient, from a data storage standpoint, and kinder to the programmer, by limiting the number of unique “objects” that must be managed. It is also self-documenting by providing a visible overview of changes in the cue/subcue structure, and it adds editing efficiency by providing a way to effect global changes by simply updating an original subcue referenced by multiple cues.

Building Cues

One way to view the creation of a CueStation project is in terms of the opposition of “top down” versus “bottom up” methods.

Top-Down

Top-down programming starts with the creation of a cue list, and then proceeds by the creation of cues within that list. According to the situation these cues might be created in sequential performance order, or they might be created as a series of "master cues" and rearranged as needed. In either case the typical procedure is to set all the signal routing, level controls, EQ, and delay and then invoke the "New Cue & Cue Entry & Capture All" command (quickly done by typing the "F4" function key and then "OK" in the dialog that appears).

As the name suggests, this command does three things: 1) it creates a new cue, 2) it places the new cue in the cue list, 3) it captures the current state of the mixer control points and saves these in a full set of subcues inside the new cue. All this is obvious and visible in the cue list window, but "behind the scenes" the new cue is also being stored in the cue library and all the new subcues are being stored in the subcue library.

This procedure can be a very quick way to rough out a show. You can focus primarily on how the mix sounds, and secondarily on the progression of the mix over time, and give almost no thought to the inner structure of subcues. Fine-tuning and revisions are also quick. Cues can be moved, duplicated, modified, and deleted with simple key commands, again without thought to their inner structure.

Bottom Up

The bottom-up approach starts with the creation of subcues, and then assembles these into cues. This is clearly a much slower and more deliberate method than the rough and ready top-down method, but it has virtues of economy and precision. It also allows the assembly of subcue combinations that would not be possible when Capturing All into a cue, such as a timed sequence of several fader subcues as a way to construct complex fades.

Direct creation of subcues is also the only option when incorporating External and Bus Trajectory subcue types into a cue. These subcues are not part of the mixer settings and therefore they cannot be captured into a cue. They must be constructed from their basic elements: a list of commands for an External subcue and a collection of SpaceMap, Trajectory, and playback parameters for a Bus Trajectory subcue.

Playing Both Ends Against the Middle.

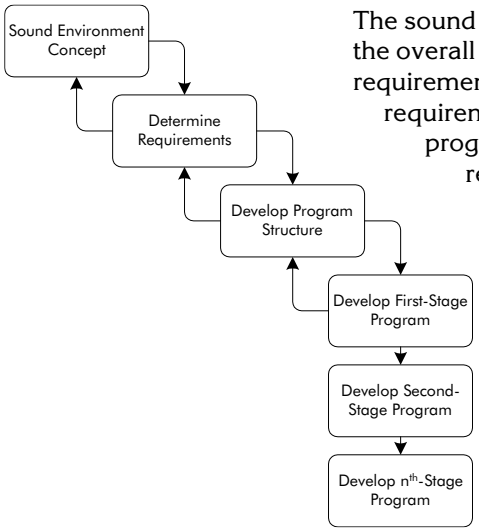
In the real world creating a CueStation project entirely according to one method or the other is uncommon. Most programmers will combine the two in a congenial and workable balance. For instance, it may make sense to establish a master mix that sets up the state of the entire system at the beginning of a show, or perhaps at the top of each act or scene. Then the moment to moment changes might be better programmed in detail by creating cues containing, say, just Input Faders or just certain Matrix Rows.

Methodologies

Rather than flail about in an attempt to create everything on the fly, repeatedly fixing mistakes, we suggest that you take a mildly disciplined approach to programming your automation cue list. It's best to set out the requirements for the project and to design some overall architecture and structure before creating subcues, cues and cue lists.

A half-dozen well-recognized project methodologies are available to you. One excellent overview of software project development can be found in the book Rapid Development: Taming Wild Software Schedules, by Steve McConnell. In a chapter entitled “Lifecycle Planning” the author describes several alternative ways to manage a software development project. The best of these methods are:

Staged Delivery

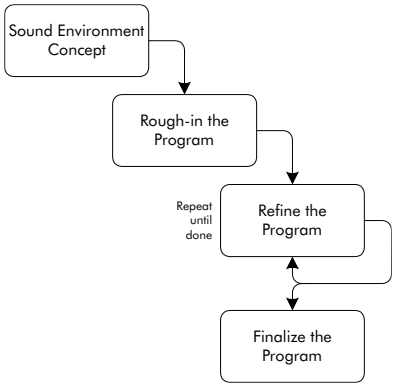


The sound environment concept is created and, as the details near completion, the overall requirements to realize the concept are determined. As these requirements are refined, the concept itself may need some revision. As the requirements near completion, the overall structure and design of the sound program are developed and, again, as this knowledge becomes refined, the requirements are revised.

Once the architectural design nears completion, you’ll determine which parts of the automation program to deliver first. Finally, CueStation automation programming can begin: you’ll develop subcues, cues and cue lists to deliver the first stage of the automation mix. As each stage nears completion, changes to the design may be made. Refinement of previous stages continues as successive stages are begun.

The strength of this approach is that it delivers the most important functionality first and encourages up-front design decisions that will make it easier to create the automation program itself. The disadvantage is that it requires careful planning.

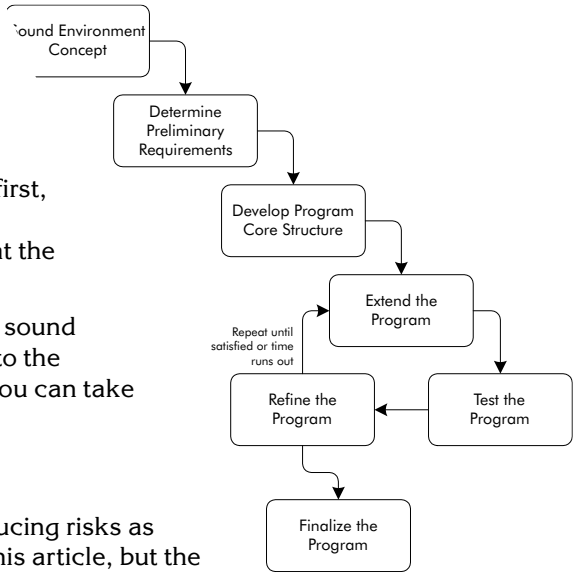
Prototyping



The sound environment concept is developed, and a prototype is created to determine the viability of the concept. The prototype will probably deal with the most significant part of the overall concept, and it is then successively refined until the sound designer is satisfied with the state of the program: the prototype evolves to become the final product. Within each iteration, requirements are clearly defined and the groundwork laid for the overall design of the subcues, cues, and cue list.

This method is especially effective when requirements change rapidly, or when the sound designer isn’t altogether confident of the concept. Its biggest drawback is that it’s impossible to set a time line or cost: you don’t know how much iteration will be necessary. It’s also easy to fall into a hack-and-fix mentality, instead of designing the structure before creating the subcue, cue, and cue list details.

Evolutionary Delivery



You can combine the staged and prototyping models of project work by using the staged approach to lay the groundwork and developing ever-increasing functionality for the overall project, while also using iterative prototyping within each stage to evolve a finished product.

Whereas prototyping tends to deal with the most audible parts first, then allows you to go back to fix up flaws and mistakes in the structure of the program, evolutionary delivery first ensures that the foundation is solid, before dealing with the flashy bits.

This approach has great flexibility: if you can't predict what the sound designer will want, you'll take more of a prototyping approach to the program; but if the requirements are well-defined and stable, you can take more of a staged approach.

Spiral

This method is highly sophisticated, and is oriented toward reducing risks as much as possible. A full discussion is far beyond the scope of this article, but the basic idea is to work on mini-projects, each dealing with a major risk factor in the success of the design and development of the project. You'll probably prototype each mini-project, eking out the minimum required performance, and then successively refining the mini-projects. When you're confident that the major risks have been dealt with, you can integrate the mini-projects, test and release the final product.

The greatest advantage to this approach is that as the project progresses-and the cost of failure increases-the actual risk of failure decreases. If there are technical problems that will prevent you from succeeding, you're going to discover them very early in the project.

The downside to this approach is that it is complicated. Determining objectives and risks can be difficult, and you must be committed to rigorously following a predetermined project action plan that keeps everything on-track and ensures all the bases are covered.

Flowcharts in this section were adapted from McConnell's "Rapid Development: Taming Wild Software Schedules."

Stages of Project Development

All rational project management methodologies share at least three common design stages, and their importance is emphasized in this analogy:

If you were dropped by airplane into a desert, armed only with a canteen of water, a map and a compass, could you find your way to the nearest oasis, some three hours walk from your starting point?

The answer: **Probably not!** Although the map clearly shows the oasis, you have no idea where you are. With no knowledge of your starting point, you cannot figure out (i.e., plan) which direction you need to walk.

To succeed in a project:

- You need to know where you are.
- You need to know where you want to be.
- You need a plan for getting there.

First, determine where you are. What signal inputs will you have? How do they behave? What hardware does the Matrix[3] system have? Where are the loudspeakers, and what are their characteristics? What is the environment like, and how variable is it? These, and many, many more facets of the overall system must be known in detail before you can effectively design your sound automation project.

Next, determine "where you want to be." What do you want the sound to do? How do you want to modify the sound environment? What special effects are needed? How flexible or fixed, simple or complex, do you want the cue list to be? Will your project be subjected to continuous changes during development, or can you rely on things staying the same?

Finally, plan how to get there. There's little to be accomplished by just throwing yourself straight into cue programming. Spend some time determining how you'll organize the structure of the cue lists, cues, and subcues; decide how you'll deal with changes, and generally plan your work before you begin. The few hours you spend on this will save you days of aggravation!

Lastly, you can begin your automation programming. The groundwork you've done will guide you, helping you work efficiently and effectively, even as circumstances change and you're forced to refine your original plan.

Cue List Structures

Single Cue List

Most theatrical stage productions can easily be served by a single cue list. Events proceed in a predetermined order, with little chance for variation.

Multiple Cue Lists

Larger productions with many cues may be more conveniently organized by using several cue lists, perhaps one for each act. This also could apply to "modular" productions, such as a repertory program of one-act plays, a musical review, or a set of multimedia shows in a themed retail store.

Another situation might require alternative versions of a cue list, as in a production with a rotating cast where there might be level differences between alternate actors' voices.

Master Cue List with Subordinate Lists

CueStation provides a **Select Cue List** command that allows switching from one cue list to another under automation control. This can be especially useful in modular productions where there might be some rotation of short programs over a day or a week. There could in fact be several Master cue lists, each of which calls up a different selection and ordering of a number of subordinate lists.

External Cue Recall without Cue List

Any cue in the Cue Library can be recalled directly by an external command, without the necessity of its being included in a cue list. This practice is most typical of a theme park or themed retail installation, or in architectural installations such as a large conference facility. The cue recall commands are typically issued by a master show control system and are transmitted through serial or MIDI connections.

Cue Structures

Full-Mixer Cues

The **Capture All** command enables you to create a cue that contains a full complement of mixer subcues, saving the current state of the mixer's entire control structure. This includes signal routing, input level controls with Wait and Fade times, matrix level changes, auxiliary send settings, and output levels.

Specialized Mixer Subsystem Cues

A cue may contain just a subset of the different subcue types such as Input Faders, Bus Assigns, and Pans or Matrix Row subcues. This allows changes in a particular part of the mixer control structure, as in a case where the state of the matrix remains unchanged while the inputs and pans are used for all dynamic mixing.

Such cues can be created in any of several ways: by dragging selected subcues into an empty cue, by using the subcue selection option in the **Capture All** command, or by deleting unwanted subcue types from an existing cue.

Cues Containing External Subcues

Capture All is an efficient way to capture mixer subcues into a cue, but in order for a cue to contain External subcues the needed Externals must be deliberately placed inside the cue. A cue may contain just one or several Externals, and these can reside in their own separate cue or can coexist with mixer subcues.

The choice of whether to combine mixer subcues and External subcues or to keep them separate is a matter of both programming style and practicality. For example, you might wish to bind together a set of Wild Tracks playback subcues with their associated fader movements and bus trajectories.

Command Control Structures

Manual GO Steps through Cue List

A single linear cue list is frequently used for live shows where the timing of cues must be synchronized with live performers. The sound operator directly triggers cue recall.

External Cue Recall without Cue List

Theme park shows frequently rely on a Master Show Controller to manage the synchronization of many subsystems, including sound, lighting, pyrotechnics, mechanical effects, etc. In such cases it is common for individual CueStation cues to be recalled directly by the show controller via a serial or MIDI connection to the Matrix^3 hardware.

Autofollow Sequences

Sequential chains of individual cues can be constructed using the Autofollow feature. The first cue in such a chain may be triggered manually, and then each subsequent cue is triggered automatically after a programmed time interval. Autofollow sequences are commonly interspersed throughout manually controlled cue lists.

Time Code Cue Recall

Cues within a cue list can be triggered automatically by SMPTE Time Code generated either by an external source or internally within the Matrix[3]. Time code can run continuously throughout a show or can be started and stopped as needed to run several synchronized cue sequences within the larger framework of a manually controlled cue list.

Direct Parameter Control from External Show Controller

In certain situations the cue structure can be bypassed completely and system parameters can be directly controlled through serial or MIDI commands sent by a master show control system. One such application might be the external control of spatial panning, with an external controller sending continuously updated (x, y) SpaceMap location data and the SpaceMap performing the actual matrix control.

All of the Above

CueStation allows the creation of complex projects that combine several different control strategies. This is relatively common in theme park installations, where several semi-autonomous systems are interfaced through a show control system. Such installations may have fully automated sequences of synchronized media alternating with, or in combination with, live performers. Or, in a walk-through environment the automated sequences must be held in readiness until the guests are in position, and then triggered manually.

System Configuration Types

Sound System Topology

Frontal sound field

A traditional sound reinforcement speaker configuration is designed to create a single, coherent sound field that emanates from the stage area. Monaural, two-channel stereo, and Left–Center–Right systems are all common. The primary loudspeakers are mounted close to the stage, and may be supplemented by “fill” speakers in various locations.

As the number of speakers increases, and as they are physically distributed throughout a larger space, it becomes increasingly critical that they be individually “tuned” with respect to relative levels, equalization, and time delay.

This is a primary use of the matrix in an LCS mixer. The sound designer’s goal may be to create a simple L-C-R Front of House mix, but each of the individual components of this mix is distributed to many individual outputs in the matrix.

Surround sound

Multichannel surround sound has become increasingly popular in live theater, progressing from a few simple offstage sound effects to the creation of complete enveloping environments. The matrix is key in the creation of these environments, and may be used in a variety of ways.

- Sound field zones

  Ambient sound fields can be prepared in advance and played back as multitrack recordings, or they can be mixed live from discrete sources.

  The practical advantages of the former method are that a premixed soundscape is literally repeatable and that it may require fewer discrete audio channels. The

aesthetic advantages of the latter method are that the “performed” soundscape can have greater sonic depth and clarity and that living elements in a show can be dynamically responsive.

A synthesis of these two approaches can also be used. The matrix can be used to overlap, crossfade, and interleave two or more sets of multitrack ambiences to create the impression of a much richer, more spacious, and less repetitive soundscape.

- Moving sounds

  Discrete sounds can be circulated through a surround sound system through conventional panning at the input stage, by direct matrix manipulation, or by using the SpaceMap function.

  Care should be given to the selection, placement, and tuning of surround speakers to achieve smooth coverage without timbral variations or level drop-outs. Such artifacts might be unnoticed with static sounds, but will surely be revealed as a sound moves from speaker to speaker.

  Height and distance are important parameters in the simulation of sounds in space, and the SpaceMap function affords continuous control of these effects.

Compound systems

Real-world environments rarely consist of a single isolated space with uniform acoustical properties. Careful listening will reveal that even in a single room the quality of sound will vary from one location to another as the reflective and absorptive characteristics of the local surroundings differ. Electroacoustic sound design should be informed by these realities and the Matrix^3 can be used to model multiple, connected acoustic zones.

In addition, the system has the power and flexibility to create independent environments which behave with total independence or with some degree of synchronization.

- Fusion of contiguous spaces

  Not every venue will be a simple, connected space, and there will frequently be a need to design separate speaker subsystems to provide coverage for all audience areas.

  This may be one of the largest challenges to the creation of natural-sounding audio, since the acoustic effect of the physically separated loudspeakers must be overcome to create the sense of a single, coherent sound field. Again, this can be addressed in part by skillful matrix programming and in part by informed use of the output EQ and delay functions of Matrix^3.

- Multiple independent environments

  Location-based entertainment installations, such as a theme park or museum, will frequently require the creation of several autonomous environments through which an audience travels. These may be entirely isolated from each other, as in an attraction where several groups experience different shows simultaneously, or they may be coupled in an acoustically natural manner, as when several adjoining soundscapes permit the guests to range freely between them. Because CueStation is designed to permit just a single cue list to operate at any time, such multiple environments are commonly controlled with a separate master show controller.

External Interfacing

The LX-300 is designed to support interfacing to a range of external equipment by virtue of its SCSI, serial, MIDI, and Ethernet connectors. This allows the Matrix^3 to fulfill a wide range of functions within a larger show control system, assuming the role of master show controller, behaving as a simple automated mixer, or anything between.

Show control

- Matrix^3 as master

  The Matrix^3 system can function as a comprehensive show controller, with the ability to generate time code and to issue full range of commands to external equipment. CueStation provides an open-ended yet easy-to-use method for command formatting. Common MIDI commands, including MIDI Machine Control (MMC) are directly supported by menu selections, and arbitrary System Exclusive or ASCII-based serial messages can be created as hexadecimal strings.

- Matrix^3 as slave

  All Matrix^3 functions can be addressed by serial messages formatted as MIDI System Exclusive commands. This message format was chosen for maximum flexibility, since any MIDI sequencing application can be used to create and send sysex messages. In addition, any show control system can format these messages as hexadecimal code, and they can be transmitted via a serial connection.

Audio playback

- Integrated Wild Tracks playback

  Wild Tracks provides sixteen tracks of instant-start audio playback, streaming from a SCSI hard disk attached to an EtherTracks interface card. Any combination of tracks can be triggered as a group, with sample-synchronous precision, and these groupings can be redefined “on the fly” as a show progresses. Each LX-300 frame can support one set of sixteen tracks, so that greater numbers of tracks can be supported by a single system. Each set of sixteen tracks is internally sample-accurate, though there is a slight variability in synchronization between groups.

- External audio playback

  Matrix^3 can control external playback devices such as MIDI sample players, hard disk recorders, and tape decks via standard MIDI commands (typically note messages), MIDI Machine Control, or MIDI Time Code. These commands will be included in External subcues and normally will be issue by recalling a cue from a cue list.

Labor coordination

Single Programmer

The basic CueStation working situation is that of a single programmer working on a single computer. Cue creation can be done in online or offline mode, so that work can be done even when the Matrix^3 hardware is not in use. Methodical programming is recommended. Libraries of special-purpose cues can be created as individual projects, and these can be combined using the Merge Project command. The Cue and Subcue Librarian thumbnails provide useful overview

information and allow commenting of cues and subcues. The Log can also be used to document one's work and can be written to and saved anytime.

The advantage to working solo is that there is no immediate need to communicate with other programmers or to coordinate efforts. This can have the benefit of short-term efficiency since time need not be devoted to explaining one's work to others. However, working alone can sometimes lead to inadequate documentation of the working process, making it difficult for another programmer to revise the project later. Good note taking and cue naming practice is highly recommended.

Development Team on a Network

Allows several programmers to work on different aspects of a project simultaneously and in different physical locations. Different Helper Apps may be run on different networked computers, as happens where one programmer is creating Wild Tracks cues while another fine tunes a mix. This can greatly increase the speed of project development but requires good communication, team coordination, and a project management plan.

Development Team Working in Shifts

This method can be particularly useful when a sound designer is creating sound playback materials that will be integrated into the project by other team members, or when a show combines mixing of live performers and effects playback and the associated programming must be scheduled during different shifts.

The keys to avoiding chaos are clearly understood ground rules and good documentation.

See Also

There are innumerable software programming project management books that contain valuable advice, methodologies and strategies that are easily applied to the development of sound automation programs.

Steve McConnell, *Rapid Development: Taming Wild Software Schedules*. A comprehensive, fact-based overviews of all best-practices project management techniques; it's a big book, but packed full of the best information available. ISBN: 1556159005.

Steve McConnell, *Software Project Survival Guide: How to Be Sure Your First Important Project Isn't Your Last*. This is an adaptation of Steve's Rapid Development book, targeting management. It's an easier read, but lacks some essential details. A good first read, but do follow it up with the larger book. ISBN: 1572316217.

Walker Royce, *Software Project Management : A Unified Framework*. Once Steve McConnell has made a structured-project convert of you, you'll want to have this book. It's hard-core, but contains the best project management information that's currently available. ISBN: 0201309580.

Joiner Group, *The Team Handbook*. This is the must-have book for team leaders and team advisors, and is highly recommended for team participants. Everything you need to know about teamwork is in this easy-to-use book. ISBN: 1884731112.

Robert K. Wysocki, Robert, Jr Beck, David B. Crane, *Effective Project Management*. An accessible book, with excellent descriptions of various project

management structures. Comes with a computer simulation of a project, that you can use to test your understanding of project management. ISBN: 0471360287.

Peter Degrace, Leslie Hulet Stahl, *Wicked Problems, Righteous Solutions: A Catalog of Modern Engineering Paradigms*. An easy-to-read book that reflects real life, and provides an excellent understanding of why project management is so important. ISBN: 013590126X.

Tutorials 4

CueStation is a powerful audio control tool. Most users require some degree of training if they are going to use it productively.

This chapter provides some of that training. Although it currently contains only a single SpaceMaps tutorial, we expect it will grow rapidly.

SpaceMap Tutorial

This tutorial is aimed at guiding you through the use of SpaceMaps. It will assume that you are using a single frame with eight inputs and eight outputs, and that your workstation is communicating with the Matrix^3 via a serial connection.

Concepts explored include:

- System configuration.
- Console and Matrix signal flow.
- Creating cues.
- Updating cues.
- Create cue lists.
- Adding cues to cue lists.
- SpaceMap design.
- Recording and editing trajectories.
- Creating SpaceMap subcues.
- Using the same trajectory in two subcues to create stereo moves.
- Using joysticks.
- Saving and loading projects.
- Default projects.
- Differences between CueStation versions.

Basic Setup and System Operation

System Startup and Communication

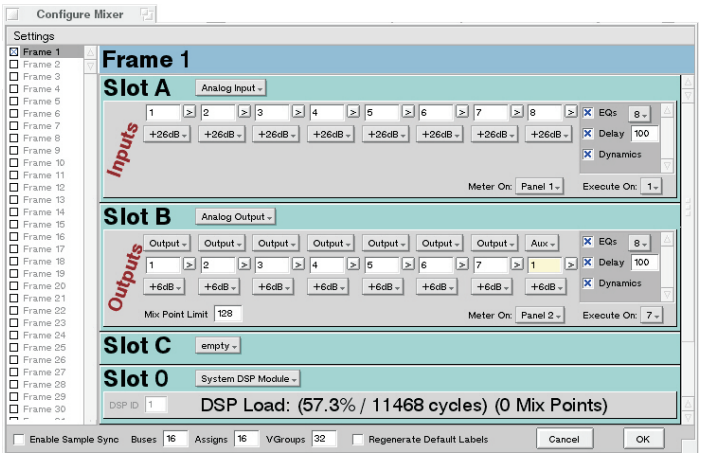
Launch CueStation by double-clicking the application icon. You will be asked to **Send Configuration and Mixer Settings** and **Send Project.** Press **Send.** The progress bar will pause for about seven seconds during this transmission. This is normal. If the progress bar does not continue, check that the serial cable is connected from the computer to the RS-232 port of the LX-300 Comm/Sync card.

To confirm that the LX-300 is operating correctly, open the **Windows** menu and choose the **System Status** command. This display will show the temperature, voltages, DSP load, number of mix points, and other values. This display will be blank if any of the following are true: CueStation is in “offline” mode, the LX-300 is turned off, or there is no serial or Ethernet connection from CueStation to the frame.

System Configuration

If your system is already configured correctly, you can skip this section.

1. Open any **Windows** menu and choose **Configure Mixer.**
2. Select **Frame 1** from the left hand side of the window and put a checkmark in the checkbox to its left.
3. On the right hand side select the appropriate I/O cards for slots as configured in your system.

   For this discussion, we will assume that an eight channel analog input card is installed in slot 1, and an eight channel analog output card is installed in slot 2. Select the appropriate full scale settings for the inputs and outputs. At the right hand side of each of the slot areas, you can configure signal processing as desired. For instance, configure the inputs for four bands of EQ, and the outputs for five bands of EQ, and delays. The maximum amount of delay available per channel is set in milliseconds with the text entry box to the right of the **Delay** label.
4. Click on the menu to the right of the **Slot 0** label, and select **System Module.** Since this system has eight inputs and eight outputs total, a total of 64 mix points will be required. Enter `64` in the **Mix Point Limit** field.

   The expected maximum DSP load for the frame is shown at the top of the window.

   At the bottom of the window, you can specify the number of mix buses, bus assign switches, and Virtual Groups.

   If you are using a multi-frame system, sample synchronicity can be turned on. If the checkbox is marked, the system delay will be uniform from any input to any output. If this is not checkmarked, a 2 to 3 millisecond variance will occur.

   When the **Regenerate Default Labels** checkbox is marked, all labels will be reset to defaults when you click **OK.**

   We suggest you specify Buses=32, Assigns=8, VGroups=32, Enable Sample Sync=off.
5. Open the **Settings** menu and choose **Renumber all channels.** This will number all the inputs and outputs automatically.
6. Click **OK.** The configuration will be sent to your frame.

Next, we will customize the user interface.

7. Open the **LCS** menu and choose **Configure User Interface.** The settings window lets you specify the number of channels that will be displayed in the various Cuestation mixer windows. We suggest that you set all of them to `8`.

   Press **OK** when you are finished.

Basic Mixer and Automation Settings

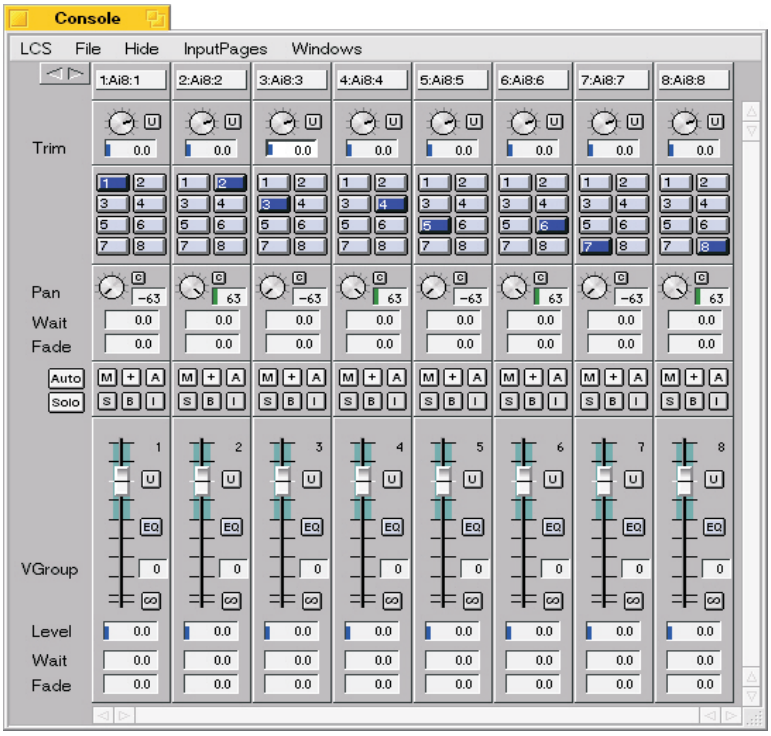
LCS uses a mix structure similar to a sound reinforcement mixer. The console section allows you to assign inputs to mix buses, and access the input EQs. The buses of the console are connected to a matrix. Bus Level faders set the levels of buses on the input of the matrix. Output Master faders set the levels of the outputs of the matrix, and provide access to the EQ section. Each row of the matrix specifies the output levels for each of the mix buses.

We will start by setting up inputs 1 through 8 to buses 1 through 8, and buses 1 through 8 to outputs 1 through 8, respectively.

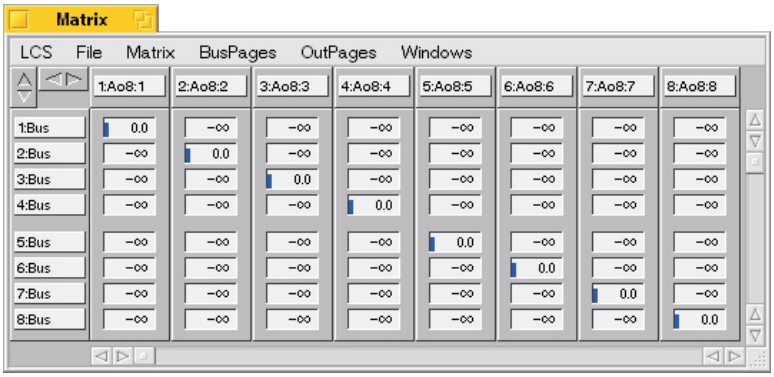
1. Open the **Windows** menu and choose **Console.** Set all inputs to unity by holding the CTRL key and clicking the **U** button on the fader. Assign Input 1 to bus 1 by clicking the **bus assign 1** button, located above the **Pan** control and below the **Trim** control. Continue by assigning input 2 to bus 2, input 3 to bus 3, etc.

   Set the pan control to opposite extremes for the even- and odd-numbered inputs. For the odd inputs, drag the **Pan** control until it reads full-left. For the even-numbered inputs, drag it to full-right.

2. Next, set up the rest of mixer points. Go to the **Bus Levels** window. Set all the buses to unit by holding the CTRL key and clicking the **U** button for each one.



3. Go to the **Matrix** window, open the **Matrix** menu and choose **Set Diagonal.** This sets unity along the normal diagonal axis, so that row 1 sends to output 1, row 2 to output 2, and so on.



4. Next, go to the **Output Masters** window and set all the faders to unity.
5. Go to the **System Level** window and set its fader to unity.
6. Finally, go to the **Manual System Level** window and set this fader to unity.

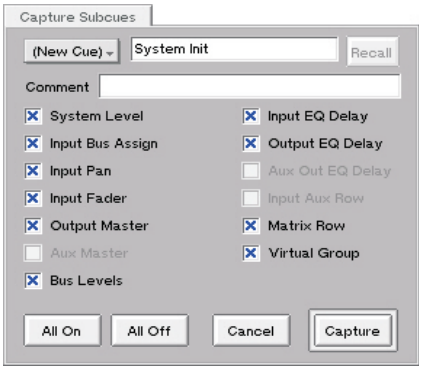
At this point any audio source patched into inputs 1–8 of the system should be heard on outputs 1–8.

Create a setup cue, and save defaults

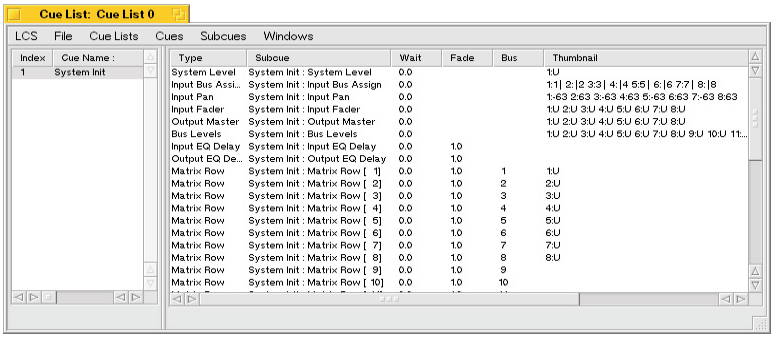
LCS presets are stored as “cues.” Each cue is a container for the details of a cue. The details are called “subcues.” The subcues contain the basic information for changing the settings of related sets of mix points (for instance, the input faders or matrix row settings). Cues are stored in the Cue Library, and the Cue Library window is available so that you can browse all the cues in your project. Cues can be placed in a sequential list.

1. Go to the **Cue List** window. Open the **Cue Lists** menu and choose **New Cue List.** Give the list a name (we used “Training List”).
2. Create a cue and place it in the active cue list. First, press the F4 key. The **Capture Subcues** dialog window will appear, with a default name given to the cue (“cue 0”). Rename the cue to `System Init.`

   The checkboxes let you specify what settings will be stored in the cue. Leave them all checkmarked for this exercise.



3. Click the **Capture** button. The dialog window will close, and your new cue will appear in the left-hand pane of the **Cue List** window. If you select a cue list entry, its details are shown in the right-hand pane.



In addition, other types of subcues are available. These include *Externals,* primarily for sending messages to externally attached equipment such as MIDI samplers; *Wild Tracks* subcues to configure and play back audio from the LCS hard disk playback system; *VRAS* subcues to configure Variable Room Acoustics System; and *SpaceMap* subcues, for playing back a trajectory of sound on a designated matrix bus with specific scaling parameters.

From time to time we recommend that you save your work. When CueStation starts up, it loads a default project. You can make your working file the default project by opening any **File** menu and choosing **Save as Default Project.**

If you wish to create a cue but do not want it to appear in the Cue List, press F3. You'll be presented with the same dialog window as for the F4 command, but the cue won't be automatically placed in your Cue List. In both cases, the cue will appear in the cue library window.

SpaceMap and Trajectory creation

Create a SpaceMap

We will create a "5.1" SpaceMap, with outputs 1, 2, and 3 going to left, center, and right, respectively; outputs 4 and 5 connected to left and right surround speakers; and output 6 feeding the sub-woofer.

1. Open the **SpaceMap** window. An empty SpaceMap is automatically created and displayed as the main workspace in the window.
2. Click the **Insert** button. When this button is active, the mouse is used to place new speakers on the map. Refer to the Tip to learn more about mouse modes.

   By clicking on the map, place *Speaker nodes* 1, 2, and 3 on the map in a horizontal line about $\frac{1}{3}^{rd}$ of the way down from the top, distributed fairly evenly from left to right, with space between the speakers and the edge of the map.

   Place speakers 4 and 5 about $\frac{2}{3}^{rds}$ of the way down the map, again spaced evenly.

Tip

To move a node after you've placed it on the map, click the **Select** button before dragging the node to a new location (if you don't change modes, your mouse click will insert a new node!)

To delete a node, use Select mode, click on the node you want to delete, and then press the BACKSPACE key.

To re-assign a node to another output, use Select mode, click on the node you want to change, type the bus number, and pres the ENTER key.

Other details of SpaceMap node editing can be found in the Task Reference section of the manual.

3. The *Virtual node* is used to distribute power from a node on the SpaceMap to multiple speakers. We'll use it to send power to all five nodes.

   Using **Insert** mode, place a node in the center of the 5 existing nodes. Open the **Type** menu and choose **Virtual.**

   Next, link the virtual node to the 5 speaker nodes. Click the **Select** button. Hold the SHIFT key down while clicking, one after the other, the five speaker nodes and the virtual node. They should all be highlighted. Open the **Space** menu and choose **Link Virtual/Derived.**

4. The *Derived node* is used to distribute power from multiple speakers on the SpaceMap to a single speaker. This is often used to distribute power to sub-woofers. We will use a derived node so that audio in outputs 1, 2, 3 (LCR) are reinforced by the sub-woofer connected to output 6.

   Open the **Type** menu and choose **Derived.** Use **Insert** mode to place a node above output 2. Set the new node to output channel 6 by selecting it, typing 6 on the keyboard, and pressing ENTER.

   Finally, link the derived node to speaker nodes 1–3. Using **Select** mode and the SHIFT key, select the nodes and choose the **Link Virtual/Derived** command.

Tip

When you choose a Type before placing nodes, the new nodes will have the Type you specified.

If you forgot to set the Type, you can always use Select mode to go back, select the nodes, and then change their Type.

5. The SpaceMap uses *Trisets* — groups of three speakers — to distribute power. Details on SpaceMap power distribution are explained in the *Understanding Cuestation* chapter.

   Click the **Triset** button. Now, when you click, the three nodes nearest the mouse are highlighted. These nodes can be turned into a triset by opening the **Space** menu and choosing **New Triset.**

   Create five trisets, with the center Virtual node in each of the trisets. You'll connect 1–2–V, 2–3–V, 4–5–V, 1–4–V, and 3–5–V.

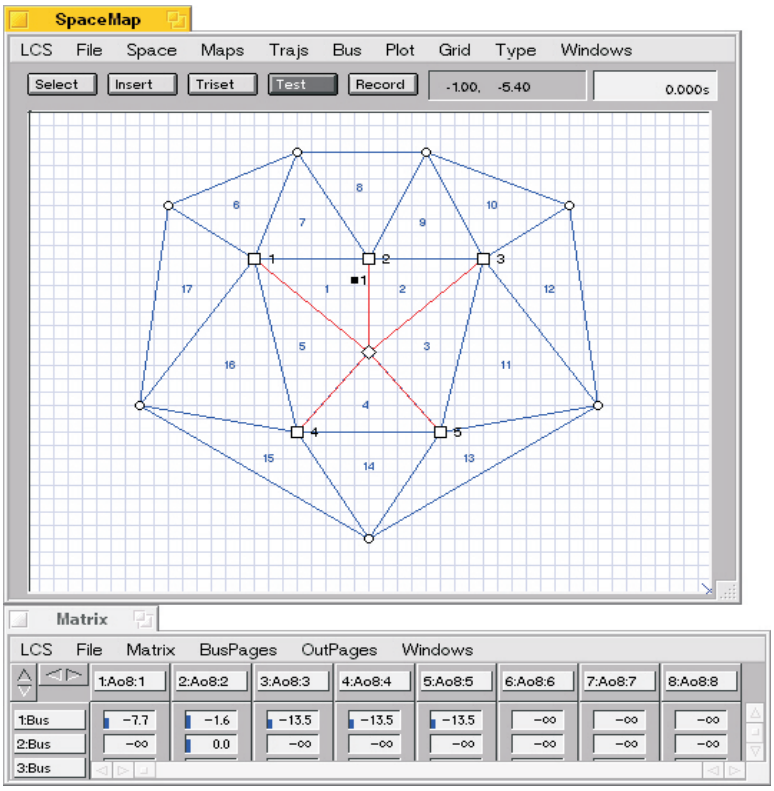
By following the instructions, you have defined a panning surface that defines the area between the five primary speakers. As well, audio that is distributed to outputs 1, 2, or 3 is also being distributed to output 6 because of the derived node.

This is a good time to check the SpaceMap.

6. Open the **Matrix** window and position it below the SpaceMap window so that you can see the first row in the map.
7. Click the **Test** button. Your mouse will now act as a panning control when you drag it.

   Click and drag in the SpaceMap window. As you do this, watch the Matrix window to see how the output levels for bus 1 are adjusted as you move the cursor around the map.

Notice that if you click outside the defined area of the map, all output levels for bus 1 will be turned off (set to -infinity). With the current design, power will abruptly drop off as you drag from within the map to outside the map.



To create a smooth transition, you can insert *Silent nodes* around the circumference of the map and define additional trisets connecting them to the speaker nodes. Your completed map should include speaker nodes, a single virtual node in the center, a single derived node, and a group of silence nodes around the map as shown below.

Since input 1 is assigned to bus 1, you can now listen to the SpaceMap by feeding a signal to input 1 and clicking or dragging through the SpaceMap. If you see the matrix values change but do not hear audio, check the mixer settings, starting with the Console and working through to both System masters.

When you have completed editing your SpaceMap, open the **Map** menu and choose **Lock SpaceMap.** This will prevent it from being accidentally deleted while creating and editing trajectories.

Create a Trajectory

Trajectories are paths of sound. SpaceMap subcues configure the rate, scaling of repetitions of Trajectories. The SpaceMap subcue is associated with a specific bus in a cue.

We will be working towards creating a stereo panning move in which the left and right channels start in the center channel and then move to the rear surrounds.

First, however, let's create a move that loops the sound around the interior of the map. This is a good way to check your SpaceMap.

1. Click the **Record** button. In this mode the mouse will be used to plot a trajectory path through the SpaceMap.
2. Click five times, about one second apart, near each speaker node on the side toward the Virtual node. Click the **Record** button again to stop recording.

As you clicked on the SpaceMap when recording the trajectory, you should have heard your audio source jump from point to point. When you play this trajectory, however, you will hear it pan smoothly from point to point.

3. Play the trajectory. Open the **Space** menu and choose **Play Trajectory.** You should hear the audio assigned to bus 1 pan around the five speakers.

   The trajectory will loop continuously. While it is looping, you can change to **Select** mode and drag points in the trajectory to move them. You can also change the duration of playback of all or part of the trajectory by selecting a group of points and then changing the value in the trajectory duration box (it's located at the upper right hand corner of the SpaceMap window.)

   To stop the trajectory, open the **LCS** menu and choose **Cancel Trajectories.** All running trajectories will be stopped. This command is in the **LCS** menu so that it can be selected from any and all CueStation windows.

4. Rename this trajectory to "Loop." Open the **Trajs** menu and choose **Rename Trajectory.** Type `Loop,` and click **OK.** We'll be referring to this trajectory later.

SpaceMap subcues

Next, you will create a SpaceMap subcue that causes the sound to pan around the map three times at twice the recorded rate.

1. Go to the **SpaceMap** window. Open the **Trajs** menu and choose **Capture SpaceMap Subcue.** This will use the current selected SpaceMap and current selected Trajectory to create a new SpaceMap subcue.

   Create a new cue in the **Cue Library** window. Open the **Cues** menu and choose **New Cue.** Name it `Loop Bus 1 Three Times.`

2. Go to the **Subcue Library** window, and position it below the **Cue Library** window. View all your SpaceMap subcues by clicking on **SpaceMap** on the left-hand side of the **Subcue Library** window.

   Using the right mouse button, drag the new SpaceMap subcue from the right-hand pane of the **Subcue Library** to the right-hand pane of the **Cue Library** window. The subcue is now in the cue.

   Test the cue by double-clicking the cue's name in the left-hand side of the **Cue Library** window. You should see and hear the trajectory play back one time.

Next, we will modify the subcue so that the trajectory plays back three times, at twice the rate at which the trajectory was recorded.

3. Double-click subcue in the right-hand side of either the **Subcue Library** or **Cue Library** window. An edit window will be displayed, providing play back options for the trajectory.

   Set **Repetitions** to `3,` and **Rate** to `2.0.` Click **OK** to apply the changes.

Now trigger the cue again, and listen to the change that you have made in the panning. Note that if you set **Repetitions** to `0,` the trajectory will play back indefinitely.

You can stop a trajectory that is playing by triggering another trajectory on the same bus, recalling a matrix preset, selecting **Cancel all Trajectories** from the **LCS** menu, or creating a Stop Trajectory subcue via the External Subcue Editor.

Stereo Panning

The scaling parameters in the SpaceMap subcue edit window provide a number of parameters that modify how a trajectory is played back. We have explored **Repetitions** and **Rate.** If you wish to create a multi-channel audio move, the **Offset** and **Scale** parameters are often used.

Let's create a cue that causes a stereo signal to be panned from the front center channel to left and rear surround channels, respectively.

1. Patch a stereo audio source into channels 1 and 2 of the system.

   Assign channel 1 to bus 1 in the Console window, and pan it hard left. Assign channel 2 to bus 2 in the Console window, and pan it hard right.

2. Create a trajectory that starts at the center channel, and ends at the left surround.

   Open the **Trajs** menu and choose **New Trajectory.** Name the trajectory `front to back` and click **OK.** Click the **Record** button and create a path between the center channel speaker and the left surround speaker. Press **Record** again to stop recording.

Next, you'll create two subcues that use this trajectory path: one for the sound effect's left channel (assigned to bus 1) and one for the sound effect's right channel (assigned to bus 2).

3. Open the **Trajs** menu and choose **Capture SpaceMap Subcue.** Name this subcue `Front to Back LEFT,` and click **OK.** Capture another SpaceMap subcue, but this time name it `Front to Back RIGHT`.

4. Take note: this second subcue is *identical* to the first: it travels from the center speaker to the left speaker. We want it to travel from center to right.

   Go to the **Subcue Library** window and select **SpaceMap** on the left-hand side. Double-click **Front to Back RIGHT** to edit its parameters. Set **X Scale** to `-1.` This will mirror its trajectory left-to-right — the original trajectory travels from center to left, so you've made this trajectory travel center to right. Click **OK.**

5. Go to the **Cue Library** window and select **New Cue.** Name the cue `Stereo Move,` and click **OK.**

   Using the right mouse button, drag the SpaceMap subcues **Front to Back LEFT** and **Front to Back RIGHT** to the **Front to Back Stereo** cue.

Both subcues will, by default, be assigned to bus 1. Assign the **Front to Back RIGHT** subcue to bus 2 by selecting the cue from the right-hand side of the menu and typing `2` in the **Bus** box at the bottom of the window, or by directly typing the number `2` in the subcue entry row of the cue. With either method, you must press ENTER to apply the change.

Trigger the cue and watch the SpaceMap window to see the relationship between the movement of buses 1 and 2. The **Offset** parameters of the subcue are used if you want to maintain a fixed linear relationship between the movement of two trajectories. Experiment by changing **X Scale** to `1.0` and then to `5.0`. Compare the difference: one will be "bigger" than the other.

Adding Cues to the Cue List

Earlier, you created a Cue List. The Cue List gives you the ability to sequence a series of cues so that they can be triggered by MIDI time code or by the operator pressing the "Go" button.

1. Go to the **Cue List** window. You should have a single entry in the list. *Using the right mouse button* drag the **Stereo Move** cue to the list. It will be inserted at the location under the cursor when you release the mouse button. Place it at the end of the list.
2. Go to the **Transport** window. Click the ← button to reset the Cue List position to the top of the list. Click the **Go** button to trigger the first cue, which should set your mixer to a default setting. Press the **Go** button again to trigger the second cue, which should be the Stereo Move cue.

You may wish to adust the relative levels of the left and right channels, or perhaps you made a mistake in the original setup cue. You can modify the setup cue to accomplish this.

3. Trigger the setup cue again.

   Reset channel 2 to unity, and assigned it to bus 2. Check that channels 1 and 2 are panned hard left and right, respectively.

4. Press F2. This will bring up the **Capture Differences** window, and lets you update the cue so that its settings include the revised console settings.

   Click the **Capture** button to modify the cue.

At this point it's probably a good time to save your project by pressing ALT-S, and/or saving as default project.

Joysticks

BeOS-supported joysticks can be used by CueStation. The joystick control is mapped to the *X*– and *Y*–axes of the Spacemap. The trigger button (primary joystick button) corresponds to the left mouse button.

Check that your joystick is recognized by BeOS. Open the **Preferences** window and select the **Joysticks** control panel. If your joystick is supported, it will be shown in the control panel.

If your joystick is not recognized, you may need to install the USB joystick driver. It is available in the *Drivers* section at http://www.bebits.com.

To configure the joystick for CueStation use, go to the **SpaceMap** window, open the **Space** menu, and choose **Configure Joystick Action.** You can set the *X* and *Y* axis to Absolute or Relative mode at the bottom of this window. If you choose Relative mode, you can adjust the sensitivity of the joystick.

Click the **Add New Joystick Action** to assign a Cue or Transport action to a specific joystick button. Click the right mouse button on the **Button, Action Type,** and **Cue** columns to change them.

Joystick settings are saved with your project file.

CueStation version differences

CueStation 3.0.1

- Clicking in **Test** mode should stop a trajectory from playing back, but does not.
- **Delete Trajectory** now works.
- **Delete SpaceMap** now works.
- Joystick **Relative** mode is now supported.
- Red virtual links are not faded.
- Trajectory color should be unique (should not be the same as the virtual link color).

CueStation 3.0

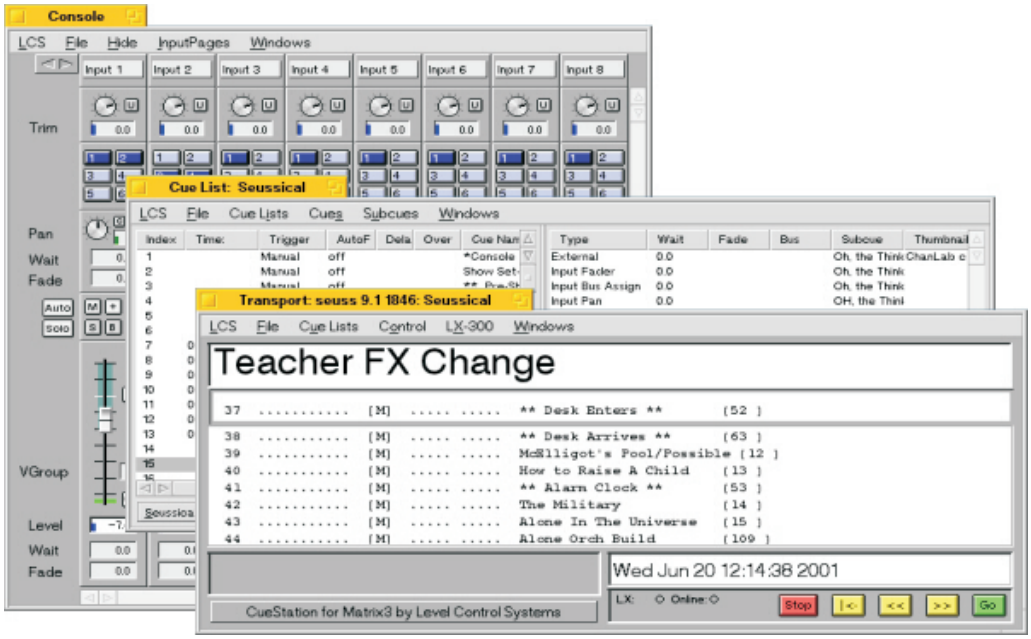
- Clicking in **Test** mode should stop a trajectory from playing back, but does not.
- **Delete Trajectory** will cause problems. Do not use it.
- **Delete SpaceMap** will cause problems. Do not use it.
- Joystick **Relative** mode is not supported.

Common Tasks 5

Beginning users may find it challenging to remember all of the procedures and commands in CueStation, even after training: it is a remarkably powerful and, at times, complex application!

To help you increase your productivity during your first weeks using CueStation — and to assist you in training assistants — we've provided a quick reference for the most common tasks that you'll typically perform while programming CueStation.

These quick references provide only the details necessary for accomplishing the task, and do not deal with esoteric techniques or applications. However, hints and tips are provided, as are cross-references to the foundations-oriented articles from *Understanding CueStation,* chapter 3.



Basic Interface

A number of shortcuts available to the CueStation user. Many of these are new or improved for CueStation v3.

Faders

To copy the setting of one fader control point (i.e., the pan or fader control), right-drag it to another channel. To copy a control to all channels (even the ones that are not visible), hold down the CTRL key while right-dragging the control.

To copy the setting to a pair of channels, hold down the SHIFT key while right-dragging the control. The control setting will be applied to the odd-numbered channel and the even-numbered channel on its right.

The Z key ("Zero") sets a control value to zero. The X key ("maX") sets a control to its maximum value while the N key ("miN") sets it to the minimum value. HOME and END are alternates for X and N.

PAGE UP and PAGE DOWN increment a control value by an appropriately small amount.

To adjust a pair of channels simultaneously, hold down the SHIFT key while dragging the control. The adjustment will be applied to the odd-numbered channel and its even-numbered channel to its right.

When viewing windows with more than one page of controls, the F7 and F8 keys page left and right. In the Matrix window, you can also the F5 and F6 keys to page up and down.

Tip

Trace the signal path for any channel by clicking the **Channel Label** and choosing **Log Signal Trace.** The **Log** window will list all the information that is known for that channel, from the location of the physical input, to its fader and processing settings, Matrix routing, group assignments, and the location of the physical output.

Cues and Subcues

Right-dragging a subcue from the Subcue Library to the Cue List window will create a new cue that contains that subcue. Right-dragging a cue within a list move it to a new position.

Lists can be sorted by clicking a column heading label.

To edit a value in a list, just click on it. You must press ENTER to have the new value accepted. Double-clicking the **Edit** dot for a Subcue will open an Editor for "external" subcues. For mixer subcues you will get a dialog box asking if you want to recall that control setting.

Double-clicking on the **Cue ID** in the **Cue Library** will execute the cue.

Double-clicking on the **Index** number for a **Cue** entry in the **Cue List** window, will execute the cue.

The shortcut keys F1 through F4 are used to capture and edit automation data.

F4 will Capture All settings, create a new Cue, and put it at the end of the active Cue List (the Cue List selected in the Transport window).

F3 will Capture All settings, create a new Cue, and put it in the Cue Library.

F2 will Capture the differences between the current control settings and the settings for the subcues found in the selected Cue. It will only compare the settings of the subcues found in the cue.

F1 will Update the settings for the Subcues within the selected Cue. This is the only destructive edit in CueStation. There is no undo function for this, and all Cues that use the updated subcue will be affected.

These commands can be used in any CueStation window and are shown in the automation window menus.

Other

The shortcut keys ALT+0 through ALT+9, and CTRL+ALT+0 through CTRL+ALT+9, open various CueStation windows. The windows menu shows a list of the keys and the windows they open.

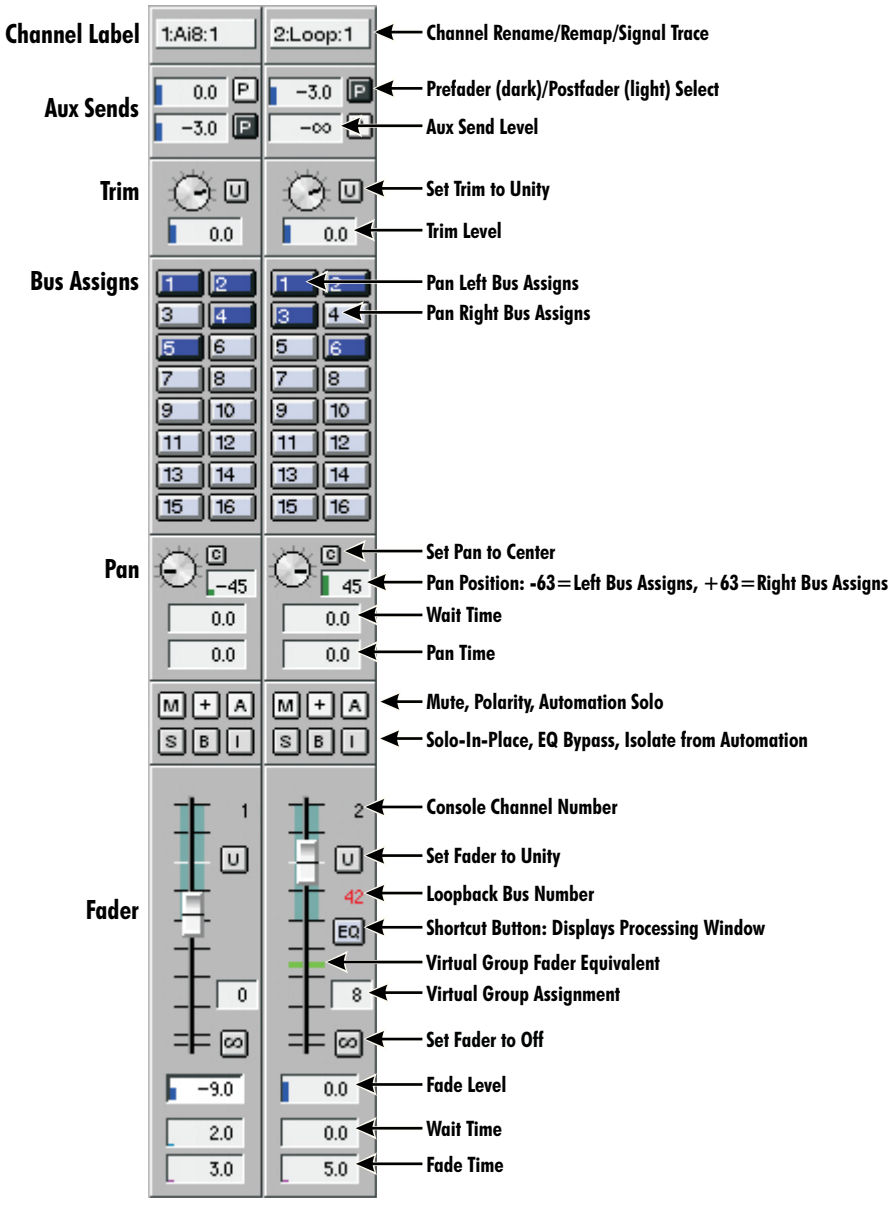
The F11 and F12 keys cycle forwards and backwards through the all windows, including closed ones; holding the SHIFT key simultaneously will close the current window before opening the next window.

Warning

You must press ENTER, TAB, or click on another data entry box when setting new values for most CueStation controls. As a visual indicator that you have not completed the data entry, you will see an ellipsis (...) after the value as long as the entry is pending. If you close the window or capture a subcue while the entry is pending, the old value will be used.

Basic Faders

There are many fader controls within CueStation; depending on their function, a fader will have some of the following controls:



For all entry boxes, you must press ENTER, TAB, or click on another entry box after typing the new value.

The **Auto** and **Solo** buttons (not shown) to the left of the faders are highlighted if any Automation Solo or a Solo-In-Place has been selected for any channel. Clicking the **Auto** or **Solo** button will select or clear the setting for all channels.

If you type a value for a control, be sure to press Enter to ensure the new setting is accepted.

Basic Cues

CueStation organizes your sound control data hierarchically.

- The ***Subcue*** is the basic building block for automation control. Each subcue contains a "snapshot" of the settings for a specific set of system controls. These system controls range from fader settings to bus trajectories, and provide detailed audio control.
- ***Cues*** hold a collection of subcues and, when triggered, adjust the system controls to match the settings specified by the subcues. If a subcue has specified delay or fade times associated with its system controls, these will be used to delay or fade the adjustment.
- ***Cue lists*** are automated, time-ordered collections of cues, with settings to trigger the cues manually, through time code, or automatically in a specified sequence.
- A ***Project*** can contain any number of used or unused subcues, cues and cue lists. You can import components from projects, which allows you to build "libraries" of common subcues, cues, and cue lists.

Projects contain
Cue Lists, which contain
Cues, which contain
Subcues, which have
Control Settings

This hierarchical construct provides a particularly flexible system, and makes it easy to tweak settings without destroying existing information. You can mix-and-match, rollback, or start again, without throwing out your earlier work.

Creating Cues

Most often, you will capture the settings of the system controls to a series of subcues, while simultaneously creating a new cue using those subcues.

When you capture the cue, you can specify which controls will be captured, and what to name the cue and its associated subcues.

If you need to change the settings used in an existing cue, refer to *Updating a Subcue* and *Modifying a Cue,* below.

1. Set your system controls — Console, Bus Levels, EQs, and so on — to the state you want them to be in after the cue is triggered.
2. In the **Cue Library** window, open the **Cue** menu and choose **New Cue & Cue Entry & Capture All.** When you capture the subcues, you can choose to capture every type of system control, or limit the capture to specific controls.

   The new subcues will appear in the Subcue Librarian window, while the Cue Library window will show the newly-created cue, which is also appended to the Cue List.

Tip

The F4 shortcut key works from any CueStation window. It creates a New Cue, adds this as a new Cue Entry at the end of the active Cue List, and performs a Capture All.

The F3 shortcut, **New Cue & Capture All** also creates a new cue; however, it puts the new Cue in the Cue Library only and does not place an entry in a Cue List.

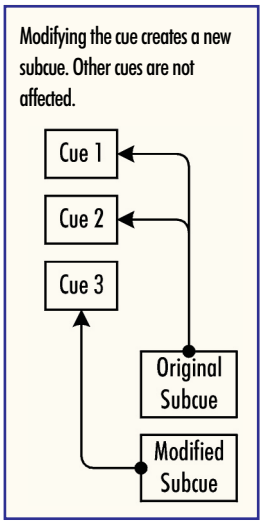
Building or Modifying a Cue List

The cue list is the sequence of cues that will be used in controlling the show. Each cue may be triggered manually, triggered by a time code value, or automatically triggered by a previous cue.

1. Bring the **Cue List** and **Cue Library** windows into the same workspace.
2. In the **Cue List** window, open the **Cue Lists** menu and choose **New Cue List.** If you're using an existing cue list, you can choose it from the bottom of the menu.
3. Right-drag cues from the **Cue Library** into the cue list. If you need to reorder cues within the list, you can right-drag them by their index values, dropping them in a new position.
4. By default, the new cue list entries will be set for **Manual** triggering.
   - To fire several cues simultaneously, or sequentially without further manual triggering, right-click the **Autofollow** entry for the leading cue. Then choose the cue that will follow it. If you provide a **Delay** value, the second cue will be fired after the delay time has elapsed.

     Note that if several cues are to be fired simultaneously, they should be placed together in the Autofollow list before any delayed cues. Delays are cumulative: cues following a delayed cue are triggered after the delay.
   - To fire a cue based on time code or at a specific time-of-day, right-click the **Trigger** entry for the cue and choose **SMPTE.** Enter the time code in the **Time** column for the cue.

Modifying a Single Cue — Capture Differences



If a particular cue is not performing as needed, you can adjust the system controls and modify its subcues. Other cues will continue to use the original subcue settings, while the modified cue will be given new subcues that reflect changes to the system controls.

If you wish to change the setting for all the cues that use a particular subcue, see *Modifying All Cues,* below.

1. In the **Cue Library,** double-click the **Cue ID** field to trigger the cue. The system controls will be updated as the cue runs.
2. Adjust the system controls as desired.
3. From the **Cue Library** window, open the **Cues** menu and choose **Capture Differences.** The subcues affected by the adjustments you made will be modified. You can provide a new name for those subcues, or you can allow CueStation to append an index number to the existing name.

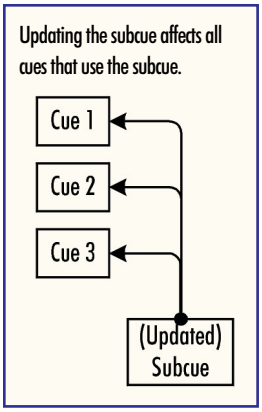
Tip

The F2 shortcut key will **Capture Differences** in any control window.

You can modify only the subcue types that are already listed within the cue.

If you need to add a subcue type to the cue, drag it in from the Subcue Library. If the right-hand pane for that type of subcue is empty, you will need to first create an instance of that subcue: from the **Subcues** menu, choose **New Subcue.** The new subcue will record the current system state.

Modifying All Cues — Update All Subcues



When you directly update a specific subcue, all cues that reference that subcue will use the new settings. This allows you to quickly and easily effect a global change. Use this with care: it also allows you to quickly effect a global mistake!

If you don't wish to change the subcue setting for all cues, see *Modifying a Single Cue,* above.

1. In the **Subcue Library** window, select the subcue you wish to globally update from the **Subcue Type** and **Subcue** lists.
2. Select the **Windows** menu, choose the appropriate application window for the type of subcue you are updating, and set its controls as desired.
3. Return to the **Subcue Library** window. From the **Subcues** menu, choose **Update Subcue.**

Tip

Use the F1 shortcut key to use **Update All Subcues** from any control window.

See Also

Understanding CueStation: Automation Techniques provides conceptual detail regarding automation.

Some Helper Apps support the creation of subcues. See *Basic SpaceMap* and *Basic Wild Tracks* for details.

Workspaces are a powerful tool for organizing the layout of your working windows and controls. See *Using BeOS: BeOS for New Users* for a brief explanation of the workspace concept and the keyboard controls associated with its use.

Basic Transport — Running a Show

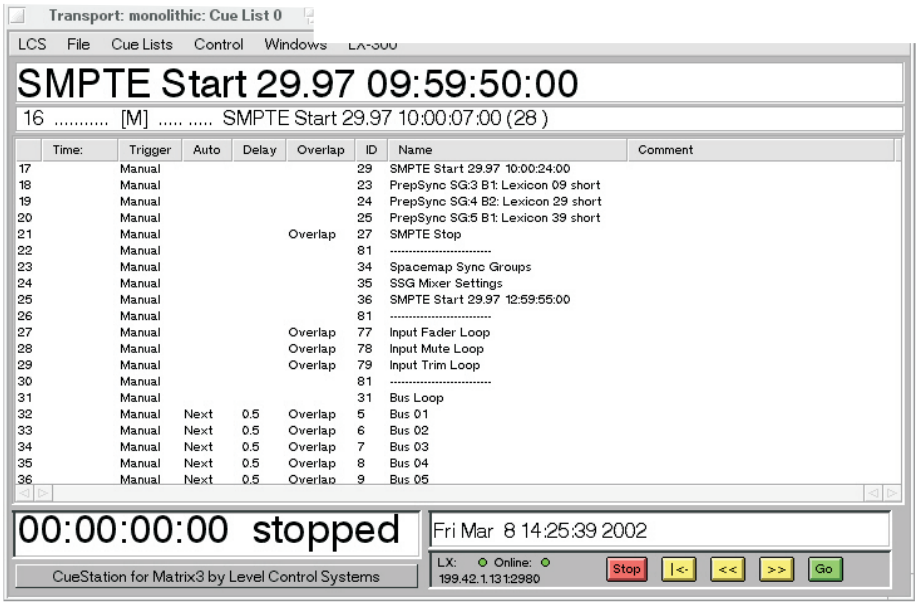
The **Transport** window has been designed to run a show from a cue list. It has a set of control buttons to move forwards and backwards through the list, and a **Go** button to fire the current cue-on-deck.

The most-recently triggered (”active”) cue is shown in large letters at the top of the window. This may be a cue from the active Cue List triggered by the **Go** button, Time Code, or Autofollow — or it could be a cue from the Cue Library triggered by an external control signal.

Below the active cue is the cue-on-deck. This is the cue in the cue list that will be triggered by the **Go** button. The next frame below the cue-on-deck shows the following cues in the active cue list.

To use the Transport:

1. In the **Transport** window, open the **Cue Lists** menu and choose a cue list. Press the yellow “rewind” button (|<) to make the first entry in the list the cue-on-deck.



2. Click the **Go** button to fire the cue-on-deck. This cue will then be shown in large print above the cue-on-deck field, and the next cue in the list will become the new cue-on-deck and will take its place at the top of the list.

   If time code is being used, time-triggered cues will be automatically fired without your intervention.

3. You can step through the cue list using the controls to the left of the **Go** button. Observe the change of the cue-on-deck as you use these controls: the **Go** button will always fire the current cue-on-deck.

Tip

The **LCS** menu contains some commands that can be very useful when you need to react rapidly — **Silence!** (CTRL+ALT+S) will mute the main system outputs by setting the Manual System Level to off. Please note that this will **not** mute any Aux sends as they are not controlled by the System Level or Manual System Level controls.

Basic EQ, Delays, and Compression

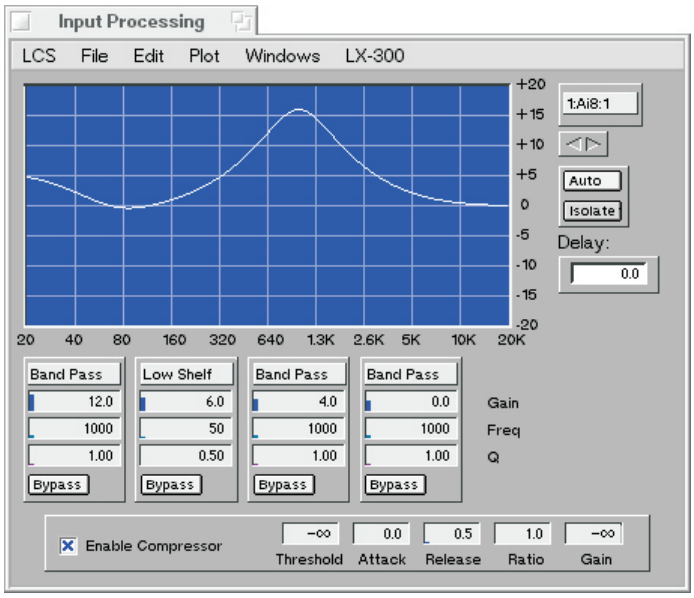
You can add up to eight bands of parametric equalization and a delay line to any input, output, or auxiliary output channel. CueStation supports bandpass, low-shelf, and high-shelf equalization. You can also add compression to a channel.

Enabling Processing

Equalization demands a lot of work from the Frame DSP, and is disabled by default. You can enable it on a frame-by-frame, card-by-card basis, eight channels at a time. To do this:

1. From any CueStation window, open the **LCS** menu and choose **Configure Mixer.**
2. For each Frame that carries a channel that requires EQ or delay:
   a. Select the Frame in the left-hand pane of the window.
   b. For the cards in the Frame that carry a channel that requires EQ or delay, select the **EQ** check box for the appropriate eight-channel block, and choose the number of channels of EQ that each channel in the block will use; and/or select the **Delay** check box and set the maximum delay time; and/or select the **Dynamics** check box.

      The **EQ** and **Delay** boxes are used to set the maximum number of channels and maximum delay time for all the channels for the card. To delay individual channels, you will use the Input Processing, Output Processing, and Aux Processing windows.



Setting EQ

EQ is available for input channels, output channels, and auxiliary output channels. To set the EQ for any channel:

1. Open the **Input Processing**, **Output Processing**, or **Aux Processing** window.
2. Select the channel, using the pointer arrows below the **Bus** box, or by right-clicking the **Bus** box and choosing the **Remap As...** command.
3. For each EQ band:
   a. Click the **Type** box to select bandpass, low-shelf, or high-shelf parametric equalization.
   b. In the **Gain** box, type a gain value (dB).
   c. In the **Frequency Center** box, type a frequency value (Hz).
   d. In the **Q** box, type a value for the bandwidth of the filter.

      For Bandpass filters with a positive gain setting, **Q** value is the standard EQ control (frequency/bandwidth).

      For Bandpass filters with a negative gain setting, the graph will show the bandwidth as calculated for the filter, but the numeric data entered is not in Q units. This will be corrected in a future release.

      For Low-Shelf and High-Shelf filters, the **Q** box sets the slope of the filter. The higher the value, the steeper the slope.
4. Press ENTER to ensure the values you have typed are accepted.

Notes

The **Bypass** buttons are used during setup to bypass a particular band, allowing a quick comparison of the settings. They are not stored in the EQ subcue, and any settings for that EQ band will be applied by a subcue recall.

The **EQ** button to the right of the fader is a shortcut to the Processing window for that channel. It will change color when the EQ is not set flat.

The **B** button above the fader is not automated and is a manual Bypass for the entire EQ. It will turn RED when the EQ has been bypassed and is not in the signal path.

Setting Channel Delay

1. Open the **Input Processing**, **Output Processing**, or **Aux Processing** window by pressing the **EQ** button next to the desired fader, or by using the **Windows** menu for the desired section.
2. If you need to select a different channel, use the pointer arrows below the label box, or use F7 and F8 to page through the channels.
3. In the **Delay** box, type the delay value (seconds). Press ENTER to ensure the value you typed is accepted.

Setting Compression

1. Open the **Input Processing, Output Processing,** or **Aux Processing** window by pressing the **EQ** button next to the desired fader, or by using the **Windows** menu for the desired section.
2. If you need to select a different channel, use the pointer arrows below the label box, or use F7 and F8 to page through the channels.
3. Select the **Enable Compressor** box. Set the compressor values:

   Threshold (dB) sets the input signal level at which the compressor begins to function. Signals below this level will not be affected.

   Attack and **Release** (mS) define how quickly the compressor starts once signal is above threshold, and how long it takes to return after the signal is back below the threshold.

   Ratio is the ratio of the output signal to input signal. This sets the amount of compression above the threshold. If set to less than 1.0, the output will expand it whenever the signal is above the threshold. (A setting of 2 means that the input must increase 2 dB for each 1 dB increase in output signal. Likewise, a setting of .5 means that for each increase of .5dB on the input, the output will increase 1 dB.)

   Gain (dB) boosts the signal after compression and is used to balance the level after processing.

Basic SpaceMap

SpaceMap provides powerful panning control surfaces that go far beyond the traditional left/right pan. With a SpaceMap, you can "fly" sound smoothly through any number of speakers, while the software automatically maintains the total power output at unity.

The SpaceMap *spatial pan control* distributes signal from its assigned input bus. As you move the spatial pan control around the map, it passes through triangular panning areas (*trisets*) defined by groups of *nodes*. The signal is distributed to the three surrounding nodes according to the current position of the pan control within the triset. With the appropriate design, you can set up three-dimensional sound movement, soundstage expansion, seamless fills — anything you can dream.

There are four kinds of nodes: Speaker, Virtual, Silent, and Derived.

- ***Speaker*** nodes represent physical audio outputs. The closer the spatial pan control is to a speaker node, the more signal that node will receive, while the other two nodes in the triset receive proportionally less.
- ***Virtual*** nodes receive signal as part of a triset, and distribute that signal among the nodes to which they are linked. By default, the distribution is equally weighted among all the nodes, but this can be changed. Virtual nodes are commonly used to break up a panning area into smaller trisets, to provide seamless transitions between trisets.
- ***Silent*** nodes discard their proportion of the signal: they're null outputs, like unplugged speakers. As the spatial pan control approaches the silent node, the other two nodes in the triset receive less signal. This is particularly useful for providing fade-out and fade-in around the perimeter of a map.
- ***Derived*** nodes represent physical audio outputs. They receive signal from the nodes to which they are linked. If they happen to be part of a triset, they discard any signal received from the spatial pan control. They're useful for fills and subwoofer sends.

To provide automated control of the movement of the SpaceMap spatial pan control, you can create a *trajectory*. The trajectory is a motion path that can be assigned to a subcue (with timing, repetition, scaling and other parameters). When a cue containing a *bus trajectory subcue* is recalled, the SpaceMap panning control follows its motion path, "flying" the sound through your speaker array.

Creating a SpaceMap

1. Go to the **SpaceMap** window. From the **Maps** menu, choose **New SpaceMap.**
2. From the **Bus** menu, choose the input bus.
3. Design the map:
   a. Add nodes to the map.
   b. Create the trisets.
   c. Link virtual and derived nodes.
   d. Test the trisets.
   e. Draw your trajectory path.
   f. Set the length of time the trajectory will take to pan.
   g. Test the trajectory.

   These steps are described in more detail, below.
4. Use the Trajectory in a cue list:
   a. From the **Trajs** menu, choose **Capture Spacemap Subcue.**
   b. If you want to adjust the parameters for the subcue, open the **Subcue Library** window, select the **Spacemap** subcue, and double-click the **Edit** column for the subcue you wish to change.
   c. Right-drag the subcue into a new or existing cue. See *Basic Cue Lists, Cues, Subcues* for further details.

Tip

The node types are represented by icons:

Speaker nodes: □

Virtual nodes: ◇

Silent nodes: ○

Derived nodes: ◯

Tip

Display a reference grid via the **Plot** menu. You can **Snap** to the grid, using various **Grid** settings.

You can zoom the scale of the window using the PAGE UP/PAGE DOWN keys.

Adding Nodes

The spatial pan control distributes the input bus signal to nodes. Nodes correspond to speakers or special signal routing controls.

1. Click the **Insert** button.
2. From the **Type** menu, choose the type of node that will be inserted.
3. Add the nodes by clicking on the map grid.

   If the node is numbered, the number represents the output bus channel. To change the output channel designation, select the node, type a new number, and press the ENTER key.

Modifying Nodes

1. Click the **Select** button.
2. Select the nodes you wish to modify. You can select a single node by clicking on it, or select several nodes by holding down the SHIFT key while clicking on them. You can also select multiple nodes by dragging a selection box around them.

   You may:

   - Drag a single to a new position.
   - Type a number to change the output channel assignment for a single node, and then press ENTER.
   - Choose a new node type from the **Type** menu to change single or multiple nodes.
   - Press the BACKSPACE key to delete single or multiple nodes.
   - Unlink the connection between virtual or derived nodes, and their associated speaker nodes.

Creating and Deleting Trisets

The spatial pan control distributes the input bus signal to the three nodes that form the triset enclosing its current trajectory position. The signal is distributed in proportion to the control's proximity to the nodes.

Creating Trisets

1. Click the **Triset** button.
2. Click inside any triplet of nodes; or select three nodes sequentially by clicking on a node and then holding down the SHIFT key while clicking on two other nodes.

   Do not overlap trisets: they won't work. The signal must be distributed to three nodes, not six. Unpredictable and somewhat bizarre panning behavior will result from overlapped trisets.
3. From the **Space** menu, choose **New Triset.**

Note

Trisets are shown using blue lines and are numbered sequentially. The numbers are for reference and do not correspond to matrix inputs or outputs.

Deleting Trisets

1. Click the **Triset** button.
2. Click inside a triset. The selected triset will be shown using bold lines. To select multiple trisets, hold down the SHIFT key while clicking.
3. Press the BACKSPACE key to delete the selected trisets.

Linking and Unlinking Virtual and Derived Nodes

Virtual nodes participate in trisets as if they were speaker nodes: the spatial pan control provides them a proportion of the input bus signal. They feed that signal to the nodes to which they are linked.

Derived nodes receive signal directly from the nodes they are linked to, and feed the summed signal to the matrix. If they happen to be in a triset, they participate as a silent node, discarding any signal proportioned to them.

Virtual nodes *receive signal from the spatial pan control* and send it to linked nodes.

Derived nodes *receive signal from nodes* and send it to matrix outputs.

Linking

1. Select the virtual or derived node, plus the other nodes to which it will be linked. For virtual nodes, these will be nodes that receive signal from the virtual node; for derived nodes, these will be nodes that send signal to the derived node.

   Hold down the SHIFT key and click on each node that you wish to include in the link.

2. Open the **Space** menu and choose **Link Virtual/Derived.**

Unlinking

1. Select the virtual or derived node, plus the other nodes that will be unlinked from it.

   Hold down the SHIFT key and click on each node that you wish to include.

2. Open the **Space** menu and choose **Unlink Virtual/Derived.**

Testing the SpaceMap

1. Set your system controls to provide signal to a specific bus.
2. Open the **Bus** menu and choose that bus from the list.
3. Click the **Test** button.
4. Drag the spatial pan control around your SpaceMap. If your speaker outputs are functional, you will hear the sound pan through the trisets.

   If you bring the Matrix window into the workspace, you can observe the output signal distribution change as you pan.

Note

The spatial pan control is represented as a small, solid square, with a number indicating its input bus assignment.

Creating a Trajectory

When you are satisfied with your triset layout, you can create a panning trajectory that can be used in your cue list.

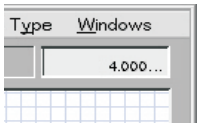
1. Open the **Trajs** menu and choose **New Trajectory.**
2. Click the **Record** button.

3. Draw the trajectory on the SpaceMap grid by clicking at its start position and dragging to its finish, or by clicking at several discrete locations along the desired trajectory path. CueStation will automatically connect the dots with line segments.

   The interval between dots corresponds to the actual time between clicks, or the speed at which you drag the mouse cursor.

4. Click the **Record** button again to stop recording.
5. In the entry box at the right of the button bar, type the duration of the trajectory.
6. You may re-record the trajectory by clicking the **Record** button again and drawing a new path.

Note



This is the unlabeled trajectory duration box. The ellipsis (...) indicates the new value hasn't been accepted yet: the user will need to press ENTER, TAB, or click another entry box.

Editing a Trajectory

1. Click the **Select** button.
2. Select individual points on the trajectory, or select multiple points by holding the SHIFT key and clicking on several nodes, or by dragging a selection box around them.

   You may:

   - Drag selected points to new positions.
   - Delete points by pressing the BACKSPACE key.
   - Insert points using **Insert Trajectory Point** (ALT+I).
   - Adjust the time duration of any segment of the trajectory.

   To adjust the time duration of a segment, select its start point and type the new value in the duration input box.

Tip

You can show the time values for every segment of the trajectory. From the **Plot** menu, choose **Draw Trajectory Node Times.**

Testing a Trajectory

1. Click the **Test** button.
2. From the **Trajs** menu, choose a trajectory.
3. Choose **Play Trajectory.**

   The trajectory will be played repeatedly. To stop playback, open the **LCS** menu and choose **Cancel Trajectories.**

Tips

Observing the Matrix window while panning will help you identify missing trisets or awkward transition zones.

If there are audible "glitches," check that the trisets are seamless, and consider using virtual nodes to provide a smooth fade for panning areas beyond the speaker output sets.

Both overlapping and missing trisets can create bizarre panning behavior, and can be difficult to identify. Counting the triset IDs can help track the problem down.

If the signal drops out completely while you are moving across a particular triset, make sure a triset has been defined for that triplet of nodes — surrounding trisets may give the appearance that a triset exists between them when it does not. Look for the triset ID number in the center of all trisets.

A trajectory that crosses any part of the SpaceMap window that is not inside a triset will result in a complete dropout of the signal. It is a useful practice to surround the main part of your SpaceMap with a "safety zone" of trisets that include one or two Silent nodes, with these Silent nodes along the perimeter of the window.

You can show the time values for every segment of the trajectory by opening the **Plot** menu and choosing **Draw Trajectory Node Times.**

Creating a Trajectory Subcue

After you have created and fine-tuned your trajectories (you can have more than one trajectory for a map):

1. Open the **Trajs** menu and choose **Capture Trajectory Subcue.**
2. If you want to adjust the parameters for the subcue, go to the **Subcue Library** window, select the **Spacemap** subcue type, and double-click the edit column for the subcue you wish to change.
3. The following settings are available:

   Trajectory chooses the trajectory that is used by the subcue.

   Map A and **Map B** are used by the **Pan** command, as described below. Map A corresponds to Pan=0.0. Map B corresponds to Pan=1.

   Repetitions sets the number of times the trajectory will be played when triggered by subcue. A value of 0.0 causes the trajectory to repeat endlessly, until interrupted by a SpaceMap or Matrix Row subcue, or a **Cancel Trajectories** command.

   The following values are *not* cumulative: they are not added (or multiplied) each time the trajectory repeats, but are applied only once and then maintained for all the repetitions.

   Rotate rotates the trajectory (measured in degrees). Positive values rotate clockwise: negative values, counterclockwise.

   X Scale and **Y Scale** enlarges or shrinks the trajectory from its original size. A value of 0.5 would reduce its size by half; a value of 2 would double its size. Negative values invert the trajectory.

   X Offset and **Y Offset** shift the trajectory from its original position. Positive values move it up or to the right: negative values, down or to the left.

Rate adjusts the playback rate as a ratio of its original speed. A value of 0.5 would play the trajectory at half-speed; a value of 2 would play it at double-speed.

Pan sets the trajectory interpolation value between **Map A** and **Map B;** with Map A represented by a value of 0.0, Map B by a value of 1, and values between 0 and 1 representing intermediate positions. This is primarily a placeholder for a future expansion of the SpaceMap.

Divergence distributes a portion of the signal to the triset, and the rest of the signal to all the other nodes in the SpaceMap: at a setting of 0.33, one-third of the signal will be "leaked" equally to all the nodes in the map, while the other two-thirds will be distributed to only the triset that the spatial pan control is passing through. A setting of 0.0 doesn't leak at all; a setting of 1 distributes the signal to every speaker equally, and leaves nothing extra for the current triset.

Level scales the fader level for the trajectory. A value of 1 represents unity: a value of 0.0 is off. This is primarily a placeholder for a future expansion of the SpaceMap and we recommend that you leave the value set to 1.

Override Pan, Override Divergence, and **Override Level** are placeholders for the future expansion of the SpaceMap. We recommend you leave the boxes un-checked.

4. Right-drag the subcue into a new or existing cue. See *Basic Cue Lists, Cues, Subcues* for further details.

Creating a SyncMap Subcue

The SyncMap subcue allows your trajectory to chase timecode. The trajectory will begin to play at a given timecode. The nodes within the trajectory will specify offsets from the start time.

1. From the **Trajs** menu, choose **Capture Trajectory Subcue.**
2. Go to the **Subcue Library** and right-drag the subcue into a new or existing cue. See *Basic Cue Lists, Cues, Subcues* for further details.

See Also

The *Understanding CueStation* chapter has an article that describes how to design and use SpaceMaps.

Basic Wild Tracks

Wild Tracks provides sixteen tracks of asynchronous, real-time, 32-bit, 48kHz digital audio playback on each LX-ELC EtherTracks module. Each track can be controlled independently and can be used alone or as part of a multi-track playback. Any track can use any audio file and the same file may be played simultaneously by multiple tracks even if using different starting points and durations.

WTX-HD Wild Tracks hard disk drives are normally attached to an LX-ELC EtherTracks card which provides support for all Wild Tracks functions. These drives can also be attached directly to the local computer running CueStation to import and export files. With the current software release you can not play back or record files on the local computer.

Setting Preferences

Please see *Configuring Ethernet,* page 5-50.

Note that Wild Tracks requires a network connection to the computer running Wild Tracks Editor. This is in addition to any connections between CueStation and the frame(s) with the LX-ELC Wild Tracks card(s).

Starting Wild Tracks Editor

1. Start Wild Tracks Editor.
2. Select a Wild Tracks volume. Open the **Utilities** menu and choose **Mount Disk.** The Wild Tracks window will show any recorded files on the volume.

You can now record, re-record, or modify the files.

Tip

If your Wild Tracks disk seems to be missing, open the **Utilities** menu and choose **Rescan SCSI.** This will find the hard drive volumes that are available to the system. If the disk is still missing, check your connectors, power, and device ID settings.

Basic Recording

To record files to a Wild Tracks disk, the disk must be connected directly to an EtherTracks card.

1. In the Wild Tracks window, click the **Record** button. A recording settings window will be shown.
2. In the recording settings window, provide the file details:
   a. Choose the recording **Device** (frame ID) and **Disk.**
   b. Type a name for the file in the **Name** box.
   c. Type a **Track Length** to specify the recording length, and specify the number of tracks to record. Tracks will be recorded simultaneously, from

the sequential channels (one channel per track) starting with the **First Record HSB Channel.**

d. In the **First Record HSB Channel** box, type the channel number for the first HSB recording channel. Tracks will be recorded from sequential channels, starting with this channel.

Tip

To determine the HSB channel assignments, backscroll the CueStation **Log** window to the point where the frame initialization is shown. This is only reported when the LX-300 frame is first started and CueStation is running.

3. Click **Prepare Record.** This will initialize the system, in preparation for recording.
4. Click **Record** (in the recording settings window) to start recording the file. Recording will stop when the track length is completed.
5. To re-record the file:
   a. In the Wild Tracks window, click **Record.**
   b. Select **Re-Record,** and choose the **File** that will be re-recorded.
   c. Click **Record.** Recording will stop automatically when the track length is filled.

Warning

Wild Tracks Editor writes the data structure information to the disk before it begins recording. If the recording is interrupted, the “length” of the file isn't changed, and playback will result in unpredictable audio signal being sent once the end of the properly-recorded section is reached.

Further, it is possible to create playback problems by editing file entries. For instance, if you record two files, and then edit the length of the first entry to make it longer, you'll end up overwriting part of the second file. When you play back the second file, it'll pick up part of the first file's audio track.

Keep it simple. Record your files in sequence.

Basic Importing and Exporting

To import an existing audio file, in AIFF, AU, SDII, or WAV format:

1. Open the **File** menu and choose **Import to Wild Tracks...**
2. Using the standard Open dialog window, find and open the file(s) you wish to import.
3. Select the recording **Device** and **Disk.**
4. Click **Import.** The file will be converted to 32bit, 48kHz format for Wild Tracks use.

To export a track to a standard audio file format, for external sound file editing:

1. Open the **File** menu and choose **Export From Wild Tracks...**
2. Select **Wild Tracks Disk** and **Wild Track File.**
3. Provide a **Name** for the region. Choose the **Sample Rate, File Type, Data Format,** and number of **Channels.** Check your audio file editing software manual, to find out what audio formats it can use.

If you wish to export only a portion of the file, specify a different **Start** and **Length.** By default, Wild Tracks Editor will export the entire file.

4. Click **Export** to export the file.

Defining Regions

Regions provide a way to play back a named portion of a file. A track can have up to ten regions. Alternatively, you can use subcues to play back unlimited, unnamed file portions.

To create a region:

1. Select a file or track by clicking on it.
2. Open the **Utilities** menu and choose **Define Region.**
3. Set the **Region Start Point** and **Region End Point** times. You can click the **Audition** button to play the region back, through channel 1 of the Wild Tracks frame, and readjust the times as necessary.
4. **Save** the region. It is ready to be used in Wild Tracks Editor subcues.

Basic Wild Tracks Subcues

There are three categories of Wild Tracks subcue, each with several variations: preparation, playback, and record. Prepare subcues are used to allow Wild Tracks to get ready to respond instantly to a play or record subcue. Play subcues play or stop playing the track; while record subcues are used to record a track when a cue is triggered. Wild Tracks subcues can be triggered manually (clicking the “Go” button, or as part of an autofollow sequence) or by time code.

The complete list of subcues is as follows:

Prep Subcues

PrepTracks — prepares a file for playback.

PrepRec — for future use (prepares a file for recording.)

Play Subcues

PlayTracks — plays the most-recently prepared file.

Stop — stops playback of of a set of Wild Tracks channels.

SyncOff — disables time code triggering of a sync group.

Prep and Play Subcues

Prep+Play — obsolete.

PlayNow — prepares and plays back a file as quickly as possible..

PrepSync — prepares a file for time code triggering. The file will be automatically triggered for playback at the specified time. Note that you must issue a set the time code clock rate at least once before triggering a PrepSync subcue. (See Creating a Wild Tracks Time Code Subcue, below.)

Rec Subcues

Record — for future use (records a previously-prepared file when the subcue is triggered.)

Creating a Wild Tracks Time Code Subcue

The Wild Tracks Time Code subcue initializes Wild Tracks in preparation for using time code triggering.

1. From **Subcue,** choose **New Time Code Subcue.**
2. Choose a **Clock Rate,** measured in frames per second (FPS) or dropped frame frames per second (DF FPS).
3. Choose the frame **ID.**
4. Provide a **Name** for the subcue.
5. Click **Save,** or click **New** to duplicate the window, allowing you to create another subcue based on the current subcue.

Creating a Wild Tracks Subcue

When you create a Wild Tracks subcue, you will be presented with a standard dialog window. The appropriate buttons and boxes will be enabled; those that don't apply to the type of subcue you choose will be disabled

1. Open the **Subcue** menu and choose **New Wild Tracks Subcue.**
2. Provide the subcue details, as appropriate for the type of subcue you are creating:

   Name allows you to provide a name for the subcue. Note that when you change the type of subcue, the name is renamed. It's best to choose the type and then provide a name.

   Type sets the type of subcue that will be created. Refer to Basic Wild Tracks subcues, above.

   File chooses the file from the Wild Tracks disk (see Box ID, below).

   Track Count is automatically set for the number of tracks in the file, and can not be changed. Tracks are sent to consecutive channels starting with the **First Playback Channel.**

   Box ID selects the Frame ID of the frame that has the Wild Tracks disk.

   Region allows you to choose a named region. See *Defining Regions,* above.

   First Playback Channel is a channel in the range of 1–16, and can be imagined to be like the physical inputs of an analog input card. Each Wild Tracks card has sixteen logical channels seen as inputs by its frame, through which file tracks can be played. Input channels can, of course, be assigned to arbitrary Console channels.

   The first playback channel identifies, with the track count, the set of logical channels used when playing the file. Each track will be sent to a separate logical channel; the channels are assigned sequentially, starting with the first playback channel.

   Only those tracks assigned channel numbers in the range 1–16 will be played. If you choose a first playback channel such that the track channel sequence goes past the sixteenth channel, those tracks will be missing.

   Start Sample ID reserved for future use.

Region Start and Region End allow you to play back a portion of a file. The default values play the entire file.

MTC Play Time specifies the time code trigger for a PrepSync subcue..

Sync Group ID sets the Sync Group ID for a PrepSync subcue (and can be used by a SyncOff subcue).

File Offset and File Length show the file start time and length. Do not change these values.

3. Click **Save** to save the subcue to the Subcue Librarian. The dialog window will close. From the Subcue Librarian, you can drag Wild Track Editor subcues to cues: they are treated the same as any other subcue.

   Click **New** to duplicate the Subcue Window, allowing you to create another subcue based on the current subcue.

   Click **Do Now** to test the subcue by triggering it.

Note

If your Wild Tracks disk is directly connected to a CueStation workstation, you will not be able to test your subcues. You must connect the disk to an EtherTracks card if you wish to audition the audio.

Basic Externals

Historically, the term “External” derives from the practice of controlling external devices — like hard disk playback units and samplers — that are connected to the LCS hardware. The term has grown to encompass control of Matrix^3 subsystems that are “external” to the central mixing functions of the LCS hardware. Examples include Hardware Status checks and CueConsole setup.

External Subcues are created from within the Subcue Library window using a step-by-step method. This is in contrast to the more “artistic” and intuitive method of capturing the mixer settings, in which you adjust the mixer until you like what you hear, and then use a function key (F3, F4) to create the subcue.

The advantage in creating External Subcues is that you gain direct access to the many different commands that may reside within any Subcue. These can range from simple MIDI commands to elaborate RIF-108 configuration settings, and using Externals gives you exact and comprehensive control of their values.

Creating an External Subcue

1. Go to the **Subcue Library** window. In the left-hand list of **Subcue Types,** select **External.** Open the **Subcue** menu and choose **New Subcue.**

   The **Subcue Edit** window will display details for the new subcue. Ignore it for the moment, and concentrate on the **Command Entry** window, which is where the commands are created.

2. Using the **Command Entry** window:

   a. Create a command by providing the following values:

      Type specifies the category of command.

      Command selects the specific command.

      Frame identifies the Matrix^3 frame to which the command will be sent. Choices are **All** or **Frames 1–32.**

      Port selects the communications port to which the command will be sent. Choices are **BoxNet** (for internal system commands), **A** (RS-422), **B** (RS-422), **C** (RS-232), or **MIDI.**

      Complete the command by providing values for the data fields that remain. These fields vary dependent on the command that you have specified.

   b. Click the **Add** button to add the configured command to the subcue.

      The Command Entry window will close, and the command will be shown in the list in the **Subcue Edit** window.

      Click the **Do Now** button to trigger the cue immediately. The Command Entry window will remain open. Do Now is useful for testing a command before adding it to the subcue.

      Click the **Cancel** button if you decide that you don’t want to add the command to the subcue. The Command Entry window will close.

3. Using the **Subcue Edit** window:

   a. Provide a **Name** for the subcue. You may add information to the **Comment** box, for future reference.

b. Click **Save Subcue** to add the subcue to the subcue library. The Subcue Edit window will remain open, allowing you to create additional External subcues. Be sure to give them unique names!

Click **Do Now** to trigger the subcue immediately. The Subcue Edit window will remain open. Do Now is useful for testing a subcue before adding it to the subcue library.

Click **New Subcue** to open a duplicate Subcue Edit window. The original subcue edit window will remain open. The duplicate window is useful for tweaking the commands without changing the original subcue.

4. To add additional commands to the subcue:

a. To add a command, open the **Commands** menu and choose **New.** Use the **Command Entry** window as before.

b. To edit a command, double-click it. Make changes using the **Command Entry** window. Click **Update** to apply the changes, and **Cancel** to throw them away.

Alternatively, you can edit the command directly in the Subcue Edit window by selecting it and using the **Command** menu. However, this has rather limited utility. Using the Command Entry window is generally easier.

5. Close the **Subcue** window by opening the **File** menu and choosing **Quit.**

Note

The Thumbnail information is not editable. It provides a condensed overview of the contents of the command, e.g. Note c1 n60 v96 d250 (MIDI Note 60 on channel 1, with a velocity of 96 and a duration of 250 ms.)

Example

Here is a step-by-step example showing how to create and use a single subcue that does three things:

- Enable CueMixer RIF-108
- Check Temperature
- Check Voltage

Create the External Subcue

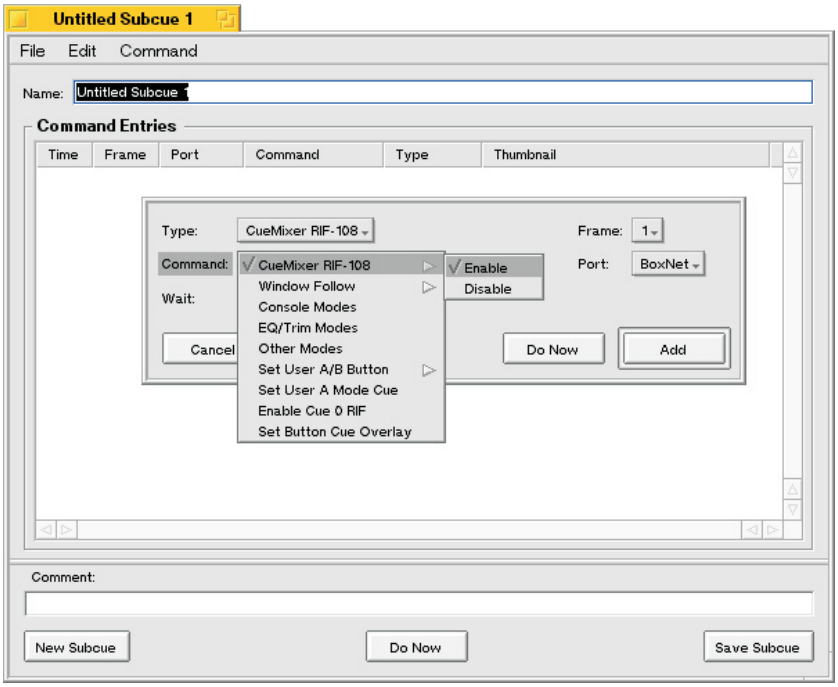
1. Open the **Subcue Library** window. From the subcue **Type** list, choose **External.**
2. From the **Subcues** menu, choose **New Subcue.** A Subcue edit window named **Untitled Subcue #N** and a **Command Entry** edit window will appear over top of it.
3. In the **Command Entry** window:

a. From the **Type** menu, choose **CueMixer RIF-108.**

b. From the **Command** menu, choose **CueMixer RIF-108>Enable.**

c. From the **Frame** and **Port** menus, select the frame to which your RIF-108 is connected, and BoxNet as the port (e.g. Frame 1, BoxNet).

d. Leave the **Wait** time at the default value of 0.0 seconds.



e. Click the **Add** button. The Command Entry window will close and the new command entry will be added to the list in the Subcue window.

4. In the **Subcue** window, open the **Command** menu and choose **New.** In the **Command Entry** window:
   - **Type:** Hardware Status
   - **Command:** Temperature
   - **Frame:** All
   - **Port:** BoxNet
   - **Wait:** `0.100`
   - **Add** the command.
5. Create another **New** command. In the **Command Entry** window:
   - **Type:** Hardware Status
   - **Command:** Check Voltage
   - **Frame:** All
   - **Port:** BoxNet
   - **Wait:** `0.200`
   - **Add** the command.
6. In the **Subcue** window, type `RIF Setup/System Check` for the **Name.**
7. From the **File** menu, choose **Quit.**

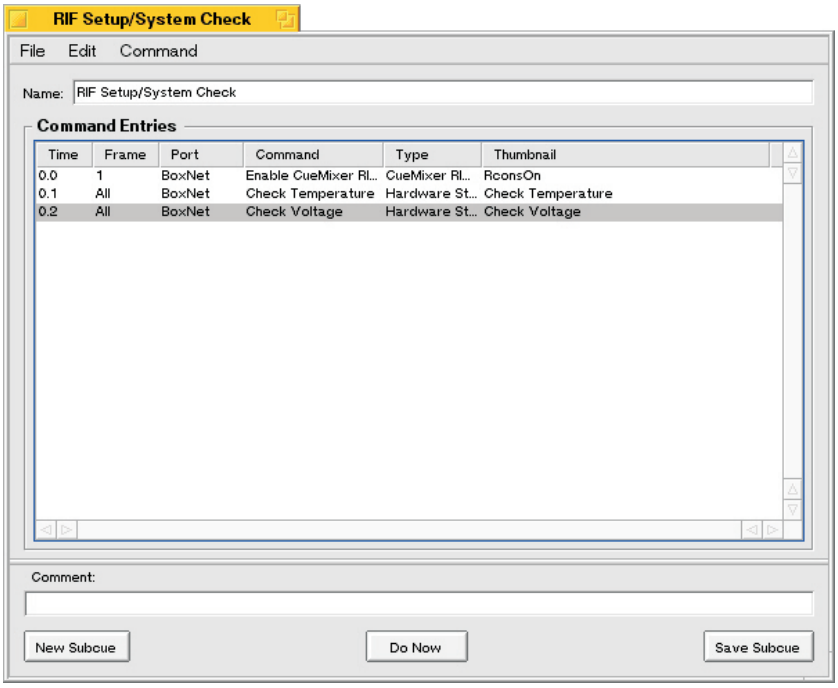
Use the External Subcue in a Cue

1. Go to the **Cue Library** window. From the **Cues** menu, choose **New Cue.** A Cue naming window will open.

   Name the cue `System Check` and click **OK.**

Right-drag the **RIF Setup/System Check** subcue from the **Subcue Library** window to the right-hand pane of the **Cue Library** window.

The cue is now ready to be used in a cue list. When triggered, it will enable your RIF-108, and check the temperature and voltages of every frame in the system.

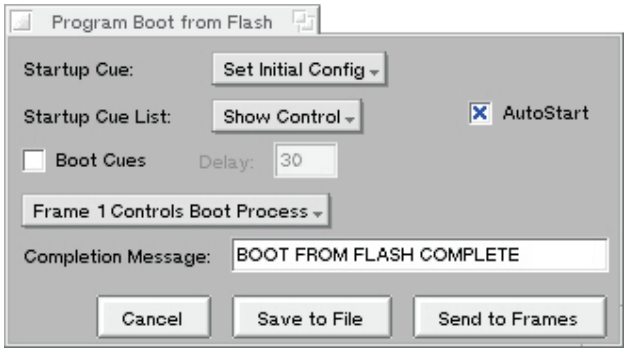


Using Boot Cues

The Matrix^3 system can be configured such that cue lists are stored in permanent memory. The cues in these lists can be triggered automatically when the system is powered-on.

Programming the Matrix^3 boot cues

1. From the **LCS** menu, choose **Program Boot from Flash.**
   - To fire a specific cue when the system is powered-on, select a cue from the **Startup Cue** menu.
   - To fire the first cue of a specific cue list (which can then auto-trigger following cues), select a cue from the **Autostart** menu.
   - To control which frame controls the boot process, select a frame from the **All Frames Boot Independently** menu.
   - If you want the frame displays to show a message after the boot cue is triggered, type the message in the **Completion Message** box.
2. To save this configuration, so that it can be sent to the frame again should the need arise, click **Save to File.** Find the project folder and give the file a name.
3. Click **Send to Frames** to have the Matrix^3 system automatically fire cue(s) when powered-on; otherwise click **Cancel.**



Configuring the Matrix³

From an automation programming perspective, CueStation treats the Matrix³ system as a monolithic device: you don't have to worry about which frame contains which channel, or specially construct your signal flow to keep input and output channels "together."

However, each LX-300 frame is a separate device, with its own main Digital Signal Processor (DSP) and dedicated memory. In order for the individual frames to work together as a system, they need to be configured cohesively, and told what type of processing to apply to the inputs and outputs hosted by the frame. CueStation is used to create and send this configuration to the frames.

The first time you use your system, you will need to configure some Matrix³ hardware: gain settings, IP addresses, and the like. You will also give CueStation a description of your Matrix³ system, so that it can correctly configure the frames. When you have completed this configuration, we recommend that you save it as a default. Then, when you start a new project, your defaults will be set properly.

Using the Configuration Window

Understanding that the DSP is a fixed resource in each frame is important. CueStation will help guide you to ensure that you're not asking the frame to do more work than it can do. You should configure the frame such that the DSP load does not exceed 100%. To help you, a load estimate is provided at the bottom of the window..

To configure your Matrix³ system through CueStation:

1. From the **LCS** menu, choose **Configure Mixer...**
2. For each LX-300 frame in your Matrix³ system:
   a. In the left-hand pane of the window, select the check box that matches the Frame ID.

   b. In the right-hand pane of the window, select and configure the cards installed in the frame. See *Card Settings,* below, for a description of the settings for each card.
3. Provide values for the settings at the bottom of the window:
   a. Select **Enable Sample Sync** to time-align all analog outputs of all frames within the system.

      If this setting is not checked, signals from the local frame inputs will arrive at the local outputs 32 samples before the signals from the inputs on other frames. This is .6 mS (.0006 seconds) - about the length of time sound takes to travel 30cm (about 1 foot) in the air.

      This option will use about 7% of your DSP resource on each frame. Currently it can only be enabled when dsp execution is limited to each local frame.

   b. In the **Buses** box, type the total for the number of matrix buses used by the system.

   c. In the **Assigns** box, type the number of bus assign switches per Console input.

d. In the **Vgroups** box, type the number of Virtual Group faders.

4. Click **OK.**
5. From the **LCS** menu, choose **Send Configuration,** to synchronize Matrix3 with CueStation.

Tip

Create a new project, define the configuration, and save the “empty” project file. You can then use this file as a template for new projects. It will already contain the correct hardware information; you can then easily tweak the settings that might change between projects

Common Settings

The following configuration options are common to almost all cards:

Channel Assignments: sets the channel number used within CueStation to refer to the specific physical inputs associated with the card. Channel numbers are referenced throughout CueStation, and are fundamental to controlling your signal flow. Channel assignments do not have to be sequential; however, they should not be duplicated within the system.

Duplicate channel numbers will be highlighted, and should be fixed.

EQ: select to enable EQ for each group of eight channels. Click the button beside the check box to select 1 to 8 bands of parametric equalization.

Delay: select to enable delay.

Dynamics: select to enable compression.

Mix Point Limit: This specifies the maximum number of inputs that the DSP can mix to the outputs on that frame. As this value is increased,the workload on the DSP increases.

A mix point is one input to one output, and is calculated by taking the number of inputs being mixed to each output on the frame, adding them, and rounding up to the next highest multiple of eight.

For example: 8 outputs each mixing 4 inputs would be 32 mix points. This is the same even if matching inputs are used for all outputs.

Meter On: selects which display panel, on this frame, will be used to display metering for this group of inputs or outputs.

Execute On: This selects the dsp in a system that will be used to process the audio for a module. You can use the new LX-EXP Expansion dsp module, or the dsp in another linked frame. This allows the work load to be balanced between multiple dsp's in a system.

Tips

Click the > button between the input channel boxes to automatically sequence the numbers.

Commands on the **Settings** menu apply to the entire system, all frames:

To obtain an intial configuration (identifying frames and cards), choose **Query Hardware Config.** The workstation must be connected to the frames, and they must be using firmware v020115 or greater.

To number channels automatically, choose **Renumber All Channels.** All channels for all frames will be numbered sequentially.

To automatically balance the DSP load for all frames, choose **Load Balance.** The submenu choices select which frames or groups of frames will be balanced. Plugins are balanced only when you select **System Plugins.**

The DSP load meters shown at the bottom of the window should not exceed 100%. CASL tick performance is degraded above 100% capacity, and Frames with 110% or greater workload will not perform all the mixing and processing assigned to them.

Frame Settings

When you select a frame in the left-hand pane, a list of its installed cards is shown in the right-hand pane. Each card is configured separately, as follows:

Analog Input

Range Setting: use the range setting drop-down menu to choose +6dB, +16dB, or +26dB scale for each channel. This sets the analog value in dBu that will be used by the A/D convertor for full scale conversion.

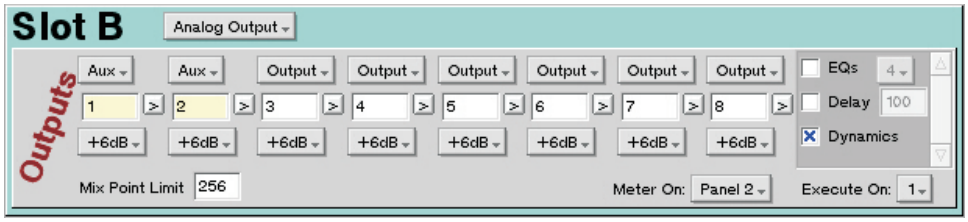
Channel Assignments, EQ, Delay, Dynamics, Mix Point Limit, Meter On, and **Execute On** settings are described in *Common Settings,* page 5-30.

Analog Output

Channel Type: click the buttons above individual output channels to choose **Output** or **Auxiliary Output.**

Range Setting: use the range setting drop-down menu to choose +6dB, +16dB, or +26dB scale for each channel. This sets the analog value in dBu that will be used by the A/D convertor for full scale conversion.

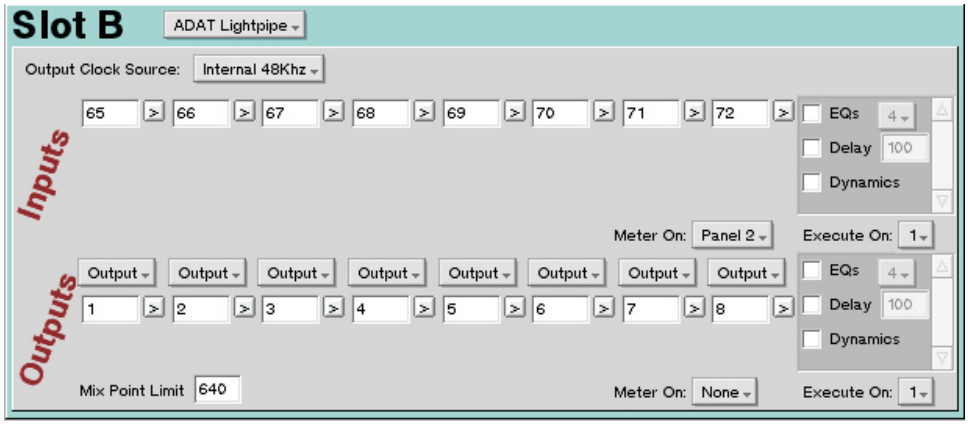
Channel Assignments, EQ, Delay, Dynamics, Mix Point Limit, Meter On, and **Execute On** settings are described in *Common Settings,* page 5-30.



ADAT Lightpipe

Output Clock Source: choose **Internal 48kHz,** or **External ADAT Input.**

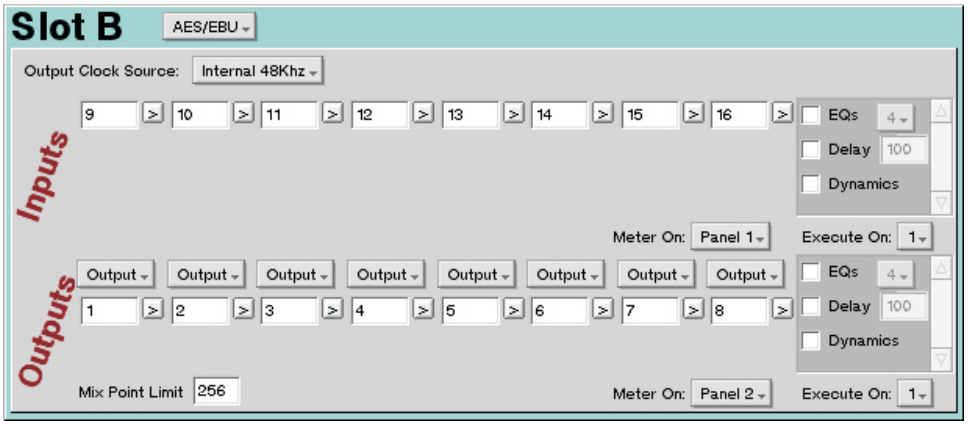
Channel Assignments, EQ, Delay, Dynamics, Mix Point Limit, Meter On, and **Execute On** settings are described in *Common Settings,* page 5-30.



AES/ABU

Output Clock Source: choose **Internal 48kHz**, **First AES Pair**, or **Each AES Pair.**

Channel Assignments, **EQ**, **Delay**, **Dynamics**, **Mix Point Limit**, **Meter On**, and **Execute On** settings are described in *Common Settings,* page 5-30.



CobraNet

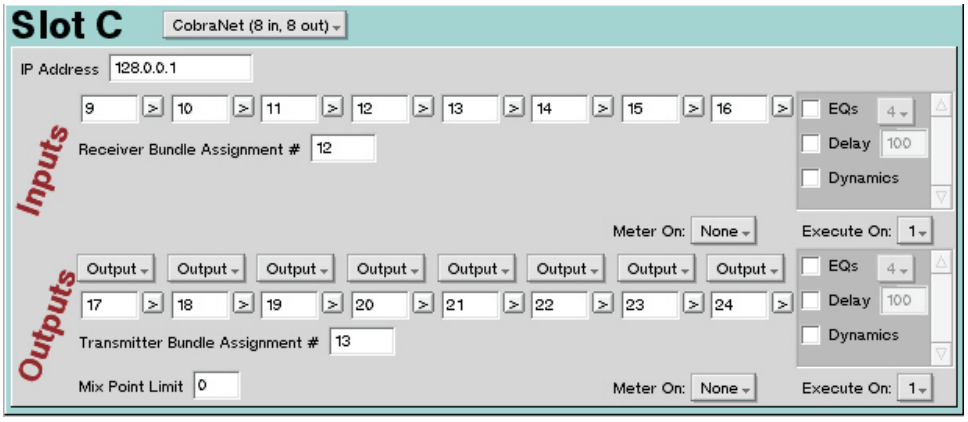
Please refer to *Configuring CobraNet,* page 5-51.

IP Address: the IP address for the card.

Bundle Assignment #: see *Configuring CobraNet.*

Channel Type: click the buttons above individual output channels to choose **Output** or **Auxiliary Output.**

Channel Assignments, **EQ**, **Delay**, **Dynamics**, **Mix Point Limit**, **Meter On**, and **Execute On** settings are described in *Common Settings,* page 5-30.

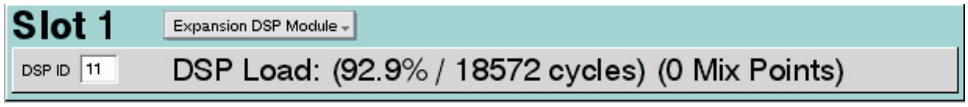


DSP (Slot 0)

DSP ID: Same as the Frame ID. Can not be modified.

Expansion DSP

DSP ID: This is the ID for the dsp and must be a unique number, not used by a frame or another expansion DSP.



Link

There are no settings for link cards.

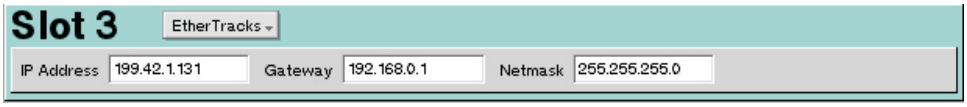
EtherTracks

Please refer to *Configuring Ethernet,* page 5-50.

IP Address: the IP address for the card. The default is `192.168.0.`*n,* where *n* is 100 + the Frame ID.

Gateway: use the default setting, or leave blank.

Netmask: set to 255.255.255.0. Sets the type of addressing used by the network.

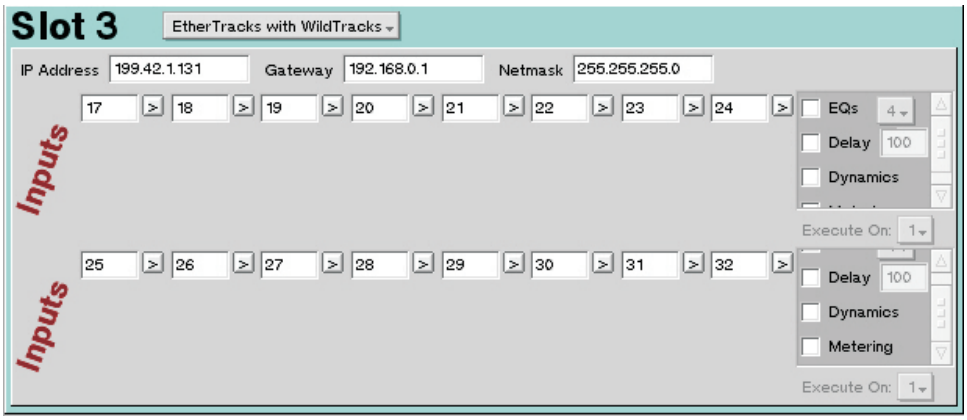


EtherTracks + Wild Tracks

Please refer to *Configuring Ethernet,* page 5-50.

Metering: select to enable NetMeters for this module.

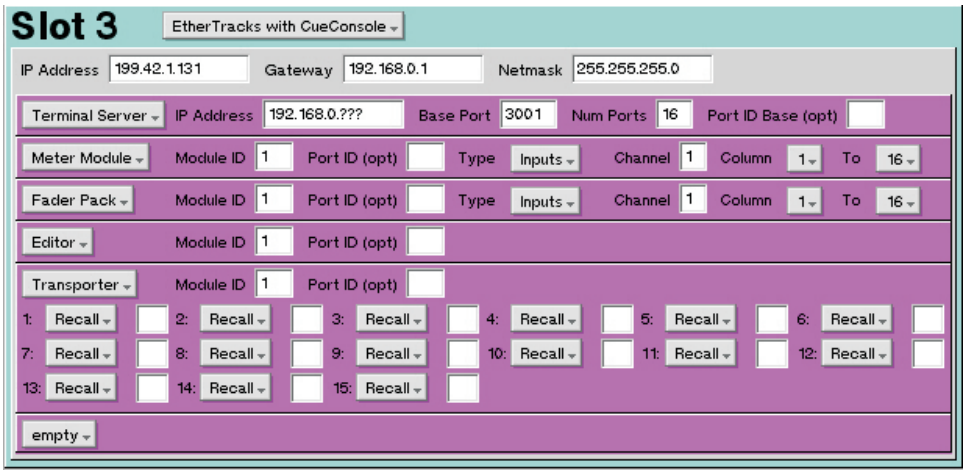
Channel Assignments, EQ, Delay, Dynamics, Mix Point Limit, Meter On, and **Execute On** settings are described in *Common Settings,* page 5-30.



EtherTracks + CueConsole

Please refer to *Configuring Ethernet,* page 5-50.

Please refer to the CueConsole documentation.



EtherTracks + Wild Tracks + CueConsole

Please refer to *Configuring Ethernet,* page 5-50.

As for EtherTracks + Wild Tracks. For the CueConsole settings, please refer to the CueConsole documentation.

Metering: select to enable NetMeters for this module.

Channel Assignments, **EQ**, **Delay**, **Dynamics**, **Mix Point Limit**, **Meter On**, and **Execute On** settings are described in *Common Settings,* page 5-30.

Comm/Synx (Slot X)

There are no settings for comm/sync cards.

Plugin Settings

At the bottom of the frames list, in the left-hand pane, is a Plugins entry. Plugins are software extensions to the Matrix³ hardware, enabling it to perform additional processing functions.

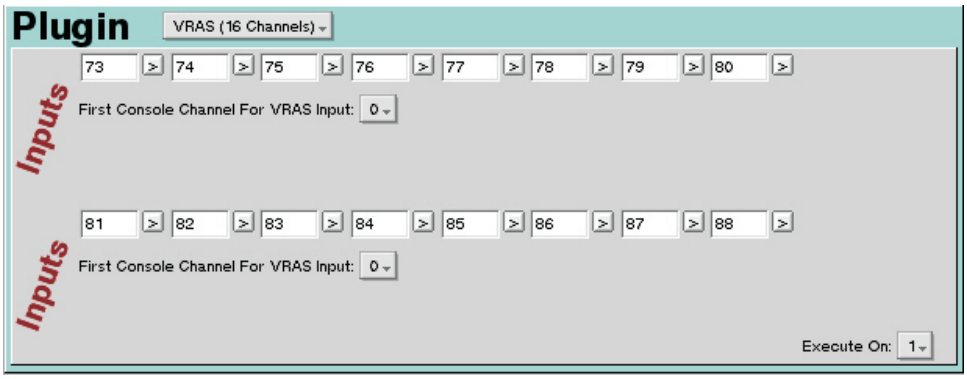
VRAS

Channel Assignments and **Execute On** settings are described in *Common Settings,* page 5-30.

First Console Channel for VRAS Input: specifies the input module used for the first microphone; other microphones must be patched into the same module using sequential channels following this channel (i.e., mic 1 patched to module Ai8 input 1; mic 2 patched to Ai8 input 2, ; et cetera.)

Note

Microphones must be connected to an external microphone pre-amp. The Matrix³ analog input is line level.



Loopback

Loopback allows you to route CueStation outputs back into CueStation inputs without using i/o modules. This provides a way to do inter-matirx delays and group signal processing and manipulation.

Inputs: This creates a set of 8 input modules in the console window that will be used for the Loopback return. The default name for the input will include the word "Loop" and the source Output channel will be shown with red lettering to the right of the fader.

CAUTION

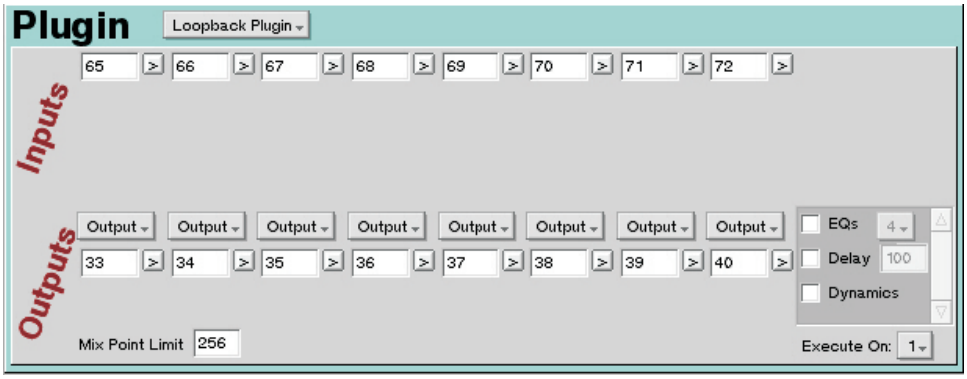
The system will allow you to route a Loopback return to the same Loopback send, creating a feedback loop that will oscillate at full volume.

Channel Type: use the drop-down menu above individual output channels to choose **Output** or **Auxiliary Output.**

Outputs: This creates output channels that are used to send a signal back to the Loopback return in the console window.

A Loopback send channel looks like a regular output except that the default name contains the word "Loop" and the console return channel number is shown in green next to the fader.

EQ, **Delay**, **Dynamics**, **Mix Point Limit**, and **Execute On** settings are described in *Common Settings,* page 5-30.



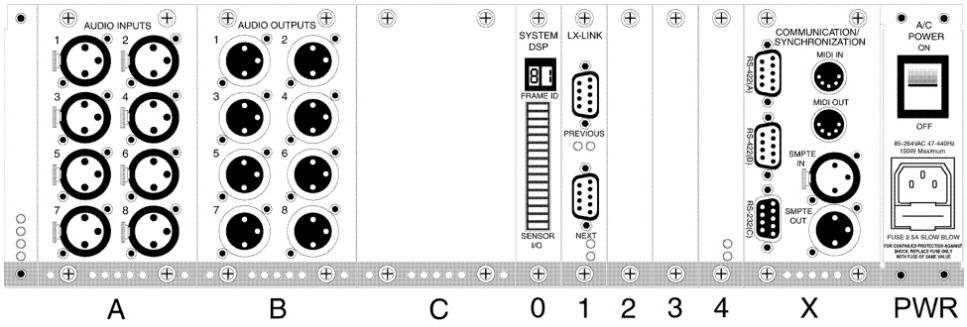
Diagnostics

Warning

Diagnostics must only be used under the direction of LCS technical support. Under no circumstances run the diagnostic program with amplifiers and speakers connected to the frame outputs.

Valid Slot Assignments

The LX-300 has specific slot assignments for most cards. These assignments are:



LX-300 Chassis Slot Assignments	
Slot	**Cards Allowed in this Slot**
A	LX-AI8 analog audio input *LX-AO8 analog audio output *LX-ADAT digital audio input/output *LX-AES digital audio input/output *except if LX-AO8 is installed in Slot C
B	LX-AI8 analog audio input LX-AO8 analog audio output LX-ADAT digital audio input/output LX-AES digital audio input/output
C	LX-CBR CobraNet audio interface *LX-AO8 analog audio output * except if LX-AO8, LX-ADAT, or LX-AES is installed in Slot A
0	LX-DSP system processor
1, 2, 3, 4	LX-LNK frame link interconnect* LX-ELC EtherTracks/Wild Tracks* LX-EXP expansion DSP (maximum 3; two if LX-ELC installed) LX-VRA variable room acoustics system© *maximum one per frame
X	LX-COS communications/synchronization

See Also

Chapter 3, *Understanding CueStation: CueStation and the Matrix³*.

Setting Up the Network

These instructions assume that BeOS has correctly detected your network card, that you are connecting to a private local network, and that the network will be used exclusively for CueStation and its Helper Apps. If any of these conditions is not true, you will need to contact your network administrator for assistance.

Configuring the CueStation Host Computer

The CueStation host computer will be the computer that is directly connected to the Matrix[3] system using a serial port.

1. From the BeOS Deskbar **Preferences** submenu, choose **Network Preferences.**
2. Select your network card from the **Network Interfaces** list.
3. Select **Interface Enabled.**
4. Click **Specify Settings.**
   a. In the **IP address** box, type `192.168.0.100`
   b. In the **Subnet mask** box, type `255.255.255.0`
   c. Leave the **Gateway** box blank.
   d. Click **Done.**
5. Click the **Save** button and then **Restart Networking.**
6. Close the preferences dialog.

Note

The IP and Subnet addresses given above are generally suitable for simple Matrix[3] configurations. If your system is complex — using multiple networks, for example — you may need to use other addresses.

For Other Workstation Computers

These computers will host the CueStation Helper Apps (Wild Tracks Editor, Cue List Editor, etc.). They will not be directly connected to the Matrix[3]; instead, they will communicate with the CueStation host computer using the network.

If you are using EtherTracks, please refer to *Configuring EtherTracks,* below, before setting up the workstation computers.

1. Follow the procedure in *For the CueStation Host Computer,* above, with the following differences:
   a. In the **IP address** box, type 192.168.0.*n,* where *n* is unique for each workstation computer. We suggest numbering the workstations sequentially, counting down starting with “99” (IP address .100 was used for the host computer, and the EtherTracks modules count up starting with .101).
   b. In the **Subnet mask** box, type `255.255.255.0`

c. Leave the **Gateway** box blank.

d. Click **Done.**

e. Click the **Save** button and then **Restart Networking.**

f. Close the preferences dialog.

2. Open **Subcue Librarian.** Open the **File** menu and choose **Preferences.**

   In the **CueLibServer Host** box, enter the IP address of the CueStation host computer (192.168.0.100, if you used the value provided in the previous procedure).

Verifying Network Operation

You can check that the network is functioning by "pinging" the other computers on the network.

1. Open the BeOS Deskbar **Applications** submenu and choose **Terminal.**
2. Type '`ping 192.168.0.`*n*' where *n* is one of the numbers used in setting the IP address for the CueStation host computer or the other workstations. Press the ENTER key to issue the command.
   - If there is no response, or if the message '`ping timed out`' appears, your network is nonfunctional.
   - If a series of nearly identical messages starting with the text '`32 bytes from...`' appears, your network is functioning correctly.
3. Press CTRL+C or close the terminal window to stop the ping command.

Advanced Networking Tips

If you are connected to other networks, the following tips may prove useful:

- If the computer connects directly with the Internet, enter the **Gateway** or **Proxy Server IP** address in the **Settings** window.
- If you wish to use FTP or Telnet for data transfers, you can enable them through the **Services** tab of the Networking Preferences window.

Network Troubleshooting

If your network is broken, investigate the following:

- That the equipment is powered-on!
- That your cable connections are secure.
- If using a hub, that your cables are plugged into the correct ports.
- If not using a hub, that you are using crossover cables.

Configuring Ethernet

These instructions assume that your BeOS networking is up and working. See *Configuring BeOS Networking* for details.

Configuring EtherTracks Cards

Your EtherTracks cards are part of your network, and must be configured to use the network before the software can communicate with them.

1. Run CueStation. Open the **LCS** menu and choose **Configure Mixer.**
2. From the left-hand listing of frames, select the frames that contain EtherTracks cards and:
   a. For the slot that the card is installed in, choose Ethernet (“EtherTracks”) or Wild Tracks (”EtherTracks with Wild Tracks”).
   b. In the **IP address** box, type `192.168.0.`*n,* where *n is 100 + the Frame ID.*
   c. Leave the **Gateway** box blank.
   d. Enter `255.255.255.0` in the NetMask box.
   e. Open the **Settings** menu and choose **Renumber All Channels.**
3. Click **OK.** You will be asked **Send Configuration & Mixer Settings?** Click **Send.**
4. After the configuration has been sent, restart your Matrix^3 system to assign the new IP addresses to the EtherTracks cards.
5. Run Wild Tracks Editor. Open the **File** menu and choose **Preferences.**
   - For the first LX-ELC on LX-300 number 1, enter the ip address of that module. Continue and list all other LX-300 Ethertracks addresses in the appropriate boxes.

Note

EtherTracks and CobraNet cards can use the same IP addresses: they are on separate networks and will not interfere with each other.

The IP and Subnet addresses given above are generally suitable for simple Matrix^3 configurations. If your system is complex — using multiple networks, for example — you may need to use other addresses.

Ethernet Troubleshooting

If Ethernet appears to be broken, investigate the following:

- That the equipment is powered-on!
- That your cable connections are secure.
- If using a hub or a network switch, that your cables are plugged into the correct ports and that the connect LED on the port lights.
- If you are making a direct connection and not using a hub or network switch, that you are using a crossover cable.

Configuring CobraNet

CobraNet provides 20bit/48kHz digital audio data transmission over Ethernet networks. It is a common networking standard for interconnecting commercial digital audio products.

Configuring CobraNet

1. Power-on the CobraNet network switch.
2. Start CueStation but do not send configuration or project. This allows the Log window to capture messages from the Matrix³ frames as they start.
3. Power-on the Matrix³ system.
4. Observe the Log window for an entry announcing that the Cobranet cards are starting.
5. Open the **LCS** menu and choose **Configure Mixer.**
6. For each frame containing an LX-CBR CobraNet interface card, select it on the left-hand pane. For Slot C of that frame, choose a CobraNet configuration type: 0, 8, or 16 inputs, and 0, 8, or 16 outputs.

   Configure the card. An overview of settings is provided in *Configuring the Matrix³*, page 5-29.

   a. Assign an **IP address** to the card. While CobraNet does not require an IP address to work, most administration software will require a unique IP for each card.

      We suggest numbering the cards using `192.168.0.`*n,* where *n* is 200 + the Frame ID.

   b. Select the **Channel Assignments** and **Channel Type** for the card.

   c. Assign **Bundle Assignment #** numbers to each set of eight channels. Refer to Bundle Assignments, below.

   d. Set **Mix Points**, **EQs**, **Delay**, **Dynamics**, **Meter On**, and **Execute On** as appropriate. See the descriptions for other cards in *Configuring the Matrix³* (page 29).

Note

CobraNet cards do not have to be connected to a CobraNet to be configured by CueStation.

EtherTracks and CobraNet cards can use the same IP addresses: they are on separate networks and will not interfere with each other.

The IP and Subnet addresses given above are generally suitable for simple Matrix³ configurations. If your system is complex — using multiple networks, for example — you may need to use other addresses.

Bundle Assignments

Digital audio data is transmitted in the CobraNet network in “bundles” of up to eight channels. Each bundle is assigned a unique identification number.

Hub-Based Networks

Due to compromises in network performance, we do not recommend using hub-based networks for CobraNet.

Switch-Based Networks

The advantage and challenge of CobraNet on a switched based network is the amount of data that you can distribute. By using the point to point addressing of Unicast, and careful layout of the network you can have large numbers of channels available. There are limits however.

On a 100base-T network, you should have no more than four multicast bundles. This is because by definition a Multicast will be sent to all nodes on a network. Multicast bundles are usually assigned IDs of from 1 to 255.

Note: A bundle ID of 0 is used to turn off a Transmitter or a Receiver.

Using Unicast you can send a bundle from a Transmitter to a Specific number of receivers. Normally Unicast is one Tx / Rx pair, but it is possible to have multiple receivers for a single Transmitter by defining the number of receivers permitted access to the Transmitter. Unicast bundles are usually assigned an ID in the range 256–65279. Both the transmitter and receiver(s) will have the same bundle ID number.

If you have more than the defined number of Unicast Receivers for a Transmitter bundle, the connections will be made according to the order that the receivers were detected on the network. If a receiver is removed and there is a receiver with a duplicate bundle ID, then the network will start sending the bundles to that receiver.

It is also possible to set more than one Transmitter to the same bundle ID. The Conductor will only allow one Transmitter to be active at a time, but should that Transmitter be removed from the network, then the redundant Transmitter will be made active.

Control of Transmitters and Receivers is done by the CobraNet Conductor. Any CobraNet device can be a Conductor. The CobraNet will determine who has priority and will use one device until it is removed from the network, at which point the Conductor status will be renegotiated and a new Conductor established automatically. It is possible to rank devices in the order that you would like them to be used as Conductors using the Conductor Priority field. The Conductor with the highest Priority is the one used by the Network. You can have duplicate priority settings and the network will automatically chose a Conductor.

Tip

A wealth of CobraNet information can be found at the Peak Audio home page: http://www.peakaudio.com/CobraNet/papers/Bundle_Assignments.html

Using CobraNet

Once configured, CobraNet is transparent to the CueStation user. Console input and output channels work as expected, and can be used in the mix just as ordinary analog signals would be used.

See Also

Peak Audio originated the CobraNet protocol, and their web site is the best place to find both basic and detailed information on using CobraNet. The following

addresses were correct during the writing of the manual; if there have been changes to the structure of the site, start at the home page.

Peak Audio home page:
http://www.peakaudio.com/

Quick overview: http://www.peakaudio.com/CobraNet/Background.html

Frequently asked questions: http://www.peakaudio.com/CobraNet/FAQ.html

Bundle assignments:
http://www.peakaudio.com/CobraNet/papers/Bundle_Assignments.html

Network design: http://www.peakaudio.com/CobraNet/Network_Design.html

Installation support: http://www.peakaudio.com/CobraNet/Support_Install.html

Collecting Files for Troubleshooting Assistance

To help our technical support staff help you, we will need several files from your system, so that we can identify problems in software, hardware or with show control. These files are:

- The CueStation Project file.
- Log files from `/beos/home/LCS_Logs`. These should include files from the sessions with problems, as well as any files you might have when the system was working correctly.
- If you have been instructed on how to do a "stack crawl", then we will need a copy of the `.txt` file created as part of the process.
- If you are using Wild Tracks, we will need an archive of the SCSI drive directory.

Multiple files can be saved in a single file by archiving them, using the file compression utility built into BeOS named Zip-O-Matic. The zip format also compresses the files, significantly reducing the size of the archive without damaging its contents and allowing it to be safely sent via email without loosing its attributes.

Creating the Archive

The following procedure will help you in creating the troubleshooting archive.

1. Right-click the Desktop and choose **New Folder** (ALT+N).
2. Open the `/beos/home/LCS_Logs` folder. Click the **Modified** column header, to sort the files by date. Copy the three most-recent log files to the new folder.
3. Open the `/beos/home/LCS_Apps` folder. Copy the `nameOfStackCrawl.ext` file to the new folder.
4. Find your CueStation Project file and copy it to the new folder.
5. Click on the folder name; after a second, you'll have the opportunity to rename the folder. We suggest giving it the show name.
6. Right-click the folder. Open the **Add-Ons** submenu and choose **Zip-O-Matic** (CTRL+ALT+Z). A new file will appear on the Desktop, using the folder name and a "`.zip`" extension. This is your archive file, ready to be e-mailed to LCS.

   You may delete the folder, as its contents are duplicated in the archive file.

Note

To Copy a File: Right-click the file icon, then drag it to its destination. When you release the file, choose **Copy.**

Directories Quick Reference

The shorthand notation for a folder location uses forward-slash symbols to separate the folder names. The names are listed in *folder/subfolder/subsubfolder* order.

You can find these folders using the BeOS Tracker or the File Dialog, as described in a separate task reference.

Common CueStation Folders

- The applications (i.e., CueStation and Wild Tracks) are stored in `/beos/home/LCS_Apps`
- Application settings are in `/beos/home/config/Settings/LCS_Settings`
- Help files are in `/beos/home/LCS_Apps/Help`
- The log files are in `/beos/home/LCS_Logs`
- CueStation automatically keeps backups of your work in `/beos/home/LCS_Backup`. Each time you quit CueStation, a new backup is created; there will be up to ten backups, with the oldest being removed when a new one is created.

Common BeOS Folders

- User applications (i.e., the programs you add to BeOS) are generally in `/beos/apps/`*name of application*
- BeOS installs several useful applications and utilities in `/beos/beos/apps`
- BeOS installs its own help files in `/beos/beos/documentation`

Note

If you are accessing files through the Terminal application or in NetPositive, pathnames begin with `/boot`, not `/beos`.

Accessing Files in an Archive

When a collection of files must be distributed by e-mail or floppy disk, the collection is often collected into a compressed (size reduced) archive. To use the files, the archive must be *expanded*.

You will use the *Expand-O-Matic* add-on to expand the archive. Its use is quite simple:

If you've received no instructions for the "Destination" of the files

1. Right-click the Desktop and choose **New Folder** (ALT+N). Don't worry about renaming the folder; it will only be a temporary folder.
2. Move the archive file to the new folder.
3. Open the folder and then double-click the archive to open it in Expand-O-Matic.
4. Click the **Expand** button. The files in the archive will be placed into the new folder. With some archives, subfolders will be created, which will also contain files.
5. Use the Tracker to move the expanded files to their destinations.

Note

To move a file: Right-click the file icon, then drag it to its destination. When you release the file, choose **Move.**

If you have been told to place the files in a specific "Destination"

1. Double-click the archive to open it in Expand-O-Matic.
2. Click the **Destination** button.
3. Use the File Dialog to find the destination folder. You must double-click the destination folder to select it.
4. Click the **Select** button to select the destination folder named on the button.
5. Click the **Expand** button. The files in the archive will be placed into the selected destination folder.

Commands Reference

6

The command reference provides two “views” of CueStation. The first section deals with commands for each application window: in other words, what commands are available within a given window.

The second section, starting on page , lists the commands by the type of action they perform: cue editing, file management, hardware setup and et cetera.

Most Windows

LCS Menu

About CueStation...
Displays the application version number, copyright, and credits

This information is needed when contacting LCS support.

Help > *list of help files*
Displays general CueStation help.

Language > *English, Japanese*
Sets the language for the CueStation menus, buttons and other interface elements.

ToolTips Enabled `Alt+H`
Shows or hides ToolTip help.

A checkmark indicates that ToolTip help is currently on, and will be turned off if the command is selected; and vice-versa.

Configure User Interface...
Displays user interface configuration options.

Configure Mixer...
Displays hardware configuration options.

This command is used when "configuring reality." It's also needed when contacting LCS support.

Send Configuration and Mixer Settings `Shift+Alt+C`
Sends hardware configuration to Matrix^3 system.

Used for "configuring reality" and for re-synchronizing the system.

Send Mixer Settings `Shift+Alt+S`
Send automation data settings to Matrix^3 system.

Used for "configuring reality" and for re-synchronizing the system.

Request Mixer Settings `Shift+Alt+R`
Retrieves the current mixer settings from the Matrix^3 system.

Does not retrieve the mixer configuration or automation data.

Send Project `Shift+Alt+D`
Sends automation data to Matrix^3 system.

Used for "configuring reality" and for re-synchronizing the system.

Pause Fades `Alt+P`
Pause all automated fades.

Resume Fades `Shift+Alt+P`
Resume automated fades.

Finish Fades `Alt+;`
Set automated fades to their final position immediately.

Cancel Fades `Alt+/`
Stop all fade automation.

Cancel Trajectories `Shift+Alt+'`
Stop all trajectory automation.

Clear Scheduler `Shift+Alt+/`
Stops all automation.

Silence! `Shift+Alt+,`
Sets the manual fade control to full-off position, and stops all automation.

Change Port > *list of ports*
Selects the communications port between the CueStation host computer and the Matrix^3 system.

Change Port Rate > *list of port rates*
Selects the communication speed for serial port connections between the CueStation host computer and the Matrix^3 system.

Offline `Alt+'`
Switches the CueStation host computer and the Matrix^3 system serial communications on and off.

Useful when programming an automation list when the computer isn't connected to the CueStation network or to the Matrix^3.

Blind Transmit
Disables the serial port handshaking protocol.

Used only when downloading new firmware to systems that boot from EPROM.

Ignore Time Code
Switches the Matrix^3 system between using and ignoring time code.

File Menu

New Project
Create a new project.

Open Project... `Alt+O`
Open an existing project.

Merge Project...
Merges files.

Save Project `Alt+S`
Save the current project using its previous file name.

If the project has not been named, you'll be asked for a name.

Save Project As...
Save the current project using a new file name.

Save as Default Project
Sets the current project as the default project.

When CueStation is started, it will load the default project automatically, saving you a few steps.

Generate Report...
Generates a report.

Back up Project `Alt+B`
Creates a backup of the current project. The file will be located in the LCS_Backup directory. It will reuse the current file name, appending the date and time to the file name.

We suggest you backup your project regularly, either every x minutes, or as you achieve mini-milestones.

Set Project Password...
Sets an access password for the project.

This is not intended to provide absolute security for your projects. It's a means to prevent casual users from changing your files, but will not stop a determined hacker.

Quit `Alt+Q`
Quits CueStation.

If the project has been changed, but has not been saved or re-saved, you will be prompted to save the file.

Application Windows

- Most Windows (common to all CueStation Windows)
- Log
- Subcue Librarian
- Transport
- Cue List
- Cue Library
- Subcue Library
- Externals
- SpaceMap
- Console
- Bus Levels
- Matrix
- Output Masters
- Aux Masters
- Virtual Groups
- Input EQs
- Output EQs
- Aux Output EQ

Windows Menu

System Status `Alt+1`
Shows the System Status window.

System Level `Alt+2`
Shows the System Level fader window.

Manual System Level `Alt+3`
Shows the Manual System Level fader window.

Transport `Alt+4`
Shows the Transport window.

SpaceMap `Alt+5`
Shows the SpaceMap window.

Console `Alt+6`
Shows the Console control window.

Input Processing `Alt+7`
Shows the Input Processing (EQ, Compressors) window.

Bus Levels `Alt+8`
Shows the Bus Levels faders window.

Matrix `Alt+9`
Shows the Matrix mapping values window.

Output Masters `Alt+0`
Shows the Output Masters faders window.

Output Processing `Ctrl+Alt+1`
Shows the Output Processing (EQ, Compressors) window.

Aux Masters `Ctrl+Alt+2`
Shows the Aux Masters faders window.

Aux Processing `Ctrl+Alt+3`
Shows the Aux Processing (EQ, Compressors) window.

Virtual Groups `Ctrl+Alt+4`
Shows the Virtual Groups window.

Cue List `Ctrl+Alt+5`
Shows the Cue List editor window.

Cue Library `Ctrl+Alt+6`
Shows the Cue Library editor window.

Subcue Library `Ctrl+Alt+7`
Shows the Subcue Library editor window.

Externals Library `Ctrl+Alt+8`
Shows the Externals Library editor window.

Subcue Librarian `Ctrl+Alt+9`
Shows the Subcue Librarian editor window.

Log `Ctrl+Alt+0`
Shows the Log window.

Project Notes
Shows the Project Notes memo window.

Subcue Librarian

File Menu

Preferences...
Sets the CueLibServer host computer IP address.

The CueLibServer host computer is directly connected to the Matrix[3].

Open CueLibServer Status Window...
Displays names and IP addresses of server and client applications.

Used for troubleshooting the network.

Check LCS Apps
Searches for multiple copies of CueStation and Helper applications.

Used for troubleshooting the network.

About
Displays information about Subcue Librarian.

Quit `Alt+Q`
Quits Subcue Librarian.

Edit Menu

Cut `Alt+X`
Moves the selected subcue to the clipboard.

Copy `Alt+C`
Copies the selected subcue to the clipboard.

Paste `Alt+V`
Copies the clipboard data to the selected subcue or, if no subcue is selected, creates a new subcue using that data.

Select All `Alt+A`
Selects everything.

Subcue Menu

New > *Wild Tracks Editor Subcue, External Subcue*
Creates a new Wild Tracks or External subcue.

Edit `Alt+Y`
Displays the editor for the selected subcue.

Do Now
Triggers the selected subcue. CueStation controls and the Matrix[3] are updated.

Delete Subcue `Backspace`
Removes the selected subcue.

Rename...
Renames the selected subcue.

Change Comment...
Changes the comment for the selected subcue.

Transport

Cue Lists Menu

[cue list series]
Selects a cue list for playback.

Control Menu

Resync `Shift+Alt+R`
Reestablishes time code synchronization.

Show/Hide Time Code `Alt+T`
Shows or hides the display of time code.

Show/Hide Time & Date `Shift+Alt+T`
Shows or hides the display of time and date.

LX-300 Menu

[various commands]
Displays status information and controls.

Used for troubleshooting.

Cue List

Cue Lists Menu

New Cue List `Alt+N`
Creates a new cue list.

Rename Cue List...
Renames the current cue list.

Duplicate Cue List `Alt+D`
Duplicates the current cue list.

Delete Cue List
Deletes the current cue list.

Optimize Cues and Mixer Subcues
Removes duplicate subcues from the Project file.

This command can not be undone.

Purge Unused Mixer Subcues
Removes unused subcues from the cue list.

This command can not be undone.

Purge Unused Cues and Mixer Subcues
Removes unused cues and subcues from the Project file.

This command can not be undone.

[list of cue lists]
Selects a cue list.

Cues Menu

New Cue Entry `Alt+E`
Creates a new cue entry.

New Cue & Cue Entry & Capture All `F4`
Captures all subcues, creates a new cue using those subcues, and creates a new entry for the cue.

Recall Cue Entry `Alt+R`
Triggers the selected cue. CueStation controls and the Matrix[3] are updated.

Duplicate Cue Entry
Duplicates the selected cue list entry.

Duplicate Cue & Cue Entry
Duplicates the selected cue and places it into the cue list.

Remove Cue Entry `Backspace`
Removes the selected cue list entry.

Rename Cue...
Renames the selected cue.

Edit Cue Comment... `Alt+M`
Changes the comment for the selected cue.

Allow Overlap `Alt+L`
Allow pending subcues to fire, regardless of wait time.

Edit Cue `Alt+Y`
Displays the Cue Library editor for the selected cue.

Capture All `F3`
Replaces the subcues of the selected cue by creating new subcues for the cue.

The new subcue will have a unique name. Only the current cue will use the new subcues; other cues will continue to use the old subcues.

Capture Differences `F2`
Replaces the subcues of the selected cue by creating new subcues for the cue. Only those subcues which differ from the existing controls will be replaced.

The new subcue will have a unique name. Only the current cue will use the new subcues; other cues will continue to use the old subcues.

Capture Differences & Follow Through
Replaces the subcues of the selected cue, and all cues following it, until a cue is found that doesn't contain the same subcue data as the first cue.

Update All Subcues `F1`
Replaces the subcues of the selected cue and all cues that use those subcues.

This change can not be easily undone. Use with caution.

Copy Cue `Ctrl+Alt+C`
Copies the selected cue to the clipboard.

Paste Cue `Ctrl+Alt+V`
Copies the clipboard data to the selected cue or, if no cue is selected, creates a new cue using that data.

Subcues Menu

Recall Subcue
Triggers the selected subcue. CueStation controls and the Matrix[3] are updated.

Instant Recall Subcue
Triggers the selected subcue, with all delays set to zero. CueStation controls and the Matrix[3] are updated.

The subcue delays are set to zero only for the recall; they retain their delays in the library.

Remove Subcue
Removes the selected subcue from the selected cue.

The subcue is not deleted: it will remain in the subcue library.

Capture New Subcue
Creates a new subcue.

Update Subcue (Recapture)
Replaces the selected subcue.

Rename Subcue...
Renames the selected subcue.

Edit Subcue Comment... `Alt+J`
Changes the comment for the selected subcue.

Copy Subcue `Alt+C`
Copies the selected subcue to the clipboard.

Paste Subcue `Alt+V`
Copies the clipboard data to the selected subcue or, if no subcue is selected, creates a new subcue using that data.

Cue Library

Cues Menu

New Cue `Alt+N`
Creates a new cue.

New Cue & Capture All `F3`
Creates new subcues and creates a new cue using those subcues.

Recall Cue `Alt+R`
Triggers the selected cue. CueStation controls and the Matrix[3] are updated.

Instant Recall Cue `Alt+T`
Triggers the selected cue, with all subcue delays set to zero. CueStation controls and the Matrix[3] are updated.

The subcue delays are set to zero only for the recall; they retain their delays in the library.

Rename Cue...
Renames the selected cue.

Edit Cue Comment... `Alt+M`
Changes the comment for the selected cue.

Duplicate Cue `Alt+D`
Duplicates the selected cue.

Allow Overlap `Alt+L`
Allow pending subcues to fire, regardless of wait time.

Capture All `F3`
Replaces the subcues of the selected cue by creating new subcues for the cue.

The new subcue will have a unique name. Only the current cue will use the new subcues; other cues will continue to use the old subcues.

Capture Differences `F2`
Replaces the subcues of the selected cue by creating new subcues for the cue. Only those subcues which differ from the existing controls will be replaced.

The new subcue will have a unique name. Only the current cue will use the new subcues; other cues will continue to use the old subcues.

Update All Subcues `F1`
Replaces the subcues of the selected cue and all cues that use those subcues.

This change can not be easily undone. Use with caution.

Copy Cue `Ctrl+Alt+C`
Copies the selected cue to the clipboard.

Paste Cue `Ctrl+Alt+V`
Copies the clipboard data to the selected cue or, if no cue is selected, creates a new cue using that data.

Subcues Menu

Recall Subcue
Triggers the selected subcue. CueStation controls and the Matrix[3] are updated.

Instant Recall Subcue
Triggers the selected subcue, with all delays set to zero. CueStation controls and the Matrix[3] are updated.

The subcue delays are set to zero only for the recall; they retain their delays in the library.

Remove Subcue
Removes the selected subcue from the selected cue.

The subcue is not deleted: it will remain in the subcue library.

Capture New Subcue
Creates a new subcue.

Update Subcue (Recapture)
Replaces the selected subcue. All cues that use the subcue will be affected.

Rename Subcue...
Renames the selected subcue.

Edit Subcue Comment... `Alt+J`
Changes the comment for the selected subcue.

Copy Subcue `Alt+C`
Copies the selected subcue to the clipboard.

Paste Subcue `Alt+V`
Copies the clipboard data to the selected subcue or, if no subcue is selected, creates a new subcue using that data.

Subcue Library

Subcues Menu

New Subcue `Alt+N`
Create a new subcue.

Recall Subcue `Alt+R`
Triggers the selected subcue. CueStation controls and the Matrix[3] are updated.

Instant Recall Subcue `Alt+T`
Triggers the selected subcue, with the subcue delays set to zero. CueStation controls and the Matrix[3] are updated.

The subcue delays are set to zero only for the recall; they retain their delays in the library.

Rename Subcue...
Renames the selected subcue.

Edit Subcue Comment...
Changes the comment for the selected subcue.

Duplicate Subcue `Alt+D`
Duplicates the selected subcue.

Edit Subcue `Alt+Y`
Sets the parameters of the selected trajectory subcue.

Update Subcue
Replaces the selected subcue. All cues that use the subcue will be affected.

Copy Subcue `Alt+C`
Copies the selected subcue to the clipboard.

Paste Subcue `Alt+V`
Copies the clipboard data to the selected subcue or, if no subcue is selected, creates a new subcue using that data.

Bus Menu

[bus number]
Set bus assignment for Matrix subcues.

Externals Library

Subcues Menu

New External Subcue `Alt+N`
Create a new subcue.

Recall Subcue `Alt+R`
Triggers the selected subcue. CueStation controls and the Matrix[3] are updated.

Instant Recall Subcue `Alt+T`
Triggers the selected subcue, with the subcue delays set to zero.

CueStation controls and the Matrix[3] are updated.

The subcue delays are set to zero only for the recall; they retain their delays in the library.

Rename Subcue...
Renames the selected subcue.

Edit Subcue Comment...
Changes the comment for the selected subcue.

Duplicate Subcue `Alt+D`
Duplicates the selected subcue.

Copy Subcue `Alt+C`
Copies the selected subcue to the clipboard.

Externals Menu

New Cmd Entry `Alt+E`
Create a new command entry for the current external subcue.

Delete Cmd Entry
Delete a command entry for the current external subcue.

Edit Cmd Data `Alt+Y`
Edit the command entry for the current external subcue.

SpaceMap

Space Menu

New Triset `Alt+T`
Create a new triset.

Select All `Alt+A`
Select all nodes on the SpaceMap.

Link Virtual/Derived `Alt+L`
Link the selected speaker nodes with the selected virtual or derived node.

Unlink Virtual/Derived `Alt+U`
Unlink the selected virtual or derived node.

Edit Virtual Weights... `Alt+Y`
Edit the scaling factors for the selected virtual node.

Configure Joystick Actions...
Configure various functions for the joystick interface to SpaceMap.

Maps Menu

New SpaceMap `Alt+N`
Create a new SpaceMap.

Duplicate SpaceMap
Duplicate the current SpaceMap.

Rename SpaceMap
Rename the current SpaceMap.

Delete SpaceMap
Deletes the current SpaceMap.

Lock SpaceMaps `Shift+Alt+L`
Prevents changes to the SpaceMap.

Useful when creating trajectories: you won't accidently drag a SpaceMap node when you meant to move a trajectory node.

[map list]
Selects a SpaceMap.

Trajs Menu

New Trajectory `Alt+J`
Create new trajectory.

Duplicate Trajectory `Alt+D`
Duplicate the current trajectory.

Delete Trajectory
Delete the current trajectory.

Rename Trajectory
Rename the current Trajectory.

Insert Trajectory Point `Alt+I`
Insert a node in the middle of the selected trajectory segment.

The selected trajectory segment follows the currently-selected trajectory node. The time value will be evenly split between the two segments.

Capture Trajectory Subcue... `Shift+Alt+T`
Create a new trajectory subcue.

Play Trajectory `Alt+R`
Play the trajectory.

[trajectory list]
Selects a Trajectory.

Bus Menu

[bus list]
Selects a Bus.

Plot Menu

Set Background Picture...
Sets the background picture for the SpaceMap.

Don't make the mistake of believing that node layout must reflect physical speaker positioning.

Draw Trajectory
Shows or hides the display of trajectories.

Draw Trajectory Node Times
Shows or hides the display of time values for each point in the trajectory.

Draw Bus Positions
Shows or hides the display of panning for each assigned bus.

Draw Background Picture
Shows or hides the display of background picture.

Draw Grid
Shows or hides the display of grid.

Draw Virtual Links
Shows or hides the display of virtual links.

Draw Derived Links
Shows or hides the display of derived links.

Draw Trisets
Shows or hides the display of trisets.

Draw Triset IDs
Shows or hides the display of triset IDs.

Draw Nodes
Shows or hides the display of nodes.

Draw Node IDs
Shows or hides the display of node IDs.

Draw Joystick Range
Shows or hides the display of the joystick's range of motion.

Snap to Grid
Shows or hides the snapping of nodes to the grid.

Grid Menu

[values list]
Sets the size of the snap-to grid.

Type Menu

insert type Speaker
Changes the selected node to a speaker node.

insert type Virtual
Changes the selected node to a virtual node.

insert type Silent
Changes the selected node to a silent node.

insert type Derived
Changes the selected node to a derived node.

Console

Hide Menu

Hide/Show Aux Sends
Shows or hides the display of auxiliary sends.

Hide/Show Bus Assigns
Shows or hides the display of bus assignment buttons.

InputPages Menu

New Page...
Creates a new Input page.

Delete Page
Deletes the current Input page.

Rename Page...
Renames the current Input page.

Series From > *(list)*
Selects the range of inputs to be placed on the current Input page.

[page list]
Selects an existing input page.

AuxPages Menu

New Page...
Creates a new Aux page.

Delete Page
Deletes the current Aux page.

Rename Page...
Renames the current Aux page.

Series From > *list of input channels*
Selects the range of sends to be placed on the current Aux page.

[page list]
Selects an existing Aux page.

Bus Levels

Pages Menu

New Page...
Creates a new Input page.

Delete Page
Deletes the current Input page.

Rename Page...
Renames the current Input page.

Series From > *list of input buses*
Selects the range of inputs to be placed on the current Input page.

[page list]
Selects an existing input page.

Matrix

Matrix Menu

Clear Matrix `Shift+Alt+Zero`
Sets all Matrix entries to off.

Effectively silences the signal flow. Used as a starting point for designing a Matrix.

Set Diagonal `Alt+\`
Sets entries where bus=output to unity.

Used for one-to-one mapping of inputs to outputs; effectively nullifies the Matrix.

Set Diagonal & Buses & Outputs `Shift+Alt+\`
Sets entries where bus=output to unity, and sets all bus and output levels to unity.

Saves time in going back to the faders to bring them up to unity.

BusPages Menu

New Page...
Creates a new bus page.

Delete Page
Deletes the current bus page.

Rename Page...
Renames the current bus page.

Series From > *list of input buses*
Selects the range of bus channels to be placed on the current bus page.

[page list]
Selects an existing bus page.

Outpages Menu

New Page...
Creates a new output page.

Delete Page
Deletes the current output page.

Rename Page...
Renames the current output page.

Series From > *list of output channels*
Selects the range of output channels to be placed on the current output page.

[page list]
Selects an existing output page.

Output Masters

Pages Menu

New Page...
Creates a new output page.

Delete Page
Deletes the current output page.

Rename Page...
Renames the current output page.

Series From > *list of output channels*
Selects the range of output channels to be placed on the current output page.

[page list]
Selects an existing output page.

Aux Masters

Pages Menu

New Page...
Creates a new Aux page.

Delete Page
Deletes the current Aux page.

Rename Page...
Renames the current Aux page.

Series From
Selects the range of sends to be placed on the current Aux page.

[page list]
Selects an existing Aux page.

Virtual Groups

Pages Menu

New Page...
Creates a new Virtual Group page.

Delete Page
Deletes the current Virtual Group page.

Rename Page...
Renames the current Virtual Group page.

Series From > *list of buses*
Selects the range of sends to be placed on the current Virtual Group page.

[page list]
Selects an existing Virtual Group page.

Input EQ

Edit Menu

Copy EQs `Alt+C`
Copy the EQ data to the clipboard.

Paste EQs `Alt+V`
Copies the clipboard data to the EQ.

Zero EQ Gains `Alt+Zero`
Set the EQ gain to zero.

Plot Menu

Show Phase
Show or hide the display of the phase response graph.

Output EQ

Edit Menu

Copy EQs `Alt+C`
Copy the EQ data to the clipboard.

Paste EQs `Alt+V`
Copies the clipboard data to the EQ.

Zero EQ Gains `Alt+Zero`
Set the EQ gain to zero.

Plot Menu

Show Phase
Show or hide the display of the phase response graph.

Aux Output EQ

Edit Menu

Copy EQs `Alt+C`
Copy the EQ data to the clipboard.

Paste EQs `Alt+V`
Copies the clipboard data to the EQ.

Zero EQ Gains `Alt+Zero`
Set the EQ gain to zero.

Plot Menu

Show Phase
Shows or hides the display of the phase response graph.

Log

File Menu

Save Log... `Alt+S`
Saves the contents of the log to a file.

Choose Log Colors...
Sets the colors of the Log window.

Application Control

File > Check LCS Apps
Searches for multiple copies of CueStation and Helper applications.
Used for troubleshooting the network.
Subcue Librarian

File > Open CueLibServer Status Window...
Displays names and IP addresses of server and client applications.
Used for troubleshooting the network.
Subcue Librarian

File > Quit `Alt+Q`
Quits CueStation.
If the project has been changed, but has not been saved or re-saved, you will be prompted to save the file.
Most Windows

File > Quit `Alt+Q`
Quits Subcue Librarian.
Subcue Librarian

Windows > Aux Masters `Shift+Alt+5`
Shows the Auxiliary Masters window.
Most Windows

Windows > Aux Output EQ `Shift+Alt+9`
Shows the Auxiliary Output EQ windows.
Most Windows

Windows > Bus Levels `Alt+0`
Shows the Bus Levels window.
Most Windows

Windows > Console `Alt+9`
Shows the Console window.
Most Windows

Windows > Cue Library `Alt+5`
Shows the Cue Library window.
Most Windows

Windows > Cue List `Alt+4`
Shows the Cue List window.
Most Windows

Windows > Externals Library `Alt+7`
Shows the Externals Library window.
Most Windows

Windows > Input EQ `Shift+Alt+7`
Shows the Input EQ window.
Most Windows

Windows > Log `Alt+1`
Shows the Log window.
Most Windows

Windows > Manual System Level `Shift+Alt+4`
Shows the Manual System Level window.
Most Windows

Windows > Matrix `Shift+Alt+1`
Shows the Matrix window.
Most Windows

Windows > Output EQ `Shift+Alt+8`
Shows the Output EQ window.
Most Windows

Windows > Output Masters `Shift+Alt+2`
Shows the Output Masters window.
Most Windows

Windows > SpaceMap `Alt+8`
Shows the SpaceMap window.
Most Windows

Windows > Subcue Librarian `Alt+2`
Shows the Subcue Librarian window.
Most Windows

Windows > Subcue Library `Alt+6`
Shows the Subcue Library window.
Most Windows

Windows > System Level `Shift+Alt+3`
Shows the System Level window.
Most Windows

Windows > System Status `Shift+Alt+0`
Shows the System Status window.
Most Windows

Windows > Transport `Alt+3`
Shows the Transport window.
Most Windows

Windows > Virtual Groups `Shift+Alt+6`
Shows the Virtual Groups window.
Most Windows

Automation Control

LCS > Cancel Fades `Alt+/`
Stop all fade automation.
Most Windows

LCS > Cancel Trajectories `Shift+Alt+'`
Stop all trajectory automation.
Most Windows

LCS > Clear Scheduler `Shift+Alt+/`
Stops all automation.
Most Windows

LCS > Finish Fades `Alt+;`
Set automated fades to their final position immediately.
Most Windows

LCS > Pause Fades `Alt+P`
Pause all automated fades.
Most Windows

LCS > Resume Fades `Shift+Alt+P`
Resume automated fades.
Most Windows

LCS > Silence! `Shift+Alt+,`
Sets the manual fade control to full-off position, and stops all automation.
Most Windows

Categories:

- Application Control
- Automation Control
- Cue Control
- Cue Edit
- Cue List Edit
- EQ Edit
- Externals Edit
- Hardware Control
- Hardware Setup
- Information
- Log
- Matrix Edit
- Project Control
- Project Edit
- Software Control
- SpaceMap Control
- SpaceMap Edit
- Subcue Control
- Subcue Edit
- UI Preferences

Cue Control

Cues > Allow Overlap `Alt+L`
Allow pending subcues to fire, regardless of wait time.
Cue List, Cue Library

Cues > Instant Recall Cue `Alt+T`
Triggers the selected cue, with all subcue delays set to zero. CueStation controls and the Matrix[3] are updated.
The subcue delays are set to zero only for the recall; they retain their delays in the library.
Cue Library

Cues > Recall Cue `Alt+R`
Triggers the selected cue. CueStation controls and the Matrix[3] are updated.
Cue Library

Cues > Recall Cue Entry `Alt+R`
Triggers the selected cue. CueStation controls and the Matrix[3] are updated.
Cue List

Cue Edit

Cue Lists > Delete Cue List
Deletes the current cue list.
Cue List

Cue Lists > Duplicate Cue List `Alt+D`
Duplicates the current cue list.
Cue List

Cue Lists > New Cue List `Alt+N`
Creates a new cue list.
Cue List

Cue Lists > Rename Cue List...
Renames the current cue list.
Cue List

Cues > Copy Cue `Ctrl+Alt+C`
Copies the selected cue to the clipboard.
Cue List, Cue Library

Cues > Duplicate Cue `Alt+D`
Duplicates the selected cue.
Cue Library

Cues > Duplicate Cue & Cue Entry
Duplicates the selected cue and places it into the cue list.
Cue List

Cues > Edit Cue `Alt+Y`
Displays the Cue Library editor for the selected cue.
Cue List

Cues > Edit Cue Comment... `Alt+M`
Changes the comment for the selected cue.
Cue List, Cue Library

Cues > New Cue `Alt+N`
Creates a new cue.
Cue Library

Cues > New Cue & Capture All `F3`
Creates new subcues and creates a new cue using those subcues.
Cue Library

Cues > Paste Cue `Ctrl+Alt+V`
Copies the clipboard data to the selected cue or, if no cue is selected, creates a new cue using that data.
Cue List, Cue Library

Cues > Rename Cue...
Renames the selected cue.
Cue List, Cue Library

Subcues > Remove Subcue
Removes the selected subcue from the selected cue.
The subcue is not deleted: it will remain in the subcue library.
Cue Library

Cue List Edit

Cues > Duplicate Cue Entry
Duplicates the selected cue list entry.
Cue List

Cues > New Cue Entry `Alt+E`
Creates a new cue entry.
Cue List

Cues > Remove Cue Entry `Backspace`
Removes the selected cue list entry.
Cue List

EQ Edit

Edit > Copy EQs `Alt+C`
Copy the EQ data to the clipboard.
Input EQ, Output EQ, Aux Output EQ

Edit > Paste EQs `Alt+V`
Copies the clipboard data to the EQ.
Input EQ, Output EQ, Aux Output EQ

Edit > Zero EQ Gains `Alt+Zero`
Set the EQ gain to zero.
Input EQ, Output EQ, Aux Output EQ

Externals Edit

Externals > Delete Cmd Entry
Delete a command entry for the current external subcue.
Externals Library

Externals > Edit Cmd Data `Alt+Y`
Edit the command entry for the current external subcue.
Externals Library

Externals > New Cmd Entry `Alt+E`
Create a new command entry for the current external subcue.
Externals Library

Hardware Control

Control > Resync `Shift+Alt+R`
Reestablishes time code synchronization.
Transport

LCS > Blind Transmit
Disables the serial port handshaking protocol.
Used only when downloading new firmware to systems that boot from EPROM.
Most Windows

LCS > Ignore Time Code
Switches the Matrix^3 system between using and ignoring time code.
Most Windows

LCS > Offline `Alt+'`
Switches the CueStation host computer and the Matrix^3 system serial communications on and off.
Useful when programming an automation list when the computer isn't connected to the CueStation network or to the Matrix^3.
Most Windows

LCS > Send Configuration and Mixer Settings `Shift+Alt+C`
Sends hardware configuration to Matrix^3 system.
Used for "configuring reality" and for re-synchronizing the system.
Most Windows

LCS > Send Mixer Settings `Shift+Alt+S`
Send automation data settings to Matrix^3 system.
Used for "configuring reality" and for re-synchronizing the system.
Most Windows

LCS > Send Project `Shift+Alt+D`
Sends automation data to Matrix^3 system.
Used for "configuring reality" and for re-synchronizing the system.
Most Windows

Hardware Setup

LCS > Change Port > *list of ports*
Selects the communications port between the CueStation host computer and the Matrix^3 system.
Most Windows

LCS > Change Port Rate > *list of port rates*
Selects the communication speed for serial port connections between the CueStation host computer and the Matrix^3 system.
Most Windows

LCS > Configure Mixer...
Displays hardware configuration options.
This command is used when "configuring reality." It's also needed when contacting LCS support.
Most Windows

Information

File > About
Displays information about Subcue Librarian.
Subcue Librarian

LCS > About CueStation...
Displays the application version number, copyright, and credits
This information is needed when contacting LCS support.
Most Windows

LCS > Help > *list of help files*
Displays general CueStation help.
Most Windows

LCS > Request Mixer Settings `Shift+Alt+R`
Retrieves the current mixer settings from the Matrix^3 system.
Does not retrieve the mixer configuration or automation data.
Most Windows

LX-300 > [various commands]
Displays status information and controls.
Used for troubleshooting.
Transport

Log

File > Save Log... `Alt+S`
Saves the contents of the log to a file.
Log

Matrix Edit

Matrix > Clear Matrix `Shift+Alt+Zero`
Sets all Matrix entries to off.
Effectively silences the signal flow. Used as a starting point for designing a Matrix.
Matrix

Matrix > Set Diagonal `Alt+\`
Sets entries where bus=output to unity.
Used for one-to-one mapping of inputs to outputs; effectively nullifies the Matrix.
Matrix

Matrix > Set Diagonal & Buses & Outputs `Shift+Alt+\`
Sets entries where bus=output to unity, and sets all bus and output levels to unity.
Saves time in going back to the faders to bring them up to unity.
Matrix

Project Control

Cue Lists > [cue list series]
Selects a cue list for playback.
Transport

File > Back up Project `Alt+B`
Creates a backup of the current project. The file will be located in the LCS_Backup directory. It will reuse the current file name, appending the date and time to the file name.

We suggest you backup your project regularly, either every x minutes, or as you achieve mini-milestones.
Most Windows

File > Set Project Password...
Sets an access password for the project.

This is not intended to provide absolute security for your projects. It's a means to prevent casual users from changing your files, but will not stop a determined hacker.
Most Windows

Project Edit

Cue Lists > Optimize Cues and Mixer Subcues
Removes duplicate subcues from the Project file.

This command can not be undone.
Cue List

Cue Lists > Purge Unused Cues and Mixer Subcues
Removes unused cues and subcues from the Project file.

This command can not be undone.
Cue List

Cue Lists > [list of cue lists]
Selects a cue list.
Cue List

File > Generate Report...
Generates a report.
Most Windows

File > Merge Project...
Merges files.
Most Windows

File > New Project
Create a new project.
Most Windows

File > Open Project... `Alt+O`
Open an existing project.
Most Windows

File > Save Project `Alt+S`
Save the current project using its previous file name.

If the project has not been named, you'll be asked for a name.
Most Windows

File > Save Project As...
ave the current project using a new file name.
Most Windows

File > Save as Default Project
Sets the current project as the default project.

When CueStation is started, it will load the default project automatically, saving you a few steps.
Most Windows

Software Control

File > Preferences...
Sets the CueLibServer host computer IP address.

The CueLibServer host computer is directly connected to the Matrix[3].
Subcue Librarian

SpaceMap Edit

Bus > [bus list]
Selects a Bus.
SpaceMap

Maps > Delete SpaceMap

SpaceMap

Maps > Duplicate SpaceMap
Duplicate the current SpaceMap.
SpaceMap

Maps > Lock SpaceMaps `Shift+Alt+L`
Prevents changes to the SpaceMap.

Useful when creating trajectories: you won't accidently drag a SpaceMap node when you meant to move a trajectory node.
SpaceMap

Maps > New SpaceMap `Alt+N`
Create a new SpaceMap.
SpaceMap

Maps > Rename SpaceMap
Rename the current SpaceMap.
SpaceMap

Maps > [map list]
Selects a SpaceMap.
SpaceMap

Space > Configure Joystick Actions...
Configure various functions for the joystick interface to SpaceMap.
SpaceMap

Space > Edit Virtual Weights... `Alt+Y`
Edit the scaling factors for the selected virtual node.
SpaceMap

Space > Link Virtual/Derived `Alt+L`
Link the selected speaker nodes with the selected virtual or derived node.
SpaceMap

Space > New Triset `Alt+T`
Create a new triset.
SpaceMap

Space > Select All `Alt+A`
Select all nodes on the SpaceMap.
SpaceMap

Space > Unlink Virtual/Derived `Alt+U`
Unlink the selected virtual or derived node.
SpaceMap

Trajs > Capture Trajectory Subcue... `Shift+Alt+T`
Create a new trajectory subcue.
SpaceMap

Trajs > Delete Trajectory
Delete the current trajectory.
SpaceMap

Trajs > Duplicate Trajectory `Alt+D`
Duplicate the current trajectory.
SpaceMap

Trajs > Insert Trajectory Point `Alt+I`
Insert a node in the middle of the selected trajectory segment.

The selected trajectory segment follows the currently-selected trajectory node. The time value will be evenly split between the two segments.
SpaceMap

Trajs > New Trajectory `Alt+J`
Create new trajectory.
SpaceMap

Trajs > Play Trajectory `Alt+R`
Play the trajectory.
SpaceMap

Trajs > Rename Trajectory
Rename the current Trajectory.
SpaceMap

Trajs > [trajectory list]
Selects a Trajectory.
SpaceMap

Type > insert type Derived
Changes the selected node to a derived node.
SpaceMap

Type > insert type Silent
Changes the selected node to a silent node.
SpaceMap

Type > insert type Speaker
Changes the selected node to a speaker node.
SpaceMap

Type > insert type Virtual
Changes the selected node to a virtual node.
SpaceMap

SubCue Control

Subcue > Do Now
Triggers the selected subcue. CueStation controls and the Matrix[3] are updated.
Subcue Librarian

Subcues > Instant Recall Subcue
Triggers the selected subcue, with all delays set to zero. CueStation controls and the Matrix[3] are updated.

The subcue delays are set to zero only for the recall; they retain their delays in the library.
Cue Library

Subcues > Instant Recall Subcue `Alt+T`
Triggers the selected subcue, with the subcue delays set to zero. CueStation controls and the Matrix[3] are updated.

The subcue delays are set to zero only for the recall; they retain their delays in the library.
Subcue Library

Subcues > Recall Subcue
Triggers the selected subcue. CueStation controls and the are updated.
Cue Library

Subcues > Recall Subcue `Alt+R`
Triggers the selected subcue. CueStation controls and the Matrix[3] are updated.
Subcue Library

Subcue Edit

Bus > [bus number]
Set bus assignment for Matrix subcues.
Subcue Library

Cue Lists > Purge Unused Mixer Subcues
Removes unused subcues from the cue list.

This command can not be undone.
Cue List

Cues > Capture All `F3`
Replaces the subcues of the selected cue by creating new subcues for the cue.

The new subcue will have a unique name. Only the current cue will use the new subcues; other cues will continue to use the old subcues.
Cue List, Cue Library

Cues > Capture Differences `F2`
Replaces the subcues of the selected cue by creating new subcues for the cue. Only those subcues which differ from the existing controls will be replaced.

The new subcue will have a unique name. Only the current cue will use the new subcues; other cues will continue to use the old subcues.
Cue List, Cue Library

Cues > Capture Differences & Follow Through
Replaces the subcues of the selected cue, and all cues following it, until a cue is found that doesn't contain the same subcue data as the first cue.
Cue List

Cues > New Cue & Cue Entry & Capture All `F4`
Captures all subcues, creates a new cue using those subcues, and creates a new entry for the cue.
Cue List

Cues > Update All Subcues `F1`
Replaces the subcues of the selected cue and all cues that use those subcues.

This change can not be easily undone. Use with caution.
Cue List, Cue Library

Edit > Copy `Alt+C`
Copies the selected subcue to the clipboard.
Subcue Librarian

Edit > Cut `Alt+X`
Moves the selected subcue to the clipboard.
Subcue Librarian

Edit > Paste `Alt+V`
Copies the clipboard data to the selected subcue or, if no subcue is selected, creates a new subcue using that data.
Subcue Librarian

Edit > Select All `Alt+A`
Subcue Librarian

Subcue > Change Comment...
Changes the comment for the selected subcue.
Subcue Librarian

Subcue > Delete Subcue `Backspace`
Removes the selected subcue.
Subcue Librarian

Subcue > Edit `Alt+Y`
Displays the editor for the selected subcue.
Subcue Librarian

Subcue > New > *Wild Tracks Editor Subcue, External Subcue*
Creates a new Wild Tracks or External subcue.
Subcue Librarian

Subcue > Rename...
Renames the selected subcue.
Subcue Librarian

Subcues > Capture New Subcue
Creates a new subcue.
Cue List, Cue Library

Subcues > Copy Subcue `Alt+C`
Copies the selected subcue to the clipboard.
Cue List, Cue Library, Subcue Library, Externals Library

Subcues > Duplicate Subcue `Alt+D`
Duplicates the selected subcue.
Subcue Library, Externals Library

Subcues > Edit Subcue `Alt+Y`
Sets the parameters of the selected trajectory subcue.
Subcue Library

Subcues > Edit Subcue Comment... `Alt+J`
Changes the comment for the selected subcue.
Cue List, Cue Library

Subcues > Edit Subcue Comment...
Changes the comment for the selected subcue.
Subcue Library, Externals Library

Subcues > Instant Recall Subcue
Triggers the selected subcue, with all delays set to zero. CueStation controls and the Matrix[3] are updated.
The subcue delays are set to zero only for the recall; they retain their delays in the library.
Cue List

Subcues > Instant Recall Subcue `Alt+T`
Triggers the selected subcue, with the subcue delays set to zero. CueStation controls and the Matrix[3] are updated.
The subcue delays are set to zero only for the recall; they retain their delays in the library.
Externals Library

Subcues > New External Subcue `Alt+N`
Create a new subcue.
Externals Library

Subcues > New Subcue `Alt+N`
Create a new subcue.
Subcue Library

Subcues > Paste Subcue `Alt+V`
Copies the clipboard data to the selected subcue or, if no subcue is selected, creates a new subcue using that data.
Cue List, Cue Library, Subcue Library

Subcues > Recall Subcue
Triggers the selected subcue. CueStation controls and the Matrix[3] are updated.
Cue List

Subcues > Recall Subcue `Alt+R`
Triggers the selected subcue. CueStation controls and the Matrix[3] are updated.
Externals Library

Subcues > Remove Subcue
Removes the selected subcue from the selected cue.
The subcue is not deleted: it will remain in the subcue library.
Cue List

Subcues > Rename Subcue...
Renames the selected subcue.
Cue List, Cue Library, Subcue Library, Externals Library

Subcues > Update Subcue
Replaces the selected subcue. All cues that use the subcue will be affected.
Subcue Library

Subcues > Update Subcue (Recapture)
Replaces the selected subcue.
Cue List

Subcues > Update Subcue (Recapture)
Replaces the selected subcue. All cues that use the subcue will be affected.
Cue Library

User Interface

AuxPages > Delete Page
Deletes the current Aux page.
Console

AuxPages > New Page...
Creates a new Aux page.
Console

AuxPages > Rename Page...
Renames the current Aux page.
Console

AuxPages > Series From > *list of input channels*
Selects the range of sends to be placed on the current Aux page.
Console

AuxPages > [page list]
Selects an existing Aux page.
Console

BusPages > Delete Page
Deletes the current bus page.
Matrix

BusPages > New Page...
Creates a new bus page.
Matrix

BusPages > Rename Page...
Renames the current bus page.
Matrix

BusPages > Series From > *list of input buses*
Selects the range of bus channels to be placed on the current bus page.
Matrix

BusPages > [page list]
Selects an existing bus page.
Matrix

Control > Show/Hide Time & Date `Shift+Alt+T`
Shows or hides the display of time and date.
Transport

Control > Show/Hide Time Code `Alt+T`
Shows or hides the display of time code.
Transport

File > Choose Log Colors...
Sets the colors of the Log window.
Log

Grid > [values list]
Sets the size of the snap-to grid.
SpaceMap

Hide > Hide/Show Aux Sends
Shows or hides the display of auxiliary sends.
Console

Hide > Hide/Show Bus Assigns
Shows or hides the display of bus assignment buttons.
Console

InputPages > Delete Page
Deletes the current Input page.
Console

InputPages > New Page...
Creates a new Input page.
Console

InputPages > Rename Page...
Renames the current Input page.
Console

InputPages > Series From > *(list)*
Selects the range of inputs to be placed on the current Input page.
Console

InputPages > [page list]
Selects an existing input page.
Console

LCS > Configure User Interface...
Displays user interface configuration options.
Most Windows

LCS > Language > *English, Japanese*
Sets the language for the CueStation menus, buttons and other interface elements.
Most Windows

LCS > ToolTips Enabled `Alt+H`
Shows or hides ToolTip help.
A checkmark indicates that ToolTip help is currently on, and will be turned off if the command is selected; and vice-versa.
Most Windows

Outpages > Delete Page
Deletes the current output page.
Matrix

Outpages > New Page...
Creates a new output page.
Matrix

Outpages > Rename Page...
Renames the current output page.
Matrix

Outpages > Series From > *list of output channels*
Selects the range of output channels to be placed on the current output page.
Matrix

Outpages > [page list]
Selects an existing output page.
Matrix

Pages > Delete Page
Deletes the current Input page.
Bus Levels

Pages > Delete Page
Deletes the current output page.
Output Masters

Pages > Delete Page
Deletes the current Aux page.
Aux Masters

Pages > Delete Page
Deletes the current Virtual Group page.
Virtual Groups

Pages > New Page...
Creates a new Input page.
Bus Levels

Pages > New Page...
Creates a new output page.
Output Masters

Pages > New Page...
Creates a new Aux page.
Aux Masters

Pages > New Page...
Creates a new Virtual Group page.
Virtual Groups

Pages > Rename Page...
Renames the current Input page.
Bus Levels

Pages > Rename Page...
Renames the current output page.
Output Masters

Pages > Rename Page...
Renames the current Aux page.
Aux Masters

Pages > Rename Page...
Renames the current Virtual Group page.
Virtual Groups

Pages > Series From
Selects the range of sends to be placed on the current Aux page.
Aux Masters

Pages > Series From > *list of buses*
Selects the range of sends to be placed on the current Virtual Group page.
Virtual Groups

Pages > Series From > *list of input buses*
Selects the range of inputs to be placed on the current Input page.
Bus Levels

Pages > Series From > *list of output channels*
Selects the range of output channels to be placed on the current output page.
Output Masters

Pages > [page list]
Selects an existing input page.
Bus Levels

Pages > [page list]
Selects an existing output page.
Output Masters

Pages > [page list]
Selects an existing Aux page.
Aux Masters

Pages > [page list]
Selects an existing Virtual Group page.
Virtual Groups

Plot > Draw Background Picture
Shows or hides the display of background picture.
SpaceMap

Plot > Draw Bus Positions
Shows or hides the display of panning for each assigned bus.
SpaceMap

Plot > Draw Derived Links
Shows or hides the display of derived links.
SpaceMap

Plot > Draw Grid
Shows or hides the display of grid.
SpaceMap

Plot > Draw Joystick Range
Shows or hides the display of the joystick's range of motion.
SpaceMap

Plot > Draw Node IDs
Shows or hides the display of node IDs.
SpaceMap

Plot > Draw Nodes
Shows or hides the display of nodes.
SpaceMap

Plot > Draw Trajectory
Shows or hides the display of trajectories.
SpaceMap

Plot > Draw Trajectory Node Times
Shows or hides the display of time values for each point in the trajectory.
SpaceMap

Plot > Draw Triset IDs
Shows or hides the display of triset IDs.
SpaceMap

Plot > Draw Trisets
Shows or hides the display of trisets.
SpaceMap

Plot > Draw Virtual Links
Shows or hides the display of virtual links.
SpaceMap

Plot > Set Background Picture...
Sets the background picture for the SpaceMap.
Don't make the mistake of believing that node layout must reflect physical speaker positioning.
SpaceMap

Plot > Show Phase
Show or hide the display of the phase response graph.
Input EQ, Output EQ

Plot > Show Phase
Shows or hides the display of the phase response graph.
Aux Output EQ

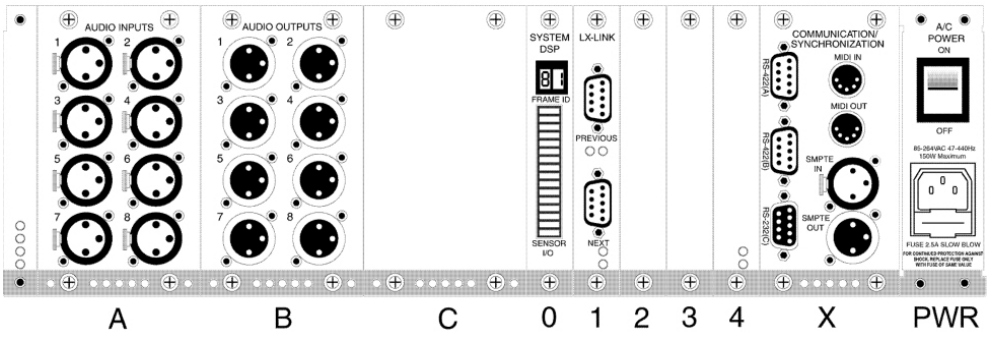
Plot > Snap to Grid
Shows or hides the snapping of nodes to the grid.
SpaceMap

Hardware Reference

7

This chapter presents detailed electrical, mechanical and physical specifications for Matrix^3 frames and cards.

This illustration shows a typical Matrix^3 back panel. The valid slot assignments are provided below.



LX-300 Chassis Slot Assignments		
Slot	**Cards Allowed in this Slot**	* indicates exclusivity; see page 7-3
A	LX-AI8 analog audio input *LX-AO8 analog audio output	*LX-ADAT digital audio input/output *LX-AES digital audio input/output
B	LX-AI8 analog audio input LX-AO8 analog audio output	LX-ADAT digital audio input/output LX-AES digital audio input/output
C	LX-CBR CobraNet audio interface	*LX-AO8 analog audio output
0	LX-DSP system processor	
1, 2, 3, 4	LX-LNK frame link interconnect LX-ELC EtherTracks/Wild Tracks	LX-EXP expansion DSP (max. 3) LX-VRA VRAS expansion
X	LX-COS communications/synchronization	

Frame Specifications

Operating Environment

0 to 40°C (32 to 100°F).

5 to 95% humidity, non-condensing.

Low dust levels.

Storage Environment

0 to 80°C (32 to 175°F).

5 to 95% humidity, non-condensing.

Power Consumption

AC, 47–70Hz, 100–250V, single phase.

150W maximum power required.

Fuse

2.5A, 240V slow-blow, 5x20mm.

Dimensions

435×127×348mm W×H×D (17.126×5×13.7 inches).

Weight

8.2kg (18lb) base configuration, with LX-DSP system card and AC power cord.

Approximately 11kg (25lb) maximum weight with cards.

Shock/Vibration

20G omni-directional.

Heat Dissipation

150W maximum.

Racks/Enclosures

Standard 19” rack mount. All four mounting points on the front panel must be secured.

Minimum 10cm (2.5 inch) unobstructed air space behind and in front of each frame.

No air space requirements above and below each frame.

Racks shall be secured to floor and ceiling, to prevent tip-over, grounded against electrical shorts, and rated for support of a minimum 150% of the weight of the equipment mounted on the rack.

Facility Power Requirements

The power supply must be capable of supporting the total power requirements of the system and must provide proper grounding of the equipment.

Each frame requires an AC power source within 2m of the frame. The receptacles must have three-prong grounding, with ground connected to protective earth.

The power supply must provide 100–250V, 47–70Hz, single-phase AC power. The power supply must be protected by a 20A circuit breaker.

It is recommended that the power supply be EMI/RFI filtered, and protected against power surges. An Uninterruptable Power Supply is recommended. It is further recommended that the overall power supply system be Ground Fault Interrupt protected, as further precaution against electrical shock.

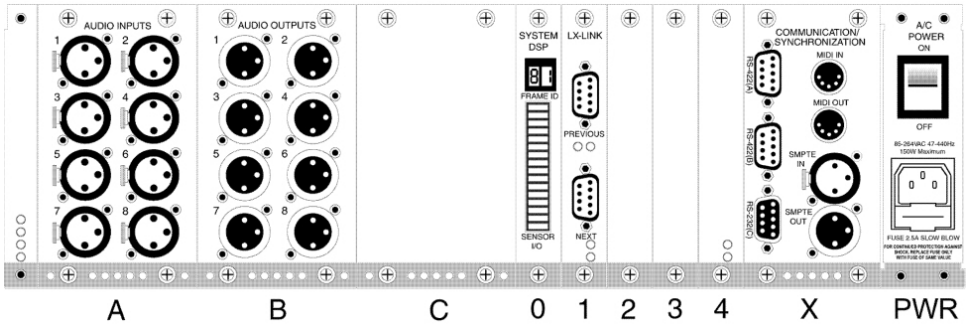
The LX-300 chassis uses a standard IEC power connector.

LX-300 IEC AC power connector	
Pin	**Name**
1	Hot
2	Gnd
3	Neutral

Component Specifications

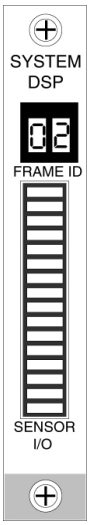
LX-300 Chassis

This illustration shows a typical Matrix[3] back panel. The valid slot assignments are provided below.



LX-300 Chassis Slot Assignments	
Slot	**Cards Allowed in this Slot**
A	LX-AI8 analog audio input *LX-AO8 analog audio output *LX-ADAT digital audio input/output *LX-AES digital audio input/output *except if LX-AO8 is installed in Slot C
B	LX-AI8 analog audio input LX-AO8 analog audio output LX-ADAT digital audio input/output LX-AES digital audio input/output
C	LX-CBR CobraNet audio interface *LX-AO8 analog audio output * except if LX-AO8, LX-ADAT, or LX-AES is installed in Slot A
0	LX-DSP system processor
Any of: 1, 2, 3, 4	LX-LNK frame link interconnect* LX-ELC EtherTracks/Wild Tracks* LX-EXP expansion DSP (maximum 3; two if LX-ELC installed) LX-VRA variable room acoustics system© *maximum one per frame
X	LX-COS communications/synchronization

LX-DSP system processor card



Performs matrix mixing automation and control, using the TI-TMS320C6701 Digital Signal Processor. This DSP card also holds 32MB SDRAM and 8MB Flash memory, allowing stand-alone operation once programmed with an automation cue list. The DSP is also interfaced to four relay contact outputs, four digital sensor inputs and two analogue sensor inputs for control and reading of external devices.

Channels

Maximum 160/512 audio I/O channels

Dynamic Range

192dB

Note: Digital Logic Inputs

LX-DSP cards with serial numbers 0000–0688 require an external pull-up resistor to use the Digital Logic Inputs. Connect a 1/4W 2K2 resistor between the Digital Logic Input terminal and the +5V reference voltage. One resistor is required for each Digital Logic Input.

Note: Special Frame ID Numbers

The Frame ID is used both to set the Frame ID and to perform special maintence operations. The following Frame IDs will erase files stored in the LX-DSP flash memory.

1. Power-off the Frame.
2. Set the Frame ID as desired:

    87 Erase System Flash

    88 Erase NV RAM

    89 Erase User Flash
3. Power-on the Frame. When the Frame has completed the operation, the Frame Status will display **REBO.**
4. Power-off the Frame.
5. Set the Frame ID to its original value.
6. Power-on the Frame.
7.

The following command will erase the EtherTracks firmware. Use the same procedure as described for erasing flash memory, above.

83 Erase LX-ELC Flash

LX-DSP Pin-outs

LX-DSP sensor I/O terminal block			
Pin	**Name**	**Signal**	**Use**
1	+5V ref	+5V	Reference voltage
2	ADC1	0–5V	Analog sensor input
3	ADC2		
4	D1	-TTL (active low with internal 2K2 pull-up resistor)	Digital logic input
5	D2		
6	D3		
7	D4		
8	GND ref	GND	Reference ground
9 10	R1	500mA@48V max	Relay N/O dry contact
11 12	R2		
13 14	R3		
15 16	R4		

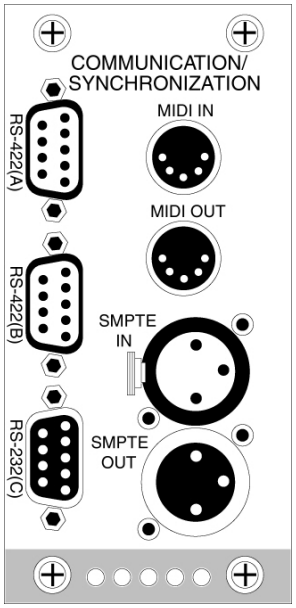
LX-EXP expansion DSP

The expansion DSP card provides additional processing power for the frame.

LX-VRA Variable Room Acoustics System DSP

The VRAS DSP provides reverb, and early and late reflections capabilities for the Matrx^3.

LX-COS comm/sync card



Provides a communications link to the CueStation host computer system (or the computer hosting your custom control software), a SMPTE Time Code signal generator and receiver, and MIDI i/o.

Ports

RS232 serial, RS422 serial, MIDI In/Out, XLR SMPTE In/Out

LX-COS Pin-outs

LX-COS nine-pin male RS232 serial port (DTE)		
Pin	**Name**	**Use**
1	—	—
2	RX	Receive data
3	TX	Transmit data
4	—	—
5	GND	Reference ground
6	—	—
7	RTS	Ready to send (not required)
8	CTS	Clear to send (not required)
9	—	—

LX-COS nine-pin female RS422 serial port (DTE)		
Pin	**Name**	**Use**
1	GND	Shield
2	—	—
3	GND	Shield
4	TX+	Transmit data
5	TX–	Transmit data
6	—	—
7	—	—
8	RX+	Receive data
9	RX–	Receive data

LX-COS five-pin DIN female MIDI In		
Pin	**Name**	**Use**
1	—	—
2	GND	Shield
3	—	—
4	RD+	Receive data
5	RD–	Receive data

LX-COS five-pin DIN female MIDI Out		
Pin	**Name**	**Use**
1	—	—
2	GND	Shield
3	—	—
4	+5V	Reference current
5	Data	Send data

LX-COS XLR female SMPTE In		
Pin	**Name**	**Use**
1	GND	Shield
2	+data	+Balanced analog input
3	-data	–Balanced analog input

LX-COS XLR male SMPTE Out		
Pin	**Name**	**Use**
1	GND	Shield
2	+data	+Balanced analog output
3	-data	–Balanced analog output

LX-LNK frame interconnect

Provides a high-speed connection between frames. Carries command data and 24bit/48kHz digital audio data. An optional optical transceiver module extends the maximum transmission range.

Transmission Range

Up to 30m (100 feet) using copper cable.

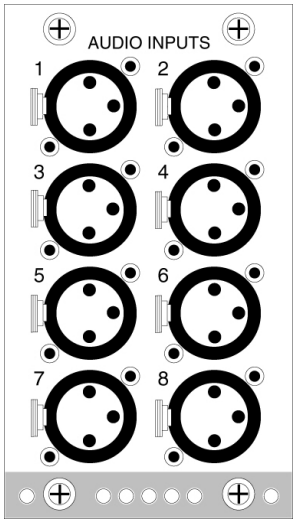
Up to 500m (1000 feet) using multimode fiber optic cable and the optional fiber optic driver module.

Pin-outs

The link cable is a standard Fibre Channel™ cable. These have precision matched conductors. Please do not try to build these without the required tooling and test equipment.

LX-AI8 analog audio input

Provides eight balanced-XLR analog audio signal inputs. Performs 24bit/48kHz digital conversion with three conversion scaling settings.



Channels

Eight analog inputs.

Frequency Response

20Hz — 20kHz.

Sampling

24bit/48kHz.

Dynamic Range

98dB.

S/N Ratio

-74dB.

THD

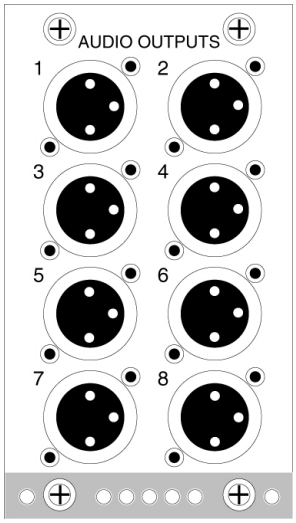
<0.005% THD from 20Hz - 20KHz at +26dBu

Input Load Impedance

7.5KΩ nominal.

Pin-outs

LX-AI8 balanced female XLR		
Pin	**Name**	**Use**
1	GND	Shield
2	Signal+	+Analog balanced input
3	Signal–	–Analog balanced onput



LX-AO8 analog audio output

Provides eight balanced-XLR analog audio signal outputs. Performs 24bit/48kHz digital conversion with three conversion scaling settings.

Channels

Eight analog outputs.

Frequency Response

20Hz — 20kHz.

Sampling

24bit/48kHz.

Dynamic Range

98dB.

S/N Ratio

-74dB.

THD

<0.005%

Output Load Impedance

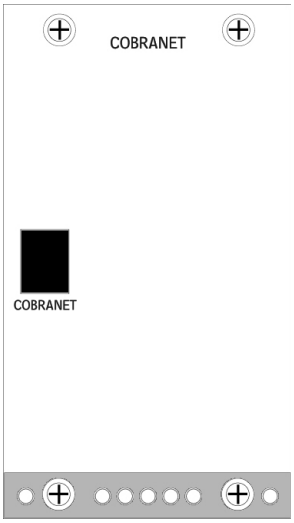
600Ω nominal.

Pin-out

LX-AO8 balanced male XLR		
Pin	**Name**	**Use**
1	GND	Shield
2	Signal+	+Balanced analog output
3	Signal–	–Balanced analog output

LX-CBR CobraNet audio interface

Provides CobraNet 20bit/48kHz digital audio data transmission over Ethernet networks. Please refer to the Peak Audio website for details (http://www.peakaudio.com/).



Pin-out

CobraNet communications: Use standard CAT5 cabling.

Note: LED Status Indicator

Green LED indicates normal operation.

Red LED indicates a fault or error condition.

LX-ELC Ethernet/Wild Tracks

Provides Ethernet communications link between external hardware and the Matrix^3 system. Also provides a SCSI hard drive interface for use with Wild Tracks playback.

Pin-out

Ethernet: Use standard CAT5 cabling.

SCSI disk connection: Ultra-SCSI. Use standard SCSI cabling. The connector is a VHDCI.

Appendices 8

Templates	page 8-1
Glossary	page 8-8
Questionnaire	page 8-11

Templates

Every Matrix³ installation includes a folder named `Templates.` This folder includes a set of files that include information used by both CueStation and External Subcue Editor to build command entries.

Each file, or "Template" contains related command specifications. Some of these are used to control hardware external to the Matrix³ hardware. This is the origin of the term "External."

Over time, however, this same facility has been used to control subsystems within the Matrix³ product family. It may help to think of these subsystems as "external" to the basic mixer functions of the Matrix³.

External Subcue Editor subcues are used to create commands that will be triggered when recalling a cue. The **LX-300** menu in the **Log** and **Transport** windows uses this same facility to create commands that control LCS subsystems interactively.

Templates are specified in a two-level hierarchy: *Type* and *Command*.

All commands of a given type are stored in a single file. Commands can usually be specified for a particular frame or for all frames. Sometimes it makes sense for commands only to be specified for all frames.

Each External Subcue can have one or more command entries. A wait time may be specified to control the inter-command timing in a subcue. The wait time specifies the time in seconds to wait from before issuing the command. The **Port** field specifies the port that will issue the message. If the command is to be issued to an LCS subsystem, the port is set to **BoxNet.** Otherwise the port can be specified to either of the RS-422 ports, the MIDI out port, or the RS-232 port.

The following guide is a reference to the command types and individual commands and parameters for each type.

Templates Release

04 September 2001:v010904
Provides self-documentation of the Templates set. Every time a new set is released, this command will be updated with the release date of the set.
Parameters: None.

Hardware Status

These commands all result in a message printed in the Log window. Sometimes it is useful to place one or more of these in a startup cue, so that the log window will contain system diagnostics.

Check Temperature
Parameters: None.

Check Voltage
Parameters: None.

Check Firmware Version
Parameters: None.

Check Flash Memory Utilization
Parameters: None.

Check SDRAM Memory Utilization
Provides information regarding how much RAM is in use
Parameters: None.

Check DSP Utilization
Parameters: None.

Request CASL Report
Provides information on the current efficiency of the DSP's mix engine.
Parameters: None.

List Debug Messages
Provides a listing of diagnostic information created by systems such as EtherTracks during startup. Sometimes this can be helpful in trouble-shooting.
Parameters: None.

Reset Debug Messages Memory
Clears the memory allocated for debug messages.
Parameters: None.

Get LX-300 Time and Date
Parameters: None.

Flash Cue Files

These commands are used in the **LX-300** menu to record system information to non-volatile flash memory. This will typically include mixer configuration, automation data, and often a startup cue recall to set the system to a specified state.

Start Flash Cue Record (one frame)
After initiating this command, all subsequent data received by the DSP will be recorded to the specified flash file, until "Stop Cue Record" is issued.
Parameters:
`File ID (0–127)`

Boot Cues from Flash
Parameters:
`File ID (0–127)`

Boot Cues from Flash (one frame sends to all)
After the system is rebooted and the system has waited the specified time, the specified file ID will be played back.
Parameters:
`File ID (0–127)`
`Delay, seconds (10–127)`

Stop Flash Cue Record
Parameters: None.

Start Flash Cue Record (all frames)
Parameters:
`File ID (0–127)`

Boot Cues from Flash (all frames send to self)
After the system is rebooted and the system has waited the specified time, the specified file ID will be played back.
Parameters:
`File ID (0–127)`
`Delay, Seconds (10–127)`

Don't Boot Cues from Flash
After restarting, the system will not play back a file from flash memory.
Parameters: None.

Erase User Flash File
Erases the specified file ID.
Parameters:
`File ID (0–127)`

Play Back Flash Cues (one frame sends to all)
Parameters:
`File ID (0–127)`

Play Back Flash Cues (all frames send to self)
Parameters:
`File ID (0–127)`

Play Flash File
After initiating this command, the data for the specified File ID will be broadcast to all linked frames in the system.
Parameters:
`File ID`

Erase Flash

Erase Main Flash on All Frames
Erases the main system firmware on all frames. Does not erase user flash files.
Parameters: None.

Erase EtherTracks Flash on All Modules
Erases the EtherTracks firmware on all frames that have an LX-ELC installed.
Parameters: None.

Hardware Control

Set Relay
Changes the specified relay to the specified position.
Parameters:
`Relay select, menu`
`Action, menu`

Set Mix Point Limit
Configures the mixer plugin. Typically this is set via the Configure Mixer window
Parameters:
`Mix Point Limit (0–2048)`

Analog I/O dBu FS
Parameters:
`Slot, menu`
`Channel (1–8)`
`Full Scale Setting, menu`

8 Ch. Analog I/O dBu FS
Controls the dBU Full Scale setting for all eight channels of an analog I/O card in the specified frame and slot.
Parameters:
`Slot, menu`
`Full Scale Setting, menu`
`Full Scale Setting, menu`
`Full Scale Setting, menu`
`Full Scale Setting, menu`
`Full Scale Setting, menu`
`Full Scale Setting, menu`
`Full Scale Setting, menu`
`Full Scale Setting, menu`

Set LX-300 Front Panel Message
Scrolls specified message to front panel of specified frame.
Parameters:
`Label String (up to 120 chars)`

Set LX-300 Time and Date
Sets the real time clock of the LX-300
Parameters:
`Hour (0–23)`
`Minutes (0–59)`
`Seconds (0–59)`
`Year (2001–2107)`
`Month, menu`
`(January–December)`
`Day in the week, menu`
`(Sunday–Saturday)`
`Day in the month (1–31)`

Set LX-300 Temperature Limits
Parameters:
`Yellow threshhold, degrees C`
`Red threshhold, degrees C`

Raw Data

Specifies raw data in hex for the desired message

Hex Format
Parameters:
`hex-encoded data.`

MIDI

Program Change
Parameters:
`Channel (1–16)`
`Program # (0–127)`

Note
Parameters:
`Channel (1–16)`
`Note # (0–127)`
`Velocity (0–127)`
`Duration (ms)`

Note On
Parameters:
`Channel (1–16)`
`Note # (0–127)`
`Velocity (0–127)`

Note Off
Parameters:
`Channel (1–16)`
`Note # (0–127)`
`Velocity (0–127)`

Control Change
Parameters:
`Channel (1–16)`
`Controller # (0–127)`
`Value (0–127)`

Polyphonic Key Pressure
Parameters:
`Channel (1–16)`
`Note # (0–127)`
`Pressure (0–127)`

Channel Pressure
Parameters:
`Channel (1–16)`
`Pressure (0–127)`

Pitch Bend
Parameters:
`Channel (1–16)`
`Value (-8192 to +8191)`

MMC

MIDI Machine Control (MMC) is part of the MIDI specification, and provides a uniform way to control devices such as hard disk recorders, audio workstations, and even SMPTE converters and generators. Device ID's range from 0 to 127. The value "127" is interpreted as the broadcast ID. All MMC devices will respond to this ID.

Stop
Parameters:
`Device ID (0–127)`

Play
Some MIDI Machine Controllable devices respond to the "Play" message, others to "Play Deferred," and some to either.
Parameters:
`Device ID (0–127)`

Play Deferred
Some MIDI Machine Controllable devices respond to the "Play" message, others to "Play Deferred," and some to either.
Parameters:
`Device ID (0–127)`

Fast Forward
Parameters:
`Device ID (0–127)`

Rewind
Parameters:
`Device ID (0–127)`

Record Strobe
Parameters:
`Device ID (0–127)`

Record Exit
Parameters:
`Device ID (0–127)`

Record Pause
Parameters:
`Device ID (0–127)`

Pause
Parameters:
`Device ID (0–127)`

Eject
Parameters:
`Device ID (0–127)`

Locate
For most playback applications, "Locate," "Play," and "Stop" will be all you need from MMC.
Parameters:
`Device ID (0–127)`
`Rate (0–3)`
`Time`

CueMixer RIF-108

Enable RIF Classic Mode
Parameters: None.

Disable RIF Classic Mode
Parameters: None.

Enable CueMixer RIF-108
Parameters: None.

Disable CueMixer RIF-108
Parameters: None.

Set MIDI-X 38.4K Console
Parameters: None.

Set MIDI-X 31.25 Classic
Parameters: None.

Enable Window Follow
Parameters: None.

Disable Window Follow
Parameters: None.

Console Modes
Parameters:
`Enable Masters`
`Enable Console Faders`
`Enable Wild Tracks (LD-88 only)`
`Enable Bus Levels`
`Enable Outputs`
`Enable Aux Outputs`
`Enable Virtual Groups`
`Enable User A`

EQ/Trim Modes
Parameters:
Enable Input EQ/Delay
Enable Output EQ/Delay
Enable Aux Out EQ/Delay)
Enable Input Trims
Enable Wild Tracks Trims (LD-88 only)
Enable Bus Level Trims
Enable Output Trims
Enable Aux Out Trims

Other Modes
Parameters:
Enable Transport
Enable Stop
Enable Editing
Enable User A and B Buttons

Set User A
Parameters: None.

Set User B
Parameters: None.

Set User A Mode Code
Parameters:
Page (1–16)
Button (1–16)
Cue ID (0–127

Enable Cue 0 RIF
Parameters: None.

Set Button Cue Overlay
Parameters:
Button Index (0–31, 127=all)
Cue ID (0–16382, 16383=disable)

External Control

These commands are provided as a way of controlling either the current Matrix^3 system or a remote Matrix^3 system.

Recall Cue
Parameters:
Cue ID

Select Cue List
Parameters:
CueList ID (0–127)

Update Cue
Causes the specified Cue ID to be updated via Capture Updates. In this way a cue may be used to store the state of a system for later restoration after a programmed sequence of events.
Parameters:
Cue ID (0–16383)

Capture Cue
Causes the specified Cue ID to be updated via Capture Differences. In this way a cue may be used to store the state of a system for later restoration after a programmed sequence of events.
Parameters:
Cue ID (0–16383)

MIDI Program Change Receive Enable
After receiving this command, the Matrix^3 will respond to program change commands.
Parameters:
MIDI channel (0–15)

MIDI Program Change Receive Disable
After receiving this command, the Matrix^3 will no longer respond to program change commands.
Parameters: None.

Trigger Cues via Timecode Enable
Causes system to respond to MIDI Time Code.
Parameters: None.

Trigger Cues via Timecode Disable
Causes system to ignore MIDI Time Code.
Parameters: None.

Trigger Cue via Digital Logic Input
Parameters:
Trigger ID (1–16)
Trigger Enable
Trigger Source, menu
Transport Action, menu

Trigger Transport Action via Digital Logic Input
Parameters:
Trigger ID (1–16)
Trigger Enable
Trigger Source, menu
Recall Cue ID (0–16383)

Trigger Cue via MIDI
Parameters:
Trigger ID (1–16)
Trigger Enable
Source MIDI event type, menu
MIDI Channel (1–16)
Note/Controller # (0–127)
Velocity/Controller value (0–127)
Cue ID (0–16383)

Trigger Transport Action via MIDI
Parameters:
Trigger ID (1–16)
Trigger Enable
Source MIDI event type, menu
MIDI Channel (1–16)
Note/Controller # (0–127)
Velocity/Controller value (0–127)
Transport Action, menu

Labels

These provide you with the ability to change labels on the fly with a cue. Designed initially for use with CueConsole, these have now become the preferred method for setting input, output, Vgroup, and bus labels.

Set Channel Label
Parameters:
Channel Type, menu
Channel # (1–512)
Label String (max 16 chars)

Set 2 Channel Labels
Parameters:
Channel Type, menu
1st Channel # (1–497)
1st Label String (max 16 chars)
2nd Label String (max 16 chars)

Set 8 Channel Labels
Parameters:
Channel Type, menu
1st Channel # (1–497)
1st Label String (max 16 chars)
2nd Label String (max 16 chars)
… (repeated)
8th Label String (max 16 chars)

Set 16 Channel Labels
Parameters:
Channel Type, menu
1st Channel # (1–497)
1st Label String (max 16 chars)
… (repeated)
16th Label String (max 16 chars)

Set Parameters

These provide a technique for changing a single control point value, even those that are not captured in cues, such as Manual System Level and Trims.

Set Console Fader
Parameters:
Channel # (1–128)
Level, dB (-100.0 to +10.0 decimal value)

Set Console Bus Assign
Parameters:
Channel # (1–128)
Switch # (1–16)
Active

Set Console Mute
Parameters:
Channel # (1–128)
Mute

Set Console Aux
Parameters:
Channel # (1–128)
Aux # (1–128)
Level, dB (-100.0 to +10.0 decimal value)

Set Input Trim Fader
Parameters:
Channel # (1–128)
Level, dB (-100.0 to +10.0 decimal value)

Set Wild Tracks Trim Fader
Parameters:
Channel # (1–128)
Level, dB (-100.0 to +10.0 decimal value)

Set Bus Level Trim Fader
Parameters:
Channel # (1–128)
Level, dB (-100.0 to +10.0 decimal value)

Set Output Trim Fader
Parameters:
Channel # (1–128)
Level, dB (-100.0 to +10.0 decimal value)

Set Aux Trim Fader
Parameters:
Channel # (1–128)
Level, dB (-100.0 to +10.0 decimal value)

Set VGroup Fader
Parameters:
Channel # (1–128)
Level, dB (-100.0 to +10.0 decimal value)

Set VGroup Mute
Parameters:
Channel # (1–128)
Mute

Set Manual System Level
Parameters:
Level, dB (-100.0 to +10.0 decimal value)

Set Matrix Crosspoint
Parameters:
Bus # (1–64)
Output # (1–128)
Level, dB (-100.0 to +10.0 decimal value)

Set Parameter
Parameters:
Category (0–127)
Index 1 # (1–128)
Index 2 # (1–128)
Level, dB (-100.0 to +10.0 decimal value)

Set Matrix Crosspoint with Fade Time
Parameters:
Bus # (1–256)
Output # (1–512)
Level, dB (-100.0 to +10.0 decimal value)
Fade Tune (tenths of a second)

Set 2 Matrix Crosspoints with Fade Times
Parameters:
(1) Bus # (1–256)
(1) Output # (1–512)
(1) Level, dB (-100.0 to +10.0 decimal value)
(1) Fade Tune (tenths of a second)
(2) Bus # (1–256)
(2) Output # (1–512)
(2) Level, dB (-100.0 to +10.0 decimal value)
(2) Fade Tune (tenths of a second)

Set 4 Matrix Crosspoints with Fade Times
Parameters:
(1) Bus # (1–256)
(1) Output # (1–512)
(1) Level, dB (-100.0 to +10.0 decimal value)
(1) Fade Tune (tenths of a second)
… (repeated)
(4) Bus # (1–256)
(4) Output # (1–512)
(4) Level, dB (-100.0 to +10.0 decimal value)
(4) Fade Tune (tenths of a second)

Set 6 Matrix Crosspoints with Fade Times
Parameters:
(1) Bus # (1–256)
(1) Output # (1–512)
(1) Level, dB (-100.0 to +10.0 decimal value)
(1) Fade Tune (tenths of a second)
… (repeated)
(6) Bus # (1–256)
(6) Output # (1–512)
(6) Level, dB (-100.0 to +10.0 decimal value)
(6) Fade Tune (tenths of a second)

Set 8 Matrix Crosspoints with Fade Times
Parameters:
(1) Bus # (1–256)
(1) Output # (1–512)
(1) Level, dB (-100.0 to +10.0 decimal value)
(1) Fade Tune (tenths of a second)
… (repeated)
(8) Bus # (1–256)
(8) Output # (1–512)
(8) Level, dB (-100.0 to +10.0 decimal value)
(8) Fade Tune (tenths of a second)

Set Input EQ (single band)
Parameters:
Channel (1–128)
Band (1–8)
Type, menu
Value, dB (-20.0 to +20.0 decimal value)
Frequency, Hz (0–20000)
Q, (0.5, 10.0)

CobraNet

Set Diagnostic Reporting Level
Parameters:
Diagnostic Reporting Level, menu

Set Bundle ID
Parameters:
Channel Range, menu
Bundle ID (0–65535 – Note: IDs 1–255 are reserved for multicast.)

Set CobraNet IP
CobraNet cards do not need to have unique IP addresses. However this facilitates diagnostics using network MIB browser tools.
Parameters:
IP address

Set CobraNet Conductor Priority
Parameters:
Priority (0–255)

List CobraNet Parameter
Parameters:
Parameter Type, menu

Set Number of Unicast Receivers
Parameters:
CobraNet Tx module, menu

List Number of Unicast Receivers
Parameters:
CobraNet Tx module, menu

Set Number of Audio Channels
Parameters:
CobraNet Tx module, menu
Number of audio channels for this transmitter (0–8)

List Number of Audio Channels
Parameters:
CobraNet Tx module, menu

List Tx parameter
Parameters:
CobraNet Tx module, menu
Tx parameter, menu

List Rx parameter
Parameters:
CobraNet Rx module, menu
Rx parameter, menu

List Error Parameter
Parameters:
CobraNet Rx module, menu

List Number of Unique Audio Channels
Parameters:
Currently, menu

EtherTracks

Check EtherTracks Firmware Version
Parameters: None.

List EtherTracks IP Address
Parameters: None.

List EtherTracks MAC Address
Parameters: None.

Open TCP Cue Connection
Parameters:
IP
Port (0–99999)
Connection ID (0–16383)
Group ID (0–16383)

Recall Cue on Connection Group
Parameters:
Connection Group ID (0–16383)
Cue ID

Close Connection Group
Parameters:
Connection Group ID (0–16383)

TCP Cue Recall

TCP cues are provided as a way to configure one LCS system with EtherTracks to recall cues on another remote LCS system with EtherTracks. TCP/IP communication is made from system to system to achieve remote cue recall. Using this system, a single master system may be configured to recall the same cue on a set of remote systems.

The concept of Connections and Groups are used. A Connection is a single point to point messaging path from sender to target. A Group is a collection of Connections. If a sender recalls a cue on a Group, then all Connections within this Group will be sent a message to recall the specified cue.

If you set the same Connection ID to more than one system, then when you recall a cue on the Connection, only one of these systems will receive the cue recall message. This feature is set up so that if you have two EtherTracks cards in a single linked target system, you can open a connection to both cards using the same Connection ID in order to achieve a redundant control path. When a cue is recalled to this Connection ID, the “cue recall” message will only be sent once.

Open TCP Cue Connection
Parameters:
Connection ID
Group ID

Recall Cue on Connection Group

Parameters:
Group ID
Cue ID
Note: the specified cue ID is recalled on all connections made in the specified group. Only one cue recall will be made to a unique connection ID. In this way multiple duplicate connection ID's may be specified to provide redundant communication paths.

Close Connection Group

Parameters:
Group ID

SpaceMap

Stop Trajectory

Parameters:
First Bus
Number of Buses

ALC

ALC Setup

Parameters:
Enable
Source channel type, menu
Source Channel (1–512)
Sample duration (0–4095 seconds, 0=off)
Source minimum (0–127 dB)
Source maximum (0–127 dB)
Calibration offset (-64 to +63 dB)
Destination category, menu
Index 1 (1–512)
Destination minimum (-64 to +63 dB)
Destination maximum (-64 to +63 dB)

ALC Do

Parameters:
ALC ID #, menu
Enable
Change target parameter
Print
Source minimum (0–127 dB)
Source maximum (0–127 dB)
Calibration offset (-64 to +63 dB)
Destination category, menu
Index 1 (1–512)

CueConsole

Clear CueConsole Mapping

Parameters:
Chain ID (127=all)
Module ID (127=all)

Add a CueConsole Mapping

Parameters:
Chain ID (127=all)
Module ID (127=all)
Start Column
Num Columns
Category, menu
Start Index

Set Single CueConsole Mapping

Parameters:
Chain ID (127=none)
Module ID
Category, menu
Start Index

Set Dual CueConsole Mapping

Parameters:
Chain ID (127=none)
Module ID
Start Column A
Num Columns A
Category A, menu
Start Index A
Start Column B
Num Columns B
Category B, menu
Start Index B

Glossary

Add-On (BeOS) An integrated application/utility, accessible through right-click context menus. *Zip-O-Matic* is one example.

ALC Automatic Level Control, a system that automatically adjusts the gain to maintain a relatively constant level of sound output.

archive (BeOS) A method of packing many files into a single file, in preparation for a file transfer. See *Zip-O-Matic*.

Autofollow (CueStation) A method to trigger cues in sequence, without user intervention. When a cue is triggered, its autofollow entry is triggered at the same time (or after a specified wait period). Autofollow wait times are cumulative, but the wait times for the cues they reference are not cumulative.

automation The process whereby system settings may be stored and retrieved, with automatic interpolation between successive value where appropriate.

automation solo An operation mode that allows a selected channel or channels to be automation-enabled while non-selected channels are not.

Aux Short for "auxiliary send." A signal bus that allows routing directly from the Input Console to an output, without passing through the Matrix section of the mix architecture.

bandwidth (network) The data-carrying capacity of a network connection.

boot (Matrix^3, workstation) To start-up an electronic device. When a device boots, it loads its basic operating software, usually checks its component status, and then loads an advanced operating system or other software.

bundle (CobraNet) The basic data unit used by *CobraNet*. Up to 8 audio channels may be carried in a bundle, and bundles may be addressed to a specific device, or to all devices.

bus assigns Control points used to direct input signals to particular buses of the matrix. Designed to emulate a set of on-off push buttons in the Console window.

capture (CueStation) A command that retrieves the current values of control points and saves them into a set of subcues.

CD-ROM (workstation) Compact Disc Read-Only Memory. A computer file storage device, and the compact disc media that it uses.

channel An audio signal path, or more generally any signal path.

client (network, workstation) See client-server.

client-server (network) Describes a relationship between computer applications, in which one program (the client) makes service requests to another program (the server), which fulfills the request. As used by CueStation, the client-server communications model allows several automation programmers to work simultaneously on a single project.

CobraNet (Matrix^3, network) A data protocol that permits real-time distribution of uncompressed, multi-channel digital audio signal over an Ethernet network.

context menu (BeOS) A menu that contains appropriate commands for an object or item that has been right-clicked. Often quicker than using the menu bar.

control point Any parameter value that may be varied by either manual or automation control.

control surface A hardware interface that allows direct manipulation of control parameters by means of multiple physical controls such as faders and knobs, used in lieu of mouse and keyboard data entry.

cue A data structure, or "bundle" of control data that is used to store and recall the state of the mixer, and which also may contain commands to control external devices.

cueconsole A modular control surface for the Matrix^3 system.

cuelibserver The server software that mediates between the CueStation application and its family of Helper Applications.

CueMixer A small hardware control surface for the Matrix^3 system, with eight motorized faders and 32 buttons. Model RIF-108

cue-on-deck The next cue in a cue list, in position to be triggered by the next "GO" command.

CueStation The principal programming software for the Matrix^3 system.

debugger A system tool used to troubleshoot software crashes.

derived node (SpaceMap) A special type of node.

Deskbar (BeOS) Appearing in the top right corner of the BeOS Desktop by default, with the BeOS logo, the Deskbar provides: the Be menu, accessible by clicking the BeOS logo; a status bar that typically displays icons, the time, and other icons; and an application list, that displays the active applications and provides application context menus when an application entry is right-clicked.

Desktop (BeOS) The Desktop is the background, over which the Deskbar, volume icons, application and file icons, trash can, and application windows and dialogs appear. When you right-click the Desktop (any area not used by the objects which are on the Desktop), a context menu is shown.

dialog A pop-up window that requests information or confirmation before finishing a command. The Save As window is an example.

directory (BeOS) A special kind of file, which is used to organize other files. Its useful to imagine directories as file folders, capable of holding files and other directories.

DSP (Matrix^3) Digital Signal Processor. The heart and soul of the Matrix^3, the DSP performs complex math transformations on digital data in real-time.

dynamic Having the ability to change over time.

embedded automation A control system that is built into a device, so that it does not require external control from another device.

EQ Equalization. Frequency-dependent gain control of an audio signal, allowing the shaping of the audio spectrum.

Ethernet (network) A local-area network protocol that uses a bus (one device connected to the next, connected to the next, and so on) or star (all devices connected to a hub or switch) topology.

EtherTracks (Matrix^3)

Expand-O-Matic (BeOS) An Add-On that retrieves files from an archive. See also Zip-O-Matic.

external devices Any devices that are not an integral part of a Matrix^3 system but which may be interfaced to it as part of a show control system.

external subcue A subcue type that contains commands to control devices outside of the basic mix architecture. The controlled devices may be entirely external devices, or they may be Matrix^3 options such as Wild Tracks.

Fade time The length of time that it takes a fader, pan, or matrix crosspoint to change continuously from one programmed value to the next in a sequence of cues.

File Finder (BeOS) A file search utility, available through the Deskbar menu.

fire (CueStation) To trigger a cue manually, either by pressing the Go button, or by double-clicking a cue.

firmware (Matrix^3) Programs or data written to read-only memory, and which are used by the operating system.

flash memory (Matrix^3) A special type of read-only memory, which can be completely erased and reprogrammed.

folder (BeOS) A file directory; usually used when referring to file system management through the graphic interface.

frame An LX-300 rack-mount enclosure, the basic hardware building block of a Matrix^3 system. Each LX-300 frame may contain a variety of plug-in cards for audio I/O, communications, audio playback, etc.

FTP (BeOS) File Transfer Protocol. A communications protocol for sending files over TCP/IP.

gain A measure of the difference in signal level induced by a level control point.

gateway (network) A combination of hardware and software that acts as an intermediary between two different networks, performing data and protocol translation, and other functions.

Helper Apps (CueStation) Applications that work in conjunction with the core CueStation program to provide additional automation programming functionality.

host computer (network, workstation) A computer that is accessed through a network connection, by a user working at a remote system. The host computer generally acts as the server.

hub-based network (network) The hub provides a common connection point for devices in the network. The hub sends incoming data to all connected devices; the devices are responsible for determining whether the data should be accepted.

import To access a file that was not created by the application you are currently using.

IP address (network) An identifier for a device on a TCP/IP network. The IP address is rather like a telephone number or street address: it identifies the sender or recipient for a data package.

isolate A control mode which allows a selected channel to be removed from automation control.

L-C-R Left–Center–Right, describing speaker positions.

legacy application A software application that is a relic of an earlier stage of computer software evolution.

LFE Low frequency effects Strictly speaking, a supplementary subwoofer channel that supplements a number of full-range channels in a surround sound system. Derives from cinema sound practice.

linked cues Cues in a cue list that form a timed sequence, with one cue automatically recalling the next in the list.

link (SpaceMap) To connect a virtual or derived node, and its associated speaker nodes, so that signal can be distributed to/from the nodes.

local-area network (network) A network that covers a relatively small area (a single building or just a few buildings).

loop-back (network) Data communication in which the receiving device sends the data back to the transmitting device. Mainly used in troubleshooting.

Matrix A fader/patch control that allows any number of buses to be assigned to any number of outputs.

meta-controller A controller of controllers. A "super controller."

MIDI Musical Instrument Digital Interface. A serial protocol used for controlling a variety of equipment, most particularly including synthesizers and other musical devices, but also extended to the control of audio, video, lighting, and other equipment.

mult Short for "multiple." A means of splitting one signal into several parallel branches or "legs."

multed Short for "multiplied." Refers to a signal that has been split into parallel branches by the use of a "mult."

multicast (network) Data communication between a single transmitting device, and multiple receiving devices.

multitasking (BeOS) To do more than one thing at a time. When a computer multitasks, it switches between programs so rapidly that it seems all of them are running at the same time.

mute To shut off a signal at a particular point in the signal path. Used as both a noun and a verb.

NetPositive (BeOS) The web browser that is included with BeOS.

network (network) A group of two or more interconnected computers. See also: local-area network, hub-based network, switch-based network.

newsgroup (network) An on-line discussion forum available through the Internet. Requires the use of a news reader.

node One of the points that define a the geometry of a SpaceMap. Nodes may represent loudspeaker positions, groups of speakers, or points of silence.

on-line (network) You are on-line when you are connected to a data network.

off-line (network) You are off-line when you are not connected to a data network.

parameter One of the individual elements that define the state of a system.

parent cue A term sometimes used when describing the relationship between subcues and the cues that contain them (the cue is the parent of the subcues, the subcues are the children of the cue, and the subcues are siblings to each other).

passband A frequency region of signal that is acted upon by a band-pass filter.

PDF (BeOS) Portable Document Format, a standard file format for sharing documents between computers. The files are viewable using Adobe Acrobat Reader. There are alternative viewers available: for BeOS, use BePDF.

ping (BeOS) Packet Internet Groper, a utility used to determine if a given IP address is accessible over a network.

port A place of access to a system, typically that input or output connector where digital data cables are attached to connect pieces of equipment.

protocol (network) A standard that determines the format for sharing data between devices. Protocols generally determine the data type, error correction, compression technique, and acknowledgment signals that the devices will use.

proxy server (network) A server that acts as an intermediary between clients and a real server. If the proxy can fulfill a server request, it does; if it can not, it passes the request on to the real server. See client-server.

ps (BeOS) The PS command displays a list of active processes. Processes are programs, some of which are invisible to the user.

real-time (BeOS, Matrix^3) A system which responds to changes immediately, without interruption or pause.

recall To retrieve the data stored in a cue or subcue and to set system parameters to those stored values.

receiver (CobraNet) A device which is accepting CobraNet data.

reverse-engineering Simulation of the performance of a piece of equipment or a software application without benefit of the original schematics or code specification.

roll-back To restore a former, working set of files.

server (network, workstation) See client-server.

shortcut (BeOS) Keyboard equivalent to a mouse command.

signal path The route a particular signal takes through the mixer architecture. In conventional analog mixers the signal path follows physical audio circuit connections, but in a digital mixer the signal path is modeled mathematically.

SMPTE Society of Motion and Television Engineers: a reference standards organization. Sometimes used as shorthand for "SMPTE Time Code."

SpaceMap A graphic programming interface for the CueStation matrix, using a visual "map" of the loudspeaker layout and allowing recording and playback of moving sound trajectories.

spatial pan control (SpaceMap) A two-dimensional pan control. In traditional consoles, the pan control is one-dimensional: it scales along a line between two points (left-right, up-down, front-back). The spatial pan control scales along an area between three points. From an audience perspective, spatial pan control gives a three dimensional sound effect.

stack crawl Part of the output of the debugger. It looks indecipherable, but our programmers can make sense of it.

subcue A data structure, or "bundle" of control data that is used to store and recall the state of one subsystem of the mixer, and which also may contain commands to control external devices.

subcue librarian A CueStation Helper Application used to organize subcues of various types, and to mediate between the various Subcue Editors and CueStation.

switch-based network (network) The switch (a form of hub) provides a common connection point for devices in the network. The switch sends incoming data to the device it is addressed to. This reduces the amount of data being transferred over the network, making better use of its bandwidth.

TCP/IP (network) Transmission Control Protocol/Internet Protocol. A set of communications protocols used to connect devices over a network.

telnet (network) A terminal emulation program for use on TCP/IP-based networks. Provides a command-line interface to a server.

Terminal (BeOS) Provides a command-line interface to BeOS. Used to issue some esoteric commands which have no graphical equivalent.

ToolTips (BeOS) Pop-up help that appears when you hover your mouse cursor over certain controls.

topology (network) The layout of a network: common topologies include bus topology, in which the devices are connected one-after-another; ring topology, in which the devices are connected one-after-another, with the last computer linked back to the first; and star topology, where the devices are connected directly to a hub device.

Tracker (BeOS) The file browser used in BeOS. It displays the contents of file storage devices, usually as a list of folders and files. Double-clicking a folder will display the folders and files within that folder; double-clicking a file will open the file in appropriate application.

transmitter (CobraNet) A device which is sending CobraNet data.

trajectory (SpaceMap) The path which the spatial pan control will follow through the SpaceMap.

trigger To activate by means of a command. Often used synonymously with “recall.”

trim A signal level control point that may be manually regulate and cannot be automated. Used as a manual override or scaling control for the automation system.

triset (SpaceMap) A triplet of nodes. Each node will receive a proportion of the input bus signal, as distributed by the spatial pan control. The closer the spatial pan control is to a node, the greater the proportion of signal received by that node.

unity, unity-gain, unity-level Used to refer to a control point value that does not alter the signal level passing through it.

unlink (SpaceMap) To disconnect the signal distribution path between a virtual or derived node, and its associated speaker nodes.

unmount (BeOS) To (virtually) disconnect a file storage device from the system. See also mount, volume.

vamping Repeating a musical phrase for an unspecified length of time, typically while waiting for a performer to complete an improvisation.

VFader See Virtual Group Fader.

VGroups Virtual Groups. Sets of control points that may be controlled as a group by assigning them to a Virtual Group Fader.

Virtual Group Fader A CueStation fader that can simultaneously control several parameters that have been assigned to it.

volume (BeOS) A file storage device. Typical volumes include hard drives, CD-ROM drives and floppy drives. By default, only the BeOS boot hard drive is directly accessible by the user. To access other volumes, they must first be mounted. Note: with floppy drives, you must unmount the floppy when you eject it, and mount it when you insert it. BeOS will not automatically detect a floppy disk change. See also mount, unmount.

VRAS Variable Room Acoustics System. Specialized hardware developed by LCS to

Wait time The time interval between the moment a cue or subcue is recalled and the moment an individual control point begins its automated movement.

Wild Tracks The integrated hard disk playback option for the LX-300 mixer, allowing independent instant-start playback of up to 16 tracks per Wild Tracks card.

workspace (BeOS) Workspaces are multiple virtual monitors. You move between workspaces using Alt+F1 through Alt+F9,. If you click on an application title bar while doing this, the application will be moved to the new workspace.

workstation (network) A computer connected to a local-area network. Except for the host computer, workstations generally act as the client in a client-server relationship.

zip (BeOS) A nickname for archive, a method of packing many files into a single meta-file.

Zip-O-Matic (BeOS) An Add-On that archives files. An archive packs many files into a single meta-file. This is especially useful when transferring files through e-mail or FTP.

Guide Questionnaire

In the quick-changing world of software development, user guides are often a step behind the software itself. Such is the case with this guide.

So I'm going to take advantage of this situation, and make it work to everyone's advantage: I'm asking *you* to provide feedback so that I can make the next release even better.

To help you provide effective feedback, the following questions are being asked. Your responses will help improve the design and effectiveness of the guide.

I'll take your feedback in e-mail (support@LCSaudio.com), by fax (+1 (626) 836-4883), snail mail (LCS Documentation Dept., 130 East Montecito Ave #236, Sierra Madre, CA 91024, USA), or written on watermelon and sent by courier: whatever format works for you works for me!

DESIGN

The users guide is not intended to replace classroom training at this point. It is meant to serve as a pre-course reading material and as a post-course quick-reference. It may also find use as a reference for "in a pinch" sound programmers who are being asked to help a lead programmer, without having received the classroom training.

1) If you have not taken part in classroom training, I'd like to know whether this guide would currently help you survive in a real-life "just do it" situation where you do have sporadic access to a trained user.

If the guide currently falls short of this mark, please describe how it could be designed to help you land on your feet. "Design" refers to things like its major section breakdown, cross-referencing, indexing, glossary, and the like. "Content" issues are addressed in the next section.

Please also describe any design features that you like or find particularly useful.

2) If you have taken part in classroom training, I'd like to know whether this guide would have helped you prepare for the course and, if not, how it could be improved.

Again, I'm looking for design issues in this question: its major section breakdown, cross-referencing, and so on.

CONTENT

The content for the guide was selected for its ability to reduce the need to contact LCS technical support services. However, we know that our understanding of needed content may differ from what our customers would identify as necessary.

1) Please identify three "conceptual subjects" that should be included in the guide. These would be topics for expanding the "Understanding CueStation" section of the users guide.

2) Please identify three "Common Tasks" that should be included. These should be tasks that you, as an expert user, would assign to a newly-trained coworker; or tasks that you, a new user, want to see included.

3) Please identify two completely new major chapters for the user guide. These do not need to be chapters that fulfill the original design goal: they can be chapters that take the guide a direction that is most useful to you.

COMPREHENSION

Any guide that can't be understood by its readers is useless. We've tried to create an accessible guide that can be understood by new users, but that isn't so simplistic that it annoys experienced users.

1) As a new user, are you finding the "Understanding CueStation" chapter easy enough to read and understand? Are there any particular articles or paragraphs that we need to rewrite to be more accessible to you?

2) As a new user, can you easily follow the step-by-step instructions in the "Common Tasks" chapter? Please identify any parts that are difficult or poorly written.

3) As an experienced user, does "Understanding CueStation" help you? Are there any points where we should be going into greater "expert detail"?

ILLUSTRATIONS

The guide needs to be better illustrated. The difficulty we face is in knowing what our customers need to have illustrated: as subject experts, what is obvious to us may not be so obvious to a less-experienced user.

1) The "Understanding CueStation" section is in dire need of illustrations. Please read the articles in that chapter, and indicate where you would insert a screenshot or diagram that would clarify the accompanying text content. If there are exact locations, please quote a bit of the paragraph so that we can identify the troublesome text.

2) I've deliberately avoided screenshots in the "Common Tasks" section: studies indicate that most users end up ignoring important text information when menus are shown as pictures.

However, there may be instances where a screenshot really would speak a thousand words for the reader of the guide. If there are any "Common Tasks" screenshots that would make or break your understanding of the text, please tell us what they would show and where you'd place them!

GLOSSARY and INDEX

These are currently incomplete, but it doesn't hurt to get feedback early and often, so:

1) Are there any terms that you'd like us to add to the Glossary?

2) Indexing is an arcane magic all in itself, and we certainly don't expect any of you to write it for us! That said, though, there may be words, terms, or phrases that you're using in real life, that are synonyms for words/terms/phrases we've included in our index (akin to "petrol" vs "gas," and "hood" vs "bonnet"). Let us know what these synonyms are, and we'll expand our index to include them.

CROSS REFERENCES

The internal cross-referencing for the guide will be completed after the first release feedback is complete. In the meantime, we'd love to start creating a list of external (i.e., book, magazine, website) references.

1) Please list three or more books that you feel all sound programmers should read, with a brief description of what makes them so good.

2) Please recommend a few trade magazines, with a description of their content and why they're the ones you recommend.

3) Please list websites that you've found useful.

Future Additions

A number of additions are planned for the manual. Help me decide their priority by numbering them by their importance to you.

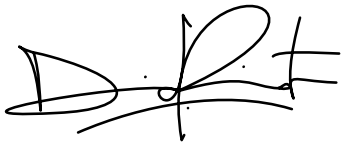
- Understanding Signal-to-Noise Ratios in Digital Systems
- Understanding the LX-LNK link network
- Wild Tracks (stand-alone manual)
- Using Time Code
- Understanding and Using Digital EQ
- Using VRAS — Variable Room Acoustics System (stand-alone manual)
- Understanding and Using ALC — Automatic Level Control
- Understanding and Using Dynamic Processing
- Understanding and Using Reverb
- Hardware Troubleshooting
- Software Troubleshooting
- Application Notes — Real Life Situations and Solutions (requires end-users to share their tips, tricks, and techniques)

And, of course, I am *very* interested in any topics that you want to have included. What will help (would have helped) you to better prepare for the in-class training? If you're an experienced user, what do you need the manual to contain as a reference or refresher?

Mistakes

I've no doubt that there are a few spelling, grammatical and technical errors that weren't spotted by me or the draft guide readers. If you find any, let me know!

Again, thank you for your assistance,



Document History

r1.3.1

Mar25 dcp

RB edits, verbatim; reduce font sizes for commands reference, templates to dictionary-sized.

r1.3

Mar25 dcp

RB edits.

Mar18 dcp

RB, RB, CK–edits.

Verify Tasks ref.

r1.2

Jan07 dcp

RB–editing.

Adjusted layout.

Feb04 dcp

Conrad–editing.

RB-editing.

Adjust layout.

Publish to HTML.

r1.1

Nov12 dcp

Added SpaceMap Tutorial.

RB–editing.

r1.02

Jun01 dcp

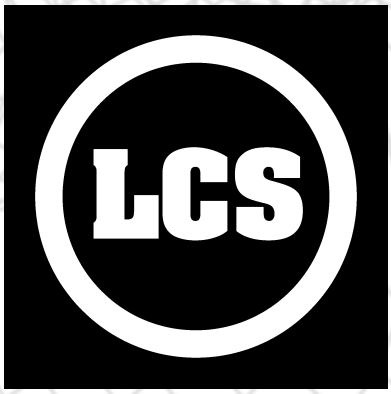
Added preliminary Externals (CASL) reference.

Added illustrations.

Colophon

The body text is set in Korinna, a recently revived typeface that first appeared in 1904, during the Art Nouveau period of design. It is complemented by headings set in Futura, which was designed by Paul Renner in 1927, during the Bauhaus period. Both faces are provided by Bitstream, Inc.

Layout and design was performed by WritersBlock Technical Communications, using Corel Ventura Publisher, and rendered directly to press output.



130 East Montecito Ave.
Sierra Madre, CA, 91024
Tel. 626.836.0446

2002-Mar-25
Rev. 1.3.1