C-Mexx Technical Pages

Connecting C-Console for remote operation

The Remote Editor

C-Console can respond to standard MIDI controller messages from other devices such as software MIDI controllers in sequencer applications or hardware MIDI controllers such as sliders, knobs and MIDI channel controllers from an external keyboard or dedicated MIDI control devices such as the Kenton Control Freak or Peavey PC1600x.

All you need to do to enable remote operation of the DS2416 is select **Connection** from the C-Console menu bar and check **MIDI Remote.** When this menu item is checked, remote operation is enabled. Uncheck it to temporarily disable processing of remote MIDI messages.

That's only the **On/Off** switch. Connecting C-Console does not automatically arm the DS2416 to respond to MIDI messages. You'll have to configure MIDI controller assignments before a MIDI device can actually do anything with the DS2416 from a program or device other than C-Console.

What can't be remotely controlled by C-Console ?

The only features that can't be automated are bus routings (bus on/off switches) and effects parameters with "detents". Any parameters whose values don't increment or decrement smoothly can't be automated by C-Console.

All effects parameters which can't be automated are marked as such in the effects parameter guide.

Software MIDI drivers

If you don't see any *software* MIDI devices in either or both lists on the **MIDI Sync Setup** dialog (under **Sync|Remote),** you will need to install internal "loopback" drivers, or MIDI device emulators. You don't need any hardware on your system with MIDI ports to use these drivers. These drivers are internal "patch cables" which allow applications to send and receive MIDI data within the PC.

Hubi's MIDI Loopback Device

There are many loopback drivers available for download on the Internet, and most of them are free software. Our favorite "loopback" driver is Hubi's MIDI Loopback Device, a tried-and-true piece of free software which has been a favorite of musicians since Windows 3.1 days. There are several versions of Hubi's available, including both 16 and 32 bit versions of the drivers. You'll find a download link for the latest 32 bit version driver on the C-Mexx Internet support page for C-Console.

Hubi's MIDI Loopback Device is neither published or supported by C-Mexx. This information is provided as a convenience. No additional support will be provided by C-Mexx for installation or use of loopback devices, virtual MIDI drivers or "patch cable" drivers.

You are not limited to Hubi's as a MIDI "patch cable" driver if you need something like this on your system. You can use MIDI ThruWay, MIDI Yoke, Sonic Foundry's Virtual MIDI Router, or any other similar driver *provided* you select a 32 bit driver. In most cases you should avoid using more than one such driver on your system, but you *can* use multiple software MIDI ports with the same driver if the driver permits it. These packages all tend to be quite stable and efficient on most systems. Check the documentation for any loopback device you

may use; not all are 32-bit compliant, and a release date prior to 1996 almost guarantees that it won't be fully Win95/Win98 compatible.

Installation tips for Hubi's MIDI Loopback Device

After downloading and unpacking the software, copy the unpacked files to a destination directory and run the supplied **.inf** file in the unpacked archive to install the driver software. The readme file supplied with Hubi's will give you additional help if you run into installation difficulties.

You may be asked during installation how many ports you wish to use. If you are not asked, then you will have to choose for yourself from the list at the left of the setup dialog which shows **LB 1** through **LB 4.** For basic use, two ports is fine but most serious MIDI users find four is not excessive. Don't select more than you think you'll need though, since there may be a bit of a "performance hit" with large numbers of ports.

When we installed Hubi's, we double-clicked and renamed **LB 1** and **LB 2** to **Hubi 1** and **Hubi 2** so we can tell at a glance what these ports connect to when we route MIDI signals from other applications.

Once this driver is installed, restart Windows, and restart C-Console. Select **Sync|Connection** from the menu bar to open the **MIDI Sync Setup** dialog. You should now see one or more loopback entries to choose from, and you can patch C-Console to other applications by selecting the same device for both C-Console and the application to which you need to sync.

Troubleshooting the installation

If you're installing to a fresh system or partition designed specifically for use with audio applications, you might need to locate a file called **CTL3DV2.DLL** from another Windows installation and copy that file to the **\System** subdirectory of your "audio-only" Windows directory. Hubi's needs this file to display its port selector dialog.

Conflicts have been reported between Hubi's Loopback and Creative Labs MIDI-capable cards used under Windows 98. These problems may be solvable by using an alternate software solution. If you encounter this problem, try MultiMID at the URL listed in the Hubi's Loopback README file, or the alternate loopback devices listed above, to see if they can overcome this limitation.

The importance of multi-client MIDI capability

Some MIDI cards are multi-client capable. Most are not. Creative Labs cards in particular have been notorious for having only single-client capability. You need multi-client capability in some form if you are using both hardware and software MIDI devices and your primary MIDI card is a Creative Labs card. Many "SB clones" are also limited to single-client capability. Correct use of devices like Hubi's Loopback allow you to make a single-client card behave "as if" it is multi-client.

The README file for Hubi's MIDI Loopback Device offers additional explanation of multi-client capability and its importance.

Configuring MIDI remote control

You can configure MIDI control parameters without having the DS2416 connected to other MIDI devices, either in hardware or software. But before C-Console can be configured as a MIDI controller interface for the DS2416, you'll need to connect the DS2416 for MIDI remote control.

The MIDI Remote Setup dialog

Inpatchmap

As you can probably guess, this is the input mapper for MIDI messages. This tab configures C-Console responses to MIDI controller messages it receives from other MIDI devices (either hardware devices connected to your PC, or messages sent by software which uses your system's MIDI drivers).

Outpatchmap

Naturally this is the output mapper for MIDI messages. It determines what MIDI messages will be sent from C-Console to external MIDI devices, or to the DSP Factory itself.

Connection Setup

This configures your available MIDI devices as "patch cables" for transmission of messages. The ability to select different input and output "cables" allows C-Console to route a whole different set of MIDI messages from its output from the messages it receives at the input. For example, you could have an external hardware sequencer controlling the DSP Factory by selecting your MIDI card's port as the **MIDI In Port,** and have the DSP Factory control a software synthesizer or software sampler by selecting a "loopback device" as the **MIDI Out Port.**

Channel Patching

The "patch bay" for MIDI signals. Note that each DS2416 channel, each bus and each auxiliary channel can send messages to, or receive messages from, a different source.

The Remote Sets dialog

Beginning with C-Console 1.3, opening the **MIDI Remote Setup** dialog also opens a smaller dialog called **Remote Sets.** This dialog will be empty when you first run the program because you won't have any remote presets to work with. When you begin to add user control sets to your session, they will appear in this dialog.

Using the Inpatchmap and Outpatchmap tabs

These dialogs look identical but they do not behave identically. The difference is that the **Inpatchmap** dialog configures DS2416 responses to controller messages it receives, while the **Outpatchmap** dialog configures how the DS2416 will control external MIDI devices using messages that it sends. Put another way, **Inpatchmap** controllers allow your external MIDI devices to control the DS2416, while **Outpatchmap** controllers allow your DS2416, via C-Console, to control your external MIDI devices.
Another important difference is the **iThru** feature in the outmap.(see Assigning new controllers)

The top button bar

The first, third and fourth buttons should be self-explanatory. **Autocreate** opens a dialog that allows you to automatically generate up to 128 controllers whose parameters are based on the currently selected controller.

Understanding the Autocreate dialog

Number of copies specifies, of course, the number of copies you want to make of the selected user control.

Increase Channel increments the DSP Factory or remote channel by one with each copy of the control so, for example, creating eight controls starting with **CH 5** creates new controls preassigned to channels 6, 7, 8, 9, 10, 11, 12 and 13.

Increase MIDI Channel increments the MIDI channel by one so that each new control transmits **(Outpatchmap)** or receives **(Inpatchmap)** on the next highest channel. For example, creating five copies of a control that uses MIDI channel 1 will create user controls preassigned to send or receive on channels 2, 3, 4, 5 and 6. MIDI channel and DS2416/Remote channel can be incremented simultaneously.

The **Check positions to increase:** field can be very confusing. Normally the only "position" you can increment with new copies is the *controller number,* or the value of the controller under the **CTL** column. The other two checkboxes in this list are reserved for system exclusive values (see below).

The Autocreate function: basic functions

Here's an example of how to use the **Autocreate** function to rapidly duplicate a set of controls.

1. Select **Control Changes>Ctl 6 (Data Entry)** under the **CTL** column for your template parameter in the **Inpatchmap** tab, perhaps to use a hardware fader as the controller.
2. Select **Yamaha DS2416>CH1** from the **CHANNEL** column.
3. Select **Fader** from the **PARAMETER** column.

4. Select **1** from the **MIDI CH** column. You have now set up an external hardware fader to control C-Console's channel 1 fader.
5. Open the **Autocreate** dialog and specify **15** under **Number of copies:.** Check **Increase Channel** (to increment the C-Console mixer channel) and **Increase MIDI Channel** (to increment the transmission channel) and press **OK.**

You now have 16 controllers assigned to the same remote fader. Each time you increment or decrement the MIDI send channel on the hardware controller, that controller's fader can manipulate a different DS2416 channel fader in C-Console.

The Autocreate function: advanced functions

The **Autocreate** function can also be used to increment MIDI system exclusive values when used in conjunction with a specific sysex message. Here's how to implement this function.

1. Select **System Exclusive...** as the controller type under the **CTL** column. This opens the **Edit Sysex Message** dialog.
2. Specify a manufacturer ID value (Realtime and Non-realtime are selectable from the bottom of the list.)
3. Enter the string representing the sysex message. (Because this is an advanced function, this will not be further described here.) The reserved word **VALUE** (all caps) is used as the placeholder for the value the newly-created user control(s) will increment or decrement. In other words, the byte represented by **VALUE** is the data that the new user control will actually modify.
4. Close the **Edit Sysex Message** dialog.
5. With the user control still highlighted, press **Autocreate.** You will notice that the number of available checkboxes in the dialog has changed. Depending on the number of bytes in the sysex message, any number of checkboxes could be available. It is up to you to know which byte values shown in this checklist correspond to which controller types and values for the MIDI device you want to control.
6. Check the desired controller or controllers in the listbox after specifying the number of controllers to autocreate and close the dialog.

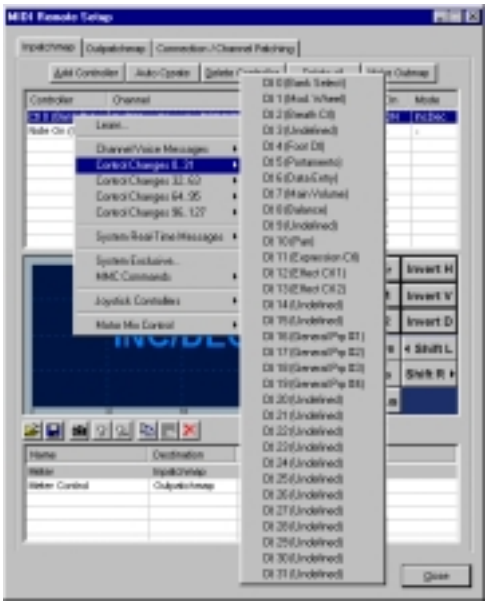
You now have multiple user controls incrementing through the specified range. Depending on which byte(s) you chose to increment, you can produce useful configurations such as an array of eight pots or faders on a remote device, each assigned to the controller whose identifying byte you checked in the checkbox list, each controller capable of manipulating the same data type represented by **VALUE** in the sysex string you entered.

This is an advanced subject but not terribly difficult to understand once you have a basic grasp of the structure of sysex data strings. This dialog offers you an excellent opportunity to learn about sysex strings and how to use them in mixing and recording, since you can get rapid feedback on your experiments and there's negligible risk of causing any corruption to an existing session.

The “make Out/Inmap” button

If a midi controller or sequencer is used as a in AND out device, it would be necessary to copy every patameter of the inmap to the outmap including a mirroring of the graphics curve. This is exactly what this feature does...

Assigning new controllers



Before you can take advantage of any MIDI control, you need to set up one or more controllers to work with - or through - C-Console. Press the **Add Controller** button to automatically create a new, unpatched controller to work with.

Once you have created your first controller, you can use **Autocreate** to "clone" that controller one or more times to set up a bank of preinitialized controllers for editing. **(Autocreate** has no effect until at least once controller has been defined.)

Modifying controller parameters

Here's where you turn the initialized controllers into useful devices. Until the controllers are assigned to particular parameters, they don't have any function or purpose.

There are two "secrets" to modifying controllers. The first secret is that right-clicking over any of the columns in the controllers chart displays a pop-up menu of options. The second secret is that the channel options you select determine what parameters you can control. The right-click **PARAMETER** menu changes to suit different types of **CHANNEL** selection. We encourage you to experiment with all the channel types to see which parameters are available for that selection group. The sample tasks for creating new user controls will show you many of the features of these dialogs.

CTL column: Right-clicking over this column calls up a pop-up menu of all available controllers. This menu is divided into several sections.

Learn... only appears in the **Inpatchmap** dialog, and allows a remote device to identify itself to C-Console. If MIDI channels and connections are set properly when you select this menu item, modifying a hardware or software MIDI control in another device will send MIDI signals to C-Console and tell C-Console which type of message you want this controller to respond to.

The next submenu consists of all standard MIDI voice messages and all 127 MIDI message assignments.

The **System Real-Time Messages** submenu is for controlling the C-Console transport...starting, stopping and continuing automation sequences. Start/stop/continue messages can control external devices or be triggered by external devices to activate the C-Console automation transport.

System Exclusive... is an advanced feature for sending or receiving hexadecimal sysex data. This feature requires an understanding of sysex hexadecimal code and will not be further explained here. See **Advanced use of the Autocreate dialog** above for a brief overview on this option's capabilities.

The **MMC Commands** submenu is for sending or receiving MIDI Machine Control messages. These messages can be used to control an MMC device's transport from C-Console or have the transport trigger another device using C-Console as the interpreter.

The **Joystick Controllers** submenu allows for configuration of joysticks as MIDI controllers. This is highly sophisticated capability and allows for the use of virtually every modern joystick control, including hat controls, as a trigger for a C-Console user control. Select **Joystick Info...** to learn about the capabilities of your currently installed joystick.

The **Motor Mix** submenu offers full interfacing capabilities with a CM AUTOmation™ Motor Mix™ console, mirroring virtually every control and parameter of the Motor Mix. It allows you to have either the Motor Mix control devices via C-Console, and/or have other devices (or C-Console itself) control Motor Mix.

Channels 1 to 8 stand for Motormixes Channel-strips with their parameters Fader, Fader-touch, Select, Mute, Solo, Multi, Rec/Ready, Rotary.

Left side/Right side covers the left/right button side of the motormix buttons.

global covers global parameters.

bargraph styles lets you choose between different styles in Motormix´s display.

CHANNEL column: There are three submenu options for **CHANNEL** selection. These correspond to internal channels of C-Console covering hardware or virtual user control channels , not MIDI channels.

The **Active Channel** menu item is available only from the **Inpatchmap** tab. **This is a dynamic selection which is not hard-wired.** When selected, C-Console autoselects the channel currently selected. This is a very useful feature which can, among other things, allow you to use an external dial or fader to, for example, set the levels of all channel faders in C-Console from the same external hardware fader. This would allow you to set channel 1's fader by clicking on the channel 1 fader and then moving the hardware fader, then selecting channel 2's fader and moving the hardware fader again.

The Hardware I/O (e.g.**Yamaha DS2416,Roland VS)** submenu allows you to select every available routing in the Hardware as the channel to send on (when selected from **Outpatchmap)** or receive on (when selected from **Inpatchmap).**

The **MIDI Remote** submenu allows for selection of any of the 14 available MIDI remote channels used by C-Console. **Active Channel** refers to the channel currently selected in C-Console. Note that MIDI remote channels do not correspond to MIDI channels, which is why there are 14 and not 16 or 32. They are used internally by C-Console; they only tell C-Console how to route the MIDI messages, not which channel to send or receive on. The actual send/receive channel is selected from the **MIDI CH** column.

The **PARAMETER** menu selections change dynamically depending upon the **CHANNEL** selection you made.

If you selected **Active Channel** as the **CHANNEL,** which you can only do from **Inpatchmap** tab (controlling C-Console from an external device) then the last parameter control selected (clicked) in C-Console becomes the parameter which your remote device controls. This is a very handy option for using a single remote fader to rapidly modify a number of controls in any dialog, since you can move the fader with one hand while clicking controls with the other.

If you selected a hardware e.g.**Yamaha DS2416** or **VS 1680** channel as the **CHANNEL,** then the available options under **PARAMETER** depend on which channel you chose. This menu is designed specifically to show you only those parameters which are controllable from the selected channel, so if you selected **BUS4,** you'll see only the parameters which apply to **BUS4.** If you select a mixer channel 1-24, you get access to all main mixing strip, EQ, dynamics parameters, and aux/bus send levels for that channel.

If you select one of the effectors, you see only the parameters that apply to that effect.

If you selected a remote channel as the **CHANNEL,** then the **PARAMETER** menu allows you to assign any of 24 available **User Variables,** seven **User Switches** or **Stereo Mode** as the parameter.

The **MIDI CH** column's right-click menu allows you to select from any of 16 available MIDI channels for sending or receiving MIDI messages.

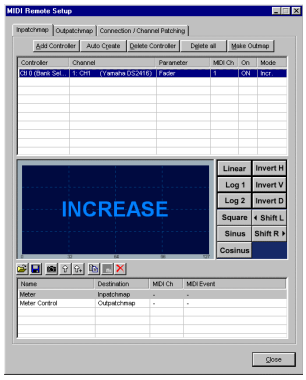
The **ON** column simply allows you to turn the specified controller on or off for a given job.

The **MODE** column is available only on the **Inpatchmap** tab, and sets the value change type. This powerful control allows you to determine how MIDI value changes sent by the sending device or control will be processed for routing to the receiving device or control. Clicking and right-clicking in this area has the same effect: it cycles through the available value-change modes.

NORMAL is the default, and it gives you access to all of the available parameters in the response curve editor.

Click once and you'll see **INC/DEC** appear in the response curve display area. When **INC/DEC** is selected, the sending control acts as a center-detented control and increases or decreases the specified receiving control. The behavior of Inc/Dec controllers is very specific. Whenever the sending control is activated, it behaves as though it has been reinitialized to 64, which is the center of the 0..127 MIDI value scale, regardless of where the control's actual position might be, and it moves the receiving control up or down from a value of 64 by the exact amount it is moved in the sending control.

Click again and you'll see **DECREASE** in the response curve area; click again and you'll see **INCREASE.** Decrementing and incrementing controls behave very differently from the Inc/Dec control type. They don't initialize the receiving control's value to 0, 64 or 127 prior to increasing or decreasing its value. Instead they increase or decrease the value of the receiving control by the exact amount that the value was increased or decreased by the sending control. Incrementing and decrementing controllers are often switches or pushbuttons, which increase or decrease the receiving control by a value of 1 or a predetermined amount specified from the sending controller.



Click again and you'll see **TOGGLE** appear under the **MODE** column and a new display in the response curve editor. Toggle mode is for use with switch-type controls, and can function in one of two ways depending on the type of parameter you wish to control. In default mode, it acts as a simple on/off switch. Activate the switch once and a value of 0 is sent; activate again and a MIDI value of 100% is sent.

Toggle mode can also increment and decrement a parameter by a user-defined amount. The response curve editor becomes a three-part switch editor dialog when this option is selected. Click the left column and drag the column up or down to increase the lower of the two values used by the switch; click the right column and drag it down to decrease the upper value. The split point, or MIDI value at which the switch is activated, can be user-defined by right-clicking over the small bar in the center and dragging it left and right.

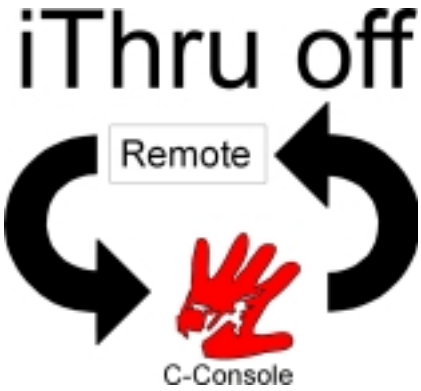
iThru (intelligent through) is a very advanced feature of the remote editor inpatchmap. Normally it is always switched to off.

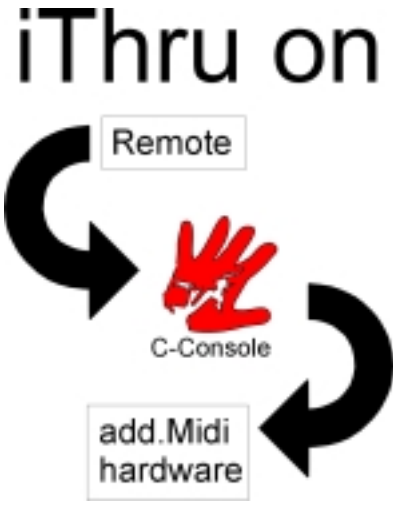
iThru ***off*** prevents your system from a midi loop that would freeze C-Console. If a midi-event is received from outside C-Console controlling a parameter of C-Console, and this parameter is also assigned to the outpatchmap iThru blocks this parameters Midi out message.
This is needed to use a midi-remote with display or motorfaders. The fader normally sends out and receives the same midi controller. So it´s of the essence, that C-Console controls the motor fader, but the fader may not control itself. (Loop!!)

IThru ***on*** is needed if C-Console should be a midi mapper, or midi is transmitted through C-Lan/C-Wan plugin. It is very important that the same midi controller does not appear in the in and outmap at the same time...

CAUTION!
YOU HAVE TO BE ABSOLUTELY POSITIVE THAT NO LOOP WILL OCCUR, HAVING SWITCHED iThru to *on*. IF A MIDI LOOP OCCURS C-CONSOLE COULD FREEZE.

If a midi remote controls a C-Console controlling another device through midi iThru has to be set to ***on***.





The bottom button bar

This button bar acts as the preset manager for working with controller sets.

Pops up a **File|Open** dialog with special characteristics. This dialog doesn't actually open whole snapshots or session files, but only imports the parts of the session file that you select. For example, if you have a session or snapshot with your default settings to allow your main MIDI keyboard controller to control C-Console, you can import the user controls (patch maps), libraries (effects, EQ settings), or both into the current session.

This functions similarly to the **Save As...** menu item, allowing you to save your current session from the **MIDI Remote Setup** dialog. It saves the *whole session,* not just the user control presets. You can do the same thing from the main menu, so this is mainly here as a visual reminder to follow your banker's advice: save early and save often.

The camera button clones your current list of user controls to a new preset and assigns it the default name **new entry.** The clone is, of course, an exact copy of the original list with one exception: the clone user control list prefers café mocha, while the original user control list always drinks cappuccino. Unfortunately no fix is possible for this...cloning is still an imperfect science.

This button has the same effect as double-clicking a user control list from the listbox below...it overwrites the current user controls list with the currently selected controls list.

In cases where you want to take an existing list of user controls and add to it, this button inserts all controls in the selected list below the last item in the current list.

These buttons are primarily for use when you have two or more copies of C-Console open at the same time, or when you want to take a controls list from one session file and insert it into another file. Click the left button to copy the selected user controls list to the clipboard. Then, either open a new session from the main menu bar and press the right-hand paste button to paste it into the newly opened list of presets, or switch to a different instance of C-Console and paste it into the **MIDI Remote Setup** dialog of that instance.

This deletes the currently selected list of user controls.

The presets list

The grid at the bottom of the dialog is a list of available presets. This list will be empty if you have not created any presets. (See **Creating a new preset** below.) A "preset" is a set of user controls.

Name is the user-defined title for the preset. Right-click over this area to modify the preset name.

Destination specifies the tab on which this preset is used. Click this area to toggle between **Inpatchmap** and **Outpatchmap.**

The **Destination** field does *not* determine where the preset goes except when MIDI preset loading is used (see below). The selected preset is automatically loaded into the currently selected tab. This field is intended only as a personal reference to remind you of the tab for which the preset is intended.

MIDI Ch and **Prg Chg** fields are for remote-control loading and unloading of presets. This dialog will respond to a MIDI program change message sent on the specified channel and when that message is received, the user controls stored in the preset assigned to that program change message will instantly overwrite the current list of user controls, offering on-the-fly modification of the entire user control configuration.

Short Xplanation

The controller assignment chart

Once you understand how the right-click menus in the **CTL, CHANNEL** and **PARAMETER** columns behave, working with the controller assignment chart should be relatively straightforward. Right-clicking in the **CTL** column provides you with access to the entire

Of special note is the **Learn** option under the **CTL** column. When this is selected, C-Console waits for you to modify a MIDI controller and autodetects the controller type. This is especially handy for use with external MIDI control surfaces. All you need to do is select this option, then press a key or button or move a knob or fader, and C-Console will figure out which MIDI controller that switch or fader is currently assigned to.

Note that identical controllers can be assigned to the same channel and parameter. This is a deliberate design decision that opens up additional flexibility in controlling the flow of MIDI messages, but it could create unpredictable responses with some devices. Until you get a strong feel for working with remote MIDI control, try to use unique controller parameters with every controller.

The response curve editor: basic functions

This area of the dialog allows you to custom-tailor the way a particular MIDI message translates to a particular DS2416 parameter or MIDI parameter. Caution is *definitely* recommended here...users have commonly caught themselves having a bit *too* much fun playing with response curves at the expense of productive work.

The *x* axis (horizontal) specifies the value of the MIDI message from 0 to 127, while the *y* axis specifies the percentage of a particular parameter to apply in percent. The display starts with a default linear response which will be fine for most values. Dragging the small arrows, or "handles", to the left and right of the display allows you to define start and end points for a linear response curve.

To the right you have a selection of preset response curves. At the bottom are the **>** and **<** buttons, which shift the entire curve across the *x*-axis.

Aside from the preset linear curve and the "square" curve, the two logarithmic curves will likely be your most-used curve types. What isn't provided is an inverse log curve. The **INV H** and **INV V** buttons perform vertical and horizontal inversion on the curves but some users find that they don't behave intuitively. You may need to try a combination of clicks on these two buttons before you see an appropriate inverse log curve.

The response curve editor: advanced functions

If you click and hold on the dividing line on the slope, you'll notice that you can draw controller curves with the mouse such as the one below. The curve starts at the left arrow or handle and ends at the right arrow, and for the technically-minded, acts as a Bezier drawing function with only one point. Once you become familiar with MIDI remote control of the DS2416, you'll likely find this particular feature to be an uncut gem. Oddball controller curves can be used to generate *very* precise expressive responses.

Manual response curve editing

Right-clicking over the display area shows two menu items. **Edit Handles** allows you to manually enter the values from 0 to 100 for the left and right arrows/handles. You are not limited to whole numbers here; you can enter decimal values (e.g. 33.3333 or 12.5) if you need this much precision.

Edit Values opens a dialog that gives you complete manual editing access to all 127 MIDI messages. This is an advanced feature designed for *very* precise expressive control. The **Auto Fill** feature allows you to select and set a value for any single MIDI message value and have the corresponding percent values automatically calculated for the rest of the MIDI messages. We urge you to experiment with it by entering a whole-number value for a particular MIDI message value to see how it behaves. It's much easier to use than it is to explain.

The default values in the right column for the 127 MIDI message values in the left column represent a handy reference to the correspondence in remote control between 0-100% and 0-127.

Create a handy "sample/hold" template controller by turning off **NumLock** on your keyboard and randomly punching in numbers from the keypad for each of the 127 values. When you move an assigned control, it will cycle the 127 random values creating a sample/hold effect.

C-Mexx will not be responsible for losses incurred from addictive use of the user control features in C-Console. Some users report that Quake might be *less* addictive than playing with this dialog. Cold-turkey withdrawal from these features is not recommended. Depression may result from use of interfaces which do not include comparable remote control capabilities.

Managing user control presets

Creating and saving a new preset

1. Load the session file where you want the preset stored. This can be a new session, an existing session used only for preset storage, or a work session. You must load the target session in advance because you can't save presets separately. They are always saved as part of a session.
2. If you want to work from an existing preset, select a preset into the **Inpatchmap** or **Outpatchmap** tab from the list at the bottom of the dialog. If you want to start with a new, initialized preset, then clear all user controls from the current tab. Hint: Highlight the topmost control and hold down the **Delete** key to rapidly remove a long list of controls.
3. Create one or more user controls.
4. Press the button to copy the current user control list into the preset list.
5. Right-click over the preset **Name** to assign a new name to it. Insure that the **Destination** tab for this preset is properly identified, even if you don't intend to load this preset remotely, so you will know where it goes when you reload it.
6. Use the save button on the button bar, or **File|Save** or **File|Save as...** from the main menu to save the session.

A single preset contains all controllers shown on the currently selected tab (either **Inpatchmap** or **Outpatchmap).** If you want to save both **Inpatchmap** and **Outpatchmap** controllers, you must save two separate lists of user controls.

Loading (importing or merging) presets from other sessions

1. Open the session in which you want to use an imported preset. Open the **MIDI Remote Setup** dialog. It does not matter whether the **Inpatchmap** or **Outpatchmap** tab is visible.
2. Use the button to load preset data from another session. You will be given the choice to merge any library files contained in these sessions with the libraries currently loaded into C-Console from **.clb** files or as part of a session file.
3. The merge operation will add any presets in the selected session **(.css)** file to the current list. This may result in an unmanageable selection of presets, or conflicting presets (i.e. presets with the same channel/program change assignments), so you may want to delete any unwanted presets from the bottom list.
4. Use the save button on the button bar, or **File|Save** or **File|Save as...** from the main menu to save the session.

Using the Connection Setup tab

This simple dialog allows you to select the default MIDI devices for sending and receiving data. As long as you have MIDI drivers installed and correctly configured either for internal or external devices, there will be at least one MIDI device for both **MIDI In** and **MIDI Out.**

The **MIDI In** device you select is the device that sends MIDI controller messages to C-Console for use by the DS2416.

The **MIDI Out** device you select will be the device that receives MIDI controller messages

Using the Channel Patching tab

This dialog might not be instantly obvious, but it is very easy to use. It sets the internal C-Console routings as they will be used for MIDI remote control, and allows you to override the default channel assignments if alternate assignments make the mixer more intuitive or easier to work with.

Select a channel from the **Remote Channels** column and select a corresponding entry from the **Select** category. The channel lists are stacked to show channels for the first card in the top half of the list; channels for a second DS2416 card are shown at the bottom.

The **Patch to** column displays the current patching relationship. **Unpatch All** removes all patches; **Set Patch** assigns default patchings to all channels (1:1 correspondence for default routings; no patching for optional routings).