

Contents

Introduction

[Registration](#)

[About Midi Quest](#)

[Working with SysX data in Midi Quest](#)

[Manual Organization](#)

[Getting Support](#)

Installation & Setup

[Installing Midi Quest / Solo Quest](#)

[Downloading New Instruments](#)

[Running Midi Quest / Solo Quest](#)

[Connecting Your Instruments](#)

[Setting Up Your Instruments](#)

Tutorial

[Part 1](#)

[Part 2](#)

[Part 3](#)

[Part 4](#)

[Part 5](#)

Editor Windows

[Group Window](#)

[Bank Editor Window](#)

[Patch Editor Window](#)

[Library Window](#)

[DBase Window](#)

[Driver List Window](#)

Other

[Cakewalk Name Support](#)

[Midi Quest MFx Plugin](#)

[Midi Quest VST Plugin](#)

Utility Windows

[Midi Controller Window](#)

[Midi Monitor Window](#)

[File Conversion Window](#)

[Driver Creator Window](#)

[Driver Creator Technical Docs](#)

Menus

[Files Menu](#)

[Midi Menu](#)

[Options Menu](#)

[Utilities Menu](#)

[Windows Menu](#)

Toolbar

[Toolbar Commands](#)

Procedures

Midi Communication Errors

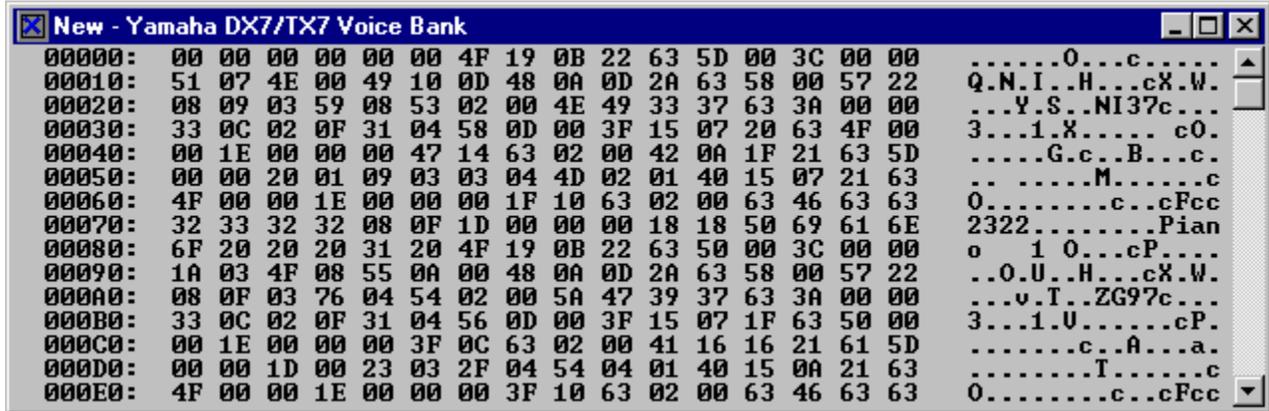
Drag and Drop between Windows

Drag/Drop from the File Manager

Keyboard Functions

Windows Key Strokes

SysX View Window



About the SysX View Window

The SysX View window is used to view data files in a raw hexadecimal and character format. The window is for display only purposes, there is no editing possible. It is used primarily for development and testing of new drivers and templates for Sound Quest programs. If you are developing new instrument support files this is a fast and easy way to view the data you are loading.

The first column shows the byte offset (in hex) of the first parameter of data for that line. For example in line 00060, the first byte is 96.

The second column is divided into sixteen sub columns. Each sub column is a data value so each row shows sixteen bytes of data from the data file. Row one shows bytes 0 ->15, row two shows bytes 16 -> 31, and so on. These values are the actual data as stored within the SysX dump. All characters are shown in black except for MIDI Status commands, these are shown in red. For example, you can see the the F0 at the beginning of the dump in the graph above is displayed in red.

The third column shows the same sixteen bytes in ASCII character format. If the value of the byte is an ASCII letter or number, it is displayed as such, otherwise, a period is displayed in its place.

The SysX data can be edited click and hold the mouse down on any of the number in the center section. Drag the mouse left or right to decrease or increase the value of the byte.

The SysX view window uses color to indicate particular meanings for various bytes

- White - the currently selected byte
- Red - the byte has been edited
- Blue - the byte is part of the header
- Yellow - start or end of SysX bytes

SysX View Menu

- Advance Voice Size advances the data display forward by the size of one voice in Bank SysX data
- Backup Voice Size backup the data display backward by the size of one voice in Bank SysX data
- Midi Parser show the individual SysX messages in the data file
- Update Settings... edit the data's basic parameters, see Update Settings

Windows Keys

Cursor Movement Keys

Dialog Box Keys

Editing Keys

Help Keys

Menu Keys

System Keys

Text Selection Keys

Window Keys

Open SysX View (Bank)

Purpose:

Open SysX View moves the contents of the Bank into the SysX View Window. In the SysX View window the data can be viewed in its raw hexadecimal form.

This feature is useful for viewing data while developing and testing instrument drivers in MIDI QUEST or MIDI QUEST Jr.

Directions:

Choose *Bank Edit/Edit/Open SysX View*.

See also: [SysX View Window](#)

See Also: [Bank Editor Window](#)

Patch Name (Bank)



Purpose:

Patch Name displays a dialog to rename a Patch in a Bank without opening a Patch Editor.

You may wish to rename Patches this way when:

- (1) Only the name of the Patch needs to be changed, its faster this way
- (2) There is no Patch Editor for the data, so this is the only way to change the name
- (3) The name is not a component of the Patch Editor so again, this is the only way to change the name

The last option applies only to some older instruments where Patch names were not included as a component of the Patch. In such cases, the program will create space so that names can be assigned and stored. Remember that these names cannot be sent back to the instrument. The instrument does not know how to store them.

Directions:

To name a Patch in a Bank:

- (1) Select one or more Patches you wish to name.
- (2) Choose the Name button or *Bank Edit/Edit/Patch Name...*
- (3) Use the displayed Patch Name Dialog to enter a new name
- (4) Press the OK button
- (5) If more than one patch was selected, the dialog will appear for each patch until each has been renamed. If audition is enabled, the patch will be auditioned before the name appears for editing.

See Also: [Bank Editor Window](#)

Mixing and Blending (Bank)



Purpose:

There are five different ways to create new Patches in the Bank Editor: Mix, Mix All, Blend, Gen4, and Morph. The three discussed here are Mix, Mix All, and Blend. In each case, an entirely new Bank of Patches is created from the selected Patches. You can quickly try out these new Patches in your music by auditioning them. You don't need to know a thing about programming your instrument.

The following is an explanation of how the Mix, Mix All and Blend functions work:

Mix creates a new Bank of Patches by randomly picking parameters from two selected Patches. The first Patch created is largely biased toward the first selected Patch. As new Patches are created, the bias increases toward the second selected Patch. The last Patch created is largely biased toward the second selected Patch. This produces the aural effect of gradually moving from the first Patch through to the second Patch. Only values actually present in either of the two Patches are used.

Mix All creates a new Bank of Patches by randomly picking parameters from all selected Patches. The probability of the parameter coming from each of the selected Patches is equal. Therefore, if 2 Patches are selected, the chance is 50% that the parameter value will come from either Patch. If four Patches are selected, there is a 25% chance that the parameter will be selected from each Patch, and so on. Each new patch is a random combination of the selected Patches.

Blend works exactly the same as Mix except that it combines larger chunks of data from each Patch thus maintaining more of their overall aural characteristics.

Directions:

To Mix or Blend Patches:

- (1) Select a Patch
- (2) Hold down the CNTRL key and select a second Patch
- (3) Choose *Bank Edit/Randomize/Mix* to Mix the 2 Patches or
Choose the Blend button or *Bank Edit/Randomize/Blend* to Blend the 2 Patches

To Mix All Patches

- (1) Select two or more Patches
- (2) Choose the Mix button or *Bank Edit/Randomize/Mix All* to Mix the Patches

Example:

To 'Mix' or 'Blend' Patches 7 and 12

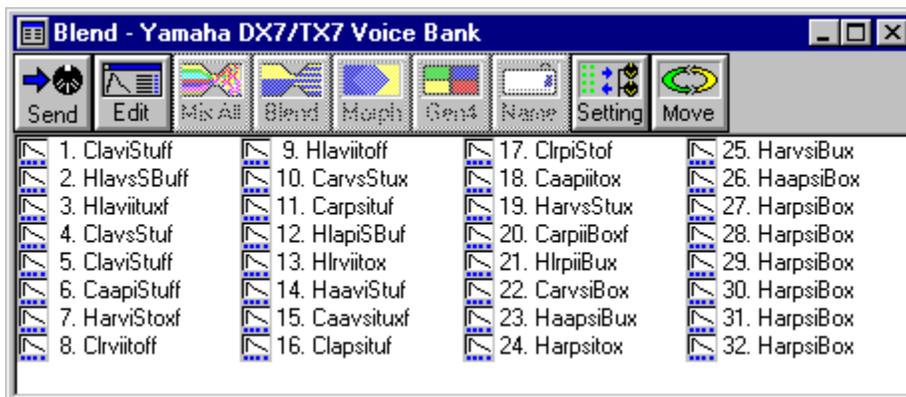
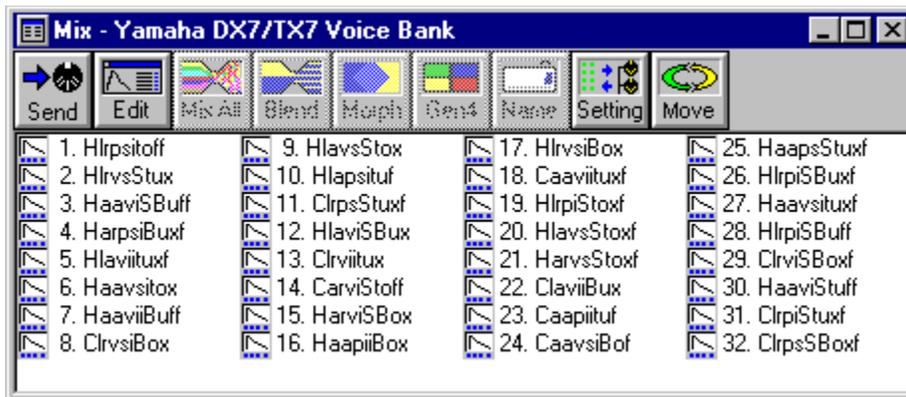
- (1) Click the mouse on Patch 7

(2) Hold the CNTRL key and click on Patch 12



(3) Choose *Bank Edit/Randomize/Mix*, or *Bank Edit/Randomize/Blend*

(4) The results would be the two following Banks, a Mix Bank and a Blend Bank



To Mix All' Patches:

- (1) Select the Patches you wish to mix
- (2) Choose the Mix All button or Bank Edit/Mix All



'Mix All' randomly picks parameters from ALL selected Patches

See Also: [Bank Editor Window](#)

Restore (Bank)

Purpose:

Restore returns the selected Patches to their pre-edited state.

This feature completely undoes all edits performed on one or more selected Patches in a Bank. This includes any changes in Patch organization or individual Patch edits in a Patch editor. The Bank Editor accomplishes this by storing a copy of the Bank data when the Bank Editor is first opened. When Restore is chosen, the program replaces the currently selected Patches with those from the backup Bank.

Note: To access Restore, *Options/Keep Backup* must be checked when the Bank is first opened

Directions:

To return the selected Patches to their pre-edited state:

- (1) Select the Patch(es) you wish to restore
- (2) Choose *Bank Edit/Restore*

See Also: [Bank Editor Window](#)

Invert (Bank)

Purpose:

Invert is used to invert the order of a range of Patches in a Bank. It operates on **all** of the Patches from the first selected Patch through to the last selected Patch. Unselected Patches between the selected Patches will also be inverted.

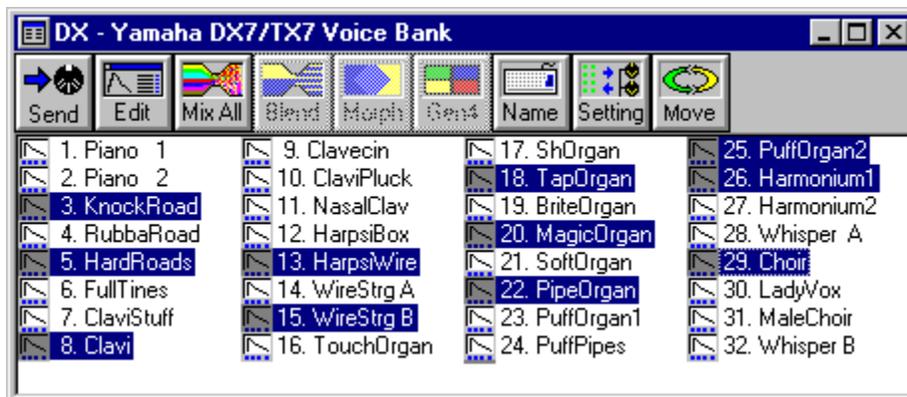
Directions:

To invert the order of Patches in a Bank:

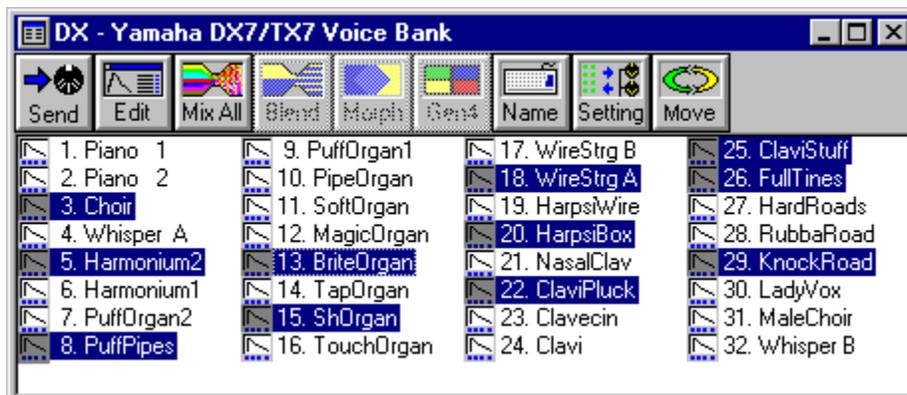
- (1) select a continuous RANGE of Patches
- (2) choose *Bank Edit/Organize/Invert*

For example:

In the first graphic, a number of Patches are selected prior to performing an invert:



The second graphic shows the Bank after an invert. Notice that the locations of all of the Patches between 3 and 29 have been inverted regardless of whether they were selected.



See Also: [Bank Editor Window](#)

Rotate (Bank)

Purpose:

Rotate shifts a range of selected Patches either up or down one location in the Bank. In this way, space for a new Patch may be created in the middle of a Bank without the subsequent order of Patches in the Bank being changed.

Note: All Patches from the first through last selected Patch are rotated. This includes any unselected Patches located in between the selected Patches.

Directions:

To rotate Patches:

- (1) Select a continuous RANGE of Patches
- (2) Choose either *Bank Edit/Organize/Rotate Range Up*, or
- (3) Choose *Bank Edit/Organize/Rotate Range Down*

Example:

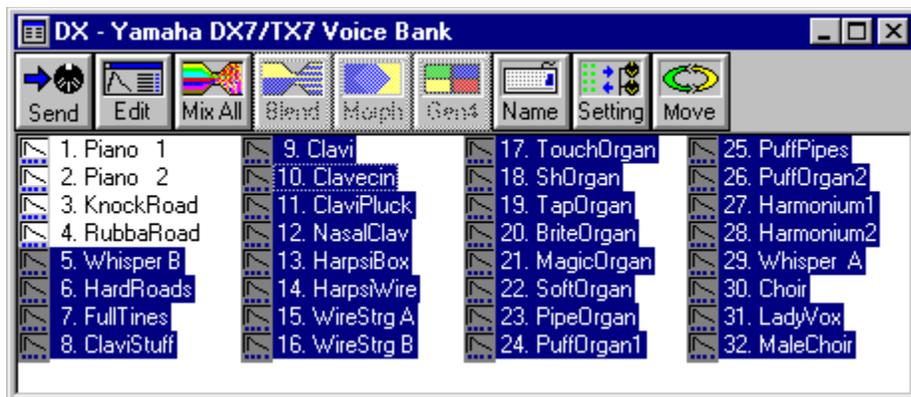
To make room for a new Patch in location 4 by rotating Patches 4 - 32 down 1:

- (1) Click the mouse on Patch 4
- (2) Hold the mouse button down and drag the mouse to Patch 32
- (3) Release the mouse button
- (4) Choose *Bank Edit/Organize/Rotate Range Down*

Before rotating down:



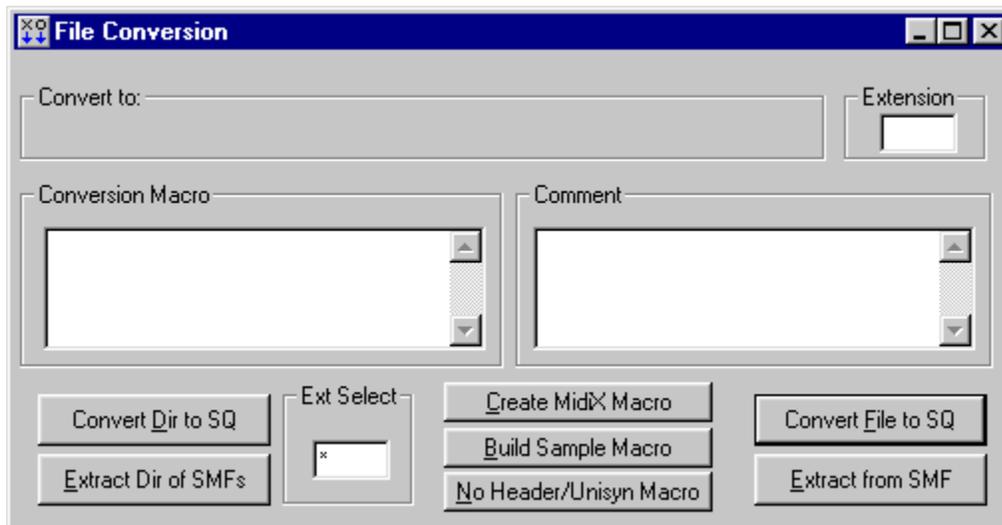
After rotating:



See also: [Bank Editor Window](#)

File Conversion Window (Utilities Menu)

Converting MidiX Files



Purpose:

The File Conversion Window (for MidiX files) is used to convert MidiX files into Sound Quest file format. If the files are in this format then the software is capable of creating a macro which can convert from the MidiX format into Sound Quest's format.

Directions:

The steps to converting a file are as follows:

1. Open the File Conversion Window.
2. Load the Sound Quest Driver that matches the type of data in the MidiX file.
3. Press the "Create MIDIX Macro" button to create the necessary macro.
4. Select "Convert File to SQ" or "Convert Directory to SQ".
5. Load the new data file to ensure proper conversion.
6. "Save" the new conversion macro to disk for later use.

NOTE: With an existing conversion macro replace 2 and 3 with 7.

7. "Open" an existing file conversion macro.

Step 1. Opening the File Conversion Window.

Use the following steps to access the File Conversion Window:

1. Bring down the "Utilities" menu.
2. Select "File Conversion Window" (item 8).

Step 2. Loading the Sound Quest Driver

The Sound Quest Driver must be loaded into the File Conversion Window because this information is an integral component of every Sound Quest Data File. It also tells the File Conversion Window what kind of data is being converted. A driver can be obtained from two locations, directly from disk or from the Driver List Window. This driver **MUST** represent the data stored in the MIDIX file or the program will most likely crash while performing conversions!

Loading the Driver from disk:

1. Select on the File Conversion Window to activate it.
2. Display the Convert menu.
3. Select the "Load Driver From Disk" item.
4. Use the displayed File Selector to select the driver.
5. Select OK to load the driver.
6. The driver name will appear below "Convert to:".

Loading the Driver from the Driver List:

1. Display the Driver List Window (if necessary).
2. Select the Driver to load into File Conversion.
3. Select on the File Conversion Window to activate it.
4. Display the Convert menu.
5. Select the "Load Driver From List" option.
6. The selected driver will appear below "Convert to:".

Step 3 - Create the Macro

Press the "Create MIDIX Macro" button to create the macro that will convert your files into that used by Sound Quest. The program will do this automatically meaning that you will require no knowledge of the MIDIX file format or Sound Quest's. The one exception to this general rule is that of Roland. Because their instruments do not dump SysX in a consistent fashion, in virtually all cases, it is necessary for you to write a conversion macro.

After the macro appears, you may type in a comment so that you can remember the type of conversion that this file performs.

Step 4. Convert the File

To convert the file, select either "Convert File to SQ" or "Convert Dir to SQ". The program takes the following steps:

1. Opens a File Selector
2. Choose the file to convert and Select "Ok"
3. A data file is allocated for the Conversion Driver
4. The selected disk file is opened and prepared.
5. The macro is executed.
6. The Sound Quest file is saved to disk with the extension currently in the "Extension" box in the window.
7. If directory processing Loop to #3 for all files.

If you wish to convert only those files with a particular extension, enter this extension in the "Ext Select" entry area. For instance, if you only wished to convert files with the ".DDF" extension, you would enter "*.DDF".

WARNING: If a directory of files is being converted, ALL of the files in that directory MUST be of the same type.

WARNING: If there are two or more files with the same name but different extensions, all of the converted files will be lost except the last one converted. The files should be renamed before the conversion process is started.

Step 5 - Load the New File to Verify

After the files have been converted, the new files should be loaded to ensure that the process proceeded properly. This is particularly important when you have just written your own conversion macro.

Take the following steps:

1. Bring down the "Windows" menu.
2. Select the "Edit Window" item.
3. A File Selector will appear.
4. Select the newly created data file and press "Ok".
5. Verify that the data has been converted properly.
6. Transmit the data to your instrument.
7. Verify the data aurally.

Congratulations, you have successfully converted file formats.

Step 6 - Save the Conversion Macro

Once the conversion macro has been tested, you can save it to disk for later use. When the macro is saved to disk, the Driver file in use is also saved with the macro and is automatically loaded with the macro. Before saving the file if you wish, you may add a comment in the comment box so that you may easily keep track of what this conversion macro does in the future.

Take the following steps to save the macro:

1. Activate the File Conversion Window (click mouse on it).
2. Display the "Files" menu.
3. Select the "Save As..." item.
4. Enter a name for the macro.
5. Select "OK" to save the macro.

Step 7 - Load a Conversion Macro

If you want to use a conversion macro you have written and saved or received from someone else, follow these steps to load it:

1. Select on the File Conversion Window to activate it.
2. Display the "Files" menu.
3. Select the "Open" item.
4. Use the File Selector to select the macro.

5. Press "OK" to load the macro.
6. MIDI QUEST will load the selected macro for use.

Note 1: the conversion macro file does not actually save the Sound Quest Driver it is using, only a path to where that driver is located on disk. If you obtain a conversion macro from someone else, make sure that you either have the requisite driver yourself or you obtain the driver as well.

Note 2: when converting a directory of MIDIX files with the 32 bit version of the software, the program will use only the 8.3 version of the file name. It will not use the long file name for the conversion. If any of the files use long file names, not only will you see a change of extension but also a change of name. If you wish to retain the full long name version, convert the files one at a time.

See also: [The Utilities Menu](#)

The Toolbar

The Toolbar is a row of buttons at the top of the main window which represent application commands. Clicking one of the buttons is a quick alternative to choosing a command from the menu. Buttons on the toolbar activate and deactivate according to the state of the program.

<u>Button</u>	<u>Action</u>	<u>Menu Equivalent</u>
	Locate and open a file	File <u>O</u> pen
	Save the file in the active window	File <u>S</u> ave
	Cut selected data to the Clipboard	Edit <u>C</u> ut
	Copy selected data to Clipboard	Edit <u>C</u> opy
	Paste data from Clipboard	Edit <u>P</u> aste
	Print the contents of the current Window	File <u>P</u> rint
	Open the Midi Controller Window	Utilities <u>M</u> idi Controller Window
	Open the Sequencer Window	Utilities <u>S</u> equencer Window
	Open a Midi Monitor Window	Utilities <u>M</u> idi Monitor Window
	Open a File Conversion Window	Utilities <u>F</u> ile Conversion Window
	Open a Driver Creator Window	Utilities <u>D</u> river Creator Window
	Display help file contents	Help Active Window

For information on how to use Help, press F1 or select Using Help from the Help menu.

File Menu

The File menu provides commands for creating new files, opening existing files, saving files, printing files, and exiting the application.

<u>New</u>	Create a new, untitled, document.
<u>Open</u>	Open an existing file.
<u>Close</u>	Close the current document.
<u>Save</u>	Save the current document if its contents have changed.
<u>Save As</u>	Save the current document under a new name.
<u>Exit</u>	Exit mq application.

Paste (Bank)



Purpose:

Paste works in combination with Copy. Paste takes the Patches you have copied to the Bank clipboard and pastes them into the locations currently selected in the active bank.

Paste is the primary method of moving one or more Patches to a new location within the same Bank or to different Banks of the same type. Reorganize Banks of Patches you already have. It's easy to replace those questionable Patches with new ones or your old favorites.

After creating a new Bank from the Driver List Window, use the Paste function to create new Banks by copying Patches from a selection of different Banks and pasting them into a new Bank.

We suggest you create a different Bank of Patches for each composition. This way it will be easier to keep track of which Patches are used by each piece of music and you will lessen the risk of inadvertently losing a Patch.

Directions:

To Paste Patches:

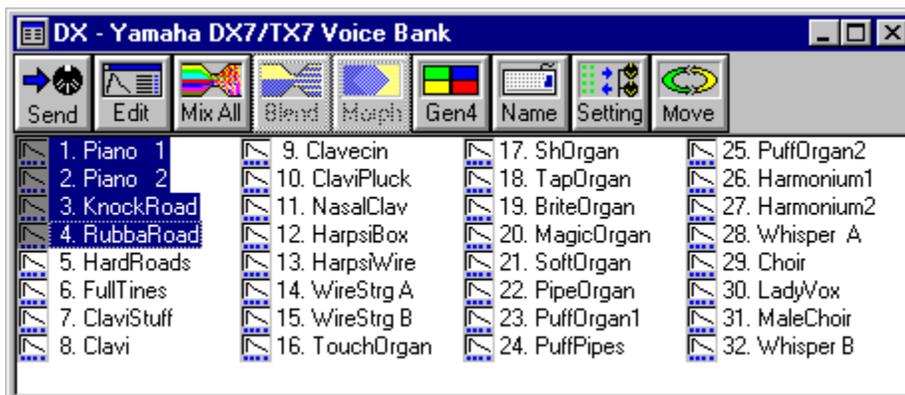
- (1) Copy the Patch(es) you wish to paste
- (2) Select the destination location(s) for the Patches
- (3) Choose *Bank Edit/Paste*

Example:

To paste Patches 1 - 4 over 8 - 12 in the same Bank

- (a) Select Patches 1 - 4.
- (b) Choose *Bank Edit/Copy*
- (c) Select Patches 9 - 12
- (d) Choose *Bank Edit/Paste*

Before the copy and paste:



After the copy and paste:



See Also: [Bank Editor Window](#)

Copy (Bank)



Purpose:

Copy places a copy of the Bank's selected patches in the Bank's clipboard. Once in the clipboard, the patches can be pasted to different locations in the same bank or in a different bank.

The Copy function is used in conjunction with Paste and Swap. It is necessary to first copy the Patches before they can be pasted or swapped with other Patches. For examples, see the Bank Editor/Paste function.

Directions:

To Copy one or more Patches:

- (1) Select the Patch(es) to copy
- (2) Choose *Bank Edit/Copy*

Note: when pasting or swapping source Patches to or with other Banks, the Bank containing the source Patches must remain open in the program.

See Also: Bank Editor Window

Window Menu

The Window menu provides commands to control the position and layout of application's windows.

<u>Cascade</u>	Resize and position all windows in an overlapping pattern.
<u>Tile</u>	Resize and position all windows in an non overlapping pattern.
<u>Arrange Icons</u>	Align all iconized windows along a grid.
<u>Close All</u>	Close all windows.

Auditioning a Patch

When Options|Audition is checked, clicking on one Patch will automatically send that Patch to the instrument via MIDI.

Double click on the Patch to play the chord currently set in the Tones Window. This allows you to immediately hear the Patch (or other type of data) that you have sent to the instrument

Note: The chord is transmitted on the MIDI Channel currently assigned to the Bank Editor Window. See Settings to change the MIDI Channel.

Note: most Banks offer the Auditioning feature. A few however do not. Functional limitations within an instrument is the usual reason for not offering this feature. To determine whether Patches can be auditioned, choose Bank Edit/Auditionable?.

Bank Editor Window



The Bank Editor is used to load and organize a collection or "Bank:" of Patches. The size of a Bank is determined by the architecture of the instrument. For example, a Yamaha DX7 has 32 Patches so the Bank size is 32.

The primary purpose of the Bank Editor is to provide a selection of tools you can use to keep your Patches organized. Use the Copy, Paste, and Swap functions to move Patches within a Bank or to a different Bank. It's easy to create custom Banks by opening a new Bank from the Driver List Window and pasting Patches from existing Banks into the new Bank Editor Window. Patches can also be moved from one window to another by using the drag and drop facilities of the program or by using the Move button.

When using the Drag and Drop facilities, it is possible to control how the drag and drop works. If a patch is dragged and dropped within the same bank, then there are three options: overwrite the destination patch with the source patch, swap the source patch and destination patch, or insert the patch at the destination patch location. When dragging and dropping to a different bank, the source patch can either overwrite the destination patch or can be inserted at the destination patch location. The drag and drop options are set in the Preferences Bank Tab where there is additional information on these functions.

Along with organizing Patches within a Bank, the Bank Editor Window provides excellent tools for creating new Patches. Begin by selecting two or more Patches and then choose one of the five different Randomize functions. The program will proceed to make useful new Patches for you. No editing knowledge necessary! Have a look at the functions: Mix, Mix All, Blend, Morph, and Gen4.

In most Banks, it's easy to audition different Patches stored in a Bank. Just click on a Patch and the program will send it to the instrument. Double clicking on a Patch will play the chord defined in the Preferences Tones Tab. There's no need to go to the synth to audition Patches. If you have problems auditioning, make sure the Midi Channel, Comm Channel, and Midi Ports are set correctly by selecting the Settings button.

Banks are also capable of responding to remote patch changes. If a patch change command arrives at a MIDI port that the program is monitoring, it will route this patch change to each bank window. If the bank is using the same MIDI channel as the patch change, the bank will automatically select the requested patch.

A useful function, particularly if you keep your Library sorted, is the Quick Search. After selecting any patch in the Library, you can press any alphanumeric key on the computer keyboard the you will automatically jump to the next Patch that starts with the selected letter. If you know that the patch you are looking for starts with an H, this is an extremely fast method of getting there.

Note 1: While many instruments use essentially the same types of data, the manufacturer's of these instruments insist on calling the data by different names. For example, a tonal source for an instrument might be called a sound, voice, patch, tone, preset, etc. Regardless of the name, the data still defines a tonal source. As a result of these inconsistencies and lack of industry standard, Sound Quest will refer to an entry in a Bank Editor Window as a "Patch" even if the data you are viewing is called something else by the manufacturer.

Note 2: Additionally, on most newer instruments there is a multi-timbral setup data structure where the different tonal sources are assigned to different MIDI channels. This structure, depending on the manufacturer, may be called a performance, multi, combi, global, section and so on. Regardless of the name, it still performs the function of creating a multi-timbral setup. In the few instances where it is necessary to distinguish between a "Patch" as the tonal source and a multi-timbral setup, the help file refers to the multi-timbral setup as a "Multi Patch".

Bank Edit Menu

The Bank Editor menu contains over 20 functions with which to organize your instrument's Banks of Patches. These functions are all contained in the Bank Edit menu. A Bank Editor Window must be active before you can see or access the Bank Edit Menu.

Undo	undo the last edit
<u>C</u> opy	Copy the currently selected Patches to the Bank clipboard
<u>P</u> aste	Paste the Bank clipboard Patches over those currently selected
<u>S</u> wap	Swap the copied Patches with the currently selected Patches
<u>C</u> lear	Zero all parameters for each selected Patch
<u>S</u> elect <u>A</u> ll	Select all of the Patches in the Bank for processing
<u>R</u> estore	Restore the selected Patches to their initial state
<u>E</u> dit/ <u>P</u> atch	Place the selected Patch in a Patch Editor Window
<u>E</u> dit/ <u>P</u> atch <u>N</u> ame...	Open the Patch Name Dialog to edit the selected Patch's name
<u>E</u> dit/ <u>A</u> uto <u>N</u> ame...	Auto name all of the patches in the bank
<u>E</u> dit/ <u>U</u> ppdate <u>S</u> ettings...	Open the Settings Dialog to set the Bank's basic parameters
<u>E</u> dit/ <u>O</u> pen <u>S</u> ysX <u>V</u> iew	Copy the Bank's data into a SysX View window for viewing
<u>R</u> andomize/ <u>M</u> ix	Create a new Bank by randomly mixing two selected Patches
<u>R</u> andomize/ <u>M</u> ix <u>A</u> ll	Create a new Bank by randomly mixing all selected Patches
<u>R</u> andomize/ <u>B</u> lend	Create a new Bank by mixing two Patches
<u>R</u> andomize/ <u>G</u> en <u>4</u> ...	Create a new Bank by taking portions of 4 Patches
<u>R</u> andomize/ <u>M</u> orph	Create a new Bank by incrementally moving from one Patch to another
<u>O</u> rganize/ <u>S</u> ort <u>A</u> ll	Sort the entries in the Bank in ascending alphabetical order
<u>O</u> rganize/ <u>R</u> otate <u>R</u> ange <u>U</u> p	Shift a range of selected Patches up
<u>O</u> rganize/ <u>R</u> otate <u>R</u> ange <u>D</u> own	Shift a range of selected Patches down
<u>O</u> rganize/ <u>I</u> nvert	Invert the locations of the selected Patches
<u>O</u> rganize/ <u>P</u> rint <u>X</u> Reference	Print a list of Patches used by Multis in a Multi Bank
<u>A</u> udition/ <u>P</u> lay <u>T</u> ones <u>C</u> hord	Play the chord defined in the Tones Window
<u>A</u> udition/ <u>P</u> lay <u>T</u> ones <u>S</u> equence	Play the sequence defined in the Tones Window
<u>A</u> udition/ <u>A</u> uditionable?	Indicates whether this type of data can be auditioned
<u>V</u> iew <u>M</u> ode/ <u>C</u> olumns	display the bank in a column
<u>V</u> iew <u>M</u> ode/ <u>N</u> ames	
<u>V</u> iew <u>M</u> ode/ <u>S</u> mall <u>I</u> cons	
<u>V</u> iew <u>M</u> ode/ <u>L</u> arge <u>I</u> cons	
<u>M</u> ove <u>T</u> o...	Moves the Bank or selected Patches into another window

You may display on your screen simultaneously as many Banks from as many different MIDI instruments as you wish (limited only by available RAM).

Exiting (Files Menu)

Purpose:

Exiting exits the application.

Directions:

(1) Choose *File/Exit* from the menu.

See Also: [Files Menu](#)

Edit Window

The edit windows in this application allow reading, editing, printing and saving text files up to 64K characters in size.

Drag/Drop from the File Manager

A file can be opened into an editor in the application by dragging the file from the File Manager, and dropping it on mq's main window.

Edit Copy Command

The Edit|Copy command leaves the selected text intact and places an exact copy of it in the clipboard. To paste the copied text into another document, choose Edit|Paste.

Edit Cut Command

The Edit|Cut command removes the selected text from your document and places the text in the clipboard. Choose Edit|Paste to paste the cut text into another document. The text remains selected in the clipboard, and can be pasted multiple times.

Edit Paste Command

The Edit|Paste command inserts the text currently selected in the clipboard into the current window at the cursor position.

File Close Command

File|Close closes the currently active window.

File Exit Command

The File|Exit command exits mq. If you've modified documents without saving, you'll be prompted to save before exiting.

File New Command

The File|New command opens a new, untitled document, and makes it the active window. The application prompts you to name untitled documents when they are closed.

File Open Command

The File|Open command displays the Open a File dialog box so you can select a file to load into a new document window. You can also create a new document by naming a file that doesn't currently exist.

File Save Command

The File|Save command saves the document in the active window to disk. If the document is unnamed, the Save File As dialog box is displayed so you can name the file, and choose where it is to be saved.

File Save As Command

The File|Save As command allows you to save a document under a new name, or in a new location on disk. The command displays the Save File As dialog box. You can enter the new file name, including the drive and directory. All windows containing this file are updated with the new name. If you choose an existing file name, you are asked if you want to overwrite the existing file.

Swap (Bank)

Purpose:

Swap is used to swap Patches in a Bank. Swap can also swap Patches between two banks.

The program allows you to select any number of Patches which are then "copied". After copying, select the Patches that are to be swapped with the "copied" Patches. The same number of Patches should be selected for swapping as were selected for copying. otherwise, the program will stop swapping when it runs out of either "copied" or selected Patches.

Directions:

To Swap multiple Patches:

- (1) Select one or more Patches to swap
- (2) Choose *Bank Edit/Copy*
- (3) Select the Patches to swap with the copied Patches
- (4) The number of copied Patches should equal the number of Patches now selected for swapping
- (5) Choose *Bank Edit/Swap*

Example:

eg. To swap Patches 1 - 4 with 9 - 12

- (a) Select Patches 1 - 4.
- (b) Choose *Bank Edit/Copy*
- (c) Select Patches 8 - 12
- (d) Choose *Bank Edit/Swap*

Before swapping:



After swapping: Patches 1-4 have been swapped with 9-12



See also: [Bank Editor Window](#)

Window Arrange Icons Command

The Window|Arrange Icons command arranges all iconized windows into rows along the bottom of the application's main window.

Window Cascade Command

The Window|Cascade command arranges all document windows from the top-left position of the application's main window so that the title bar of each is visible.

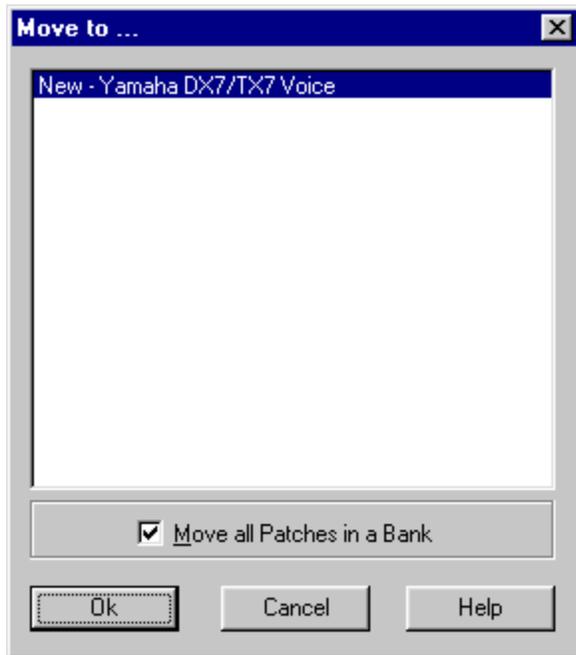
Window Close All Command

The Window|Close All command closes all document windows open in the application.

Window Tile Command

The Window|Tile command arranges all document windows side-by-side in a non-overlapping pattern.

Move To (Common Elements)



Purpose:

The Move To dialog moves data files from one window to another.

All of the major program windows have a Move To button (except the Patch editor) and a menu equivalent. These windows are the Bank Editor, Patch Editor, Group, Library, and DBase Windows. The Move to feature transfers data files from one window to another without using the program's Drag and Drop facilities. Data files can be moved from one DBase to another, from a Library to a DBase, from a Bank Editor to a Group and so on. All tolled, there are 20 different possibilities.

Apart from the directions listed below, it is important to note the following restrictions:

- 1) When transferring Patches into a Bank, individual Patches will first be placed in the selected locations in the destination Bank, followed by empty locations in the Bank. Should there be neither of these available, Patches will not be transferred.
- 2) If the source window is a Bank and the destination window is a Library or another bank, check "Move all Patches" to move all of the Patches in the Bank or leave the option unchecked to move just the selected Patches.
- 3) When transferring an entire Bank to a Bank window, the destination Bank window will be overwritten regardless of what Patches are or are not selected.

Directions:

To move data files from one window to another:

- (1) Select the data files in a Group, DBase, or Library to move. In a Bank choose the Patches to move

- (2) Press the Move To button or the equivalent menu function
- (3) Select the window to move the data to and press the OK button

See Also: [Bank Editor Move To](#)

See Also: [Patch Editor Move To](#)

See Also: [Group Move To](#)

See Also: [DBase Move To](#)

See Also: [Library Move To](#)

See Also: [Common Elements](#)

Settings (Bank)



Purpose:

Update Settings opens the Settings Dialog which is used to edit the basic settings of a Bank. Settings include: Comm Channel, MIDI channel, MIDI In Port, MIDI Out Port, Patch #, Cntrl #, and key words assigned to the Bank. Each Bank has its own set of basic settings which may be changed at any time.

For more information, you should see the [Settings Dialog](#).

Directions:

To change the basic settings:

- (1) Choose the Settings button or *Bank Edit/Edit/Update Settings...*
- (2) The Settings dialog displays the Bank's Notes, Keys, Comm Channel, MIDI Channel, Port In, Port Out, Patch #, and Cntrl #
- (3) Make any necessary changes to these settings

See also: [Settings Dialog](#)

See also: [Bank Editor Window](#)

Clear (Bank)

Purpose:

Clear sets all parameter values for each selected Patch to zero. Use this function to clear Patches from a Bank. If you clear the unwanted Patches, you will find it much easier in the future to determine where new Patches can be added since the name will be blank.

Note: Clearing Patches is also necessary when you use the Move To dialog to transfer Patches between windows in the program. The dialog places the transferred Patches into these cleared Patch locations.

Directions:

- (1) Select the Patch(es) to clear
- (2) Choose *Bank Edit/Clear*

See Also: [Bank Editor Window](#)

Cursor Movement Keys

Key(s)	Function
Arrow key	Moves the cursor left, right, up, or down in a field.
End or Ctrl+Right Arrow	Moves to the end of a field.
Home or Ctrl+Left Arrow	Moves to the beginning of a field.
Page Up or Page Down	Moves up or down in a field, one screen at a time.

Dialog Box Keys

Key(s)	Function
Tab	Moves from field to field (left to right and top to bottom).
Shift+Tab	Moves from field to field in reverse order.
Alt+letter	Moves to the option or group whose underlined letter matches the one you type.
Arrow key	Moves from option to option within a group of options.
Enter	Executes a command button. Or, chooses the selected item in a list box and executes the command.
Esc	Closes a dialog box without completing the command. (Same as Cancel)
Alt+Down Arrow	Opens a drop-down list box.
Alt+Up or Down Arrow	Selects item in a drop-down list box.
Spacebar	Cancel a selection in a list box. Selects or clears a check box.
Ctrl+Slash	Selects all the items in a list box.
Ctrl+Backslash	Cancel all selections except the current selection.
Shift+ Arrow key	Extends selection in a text box.
Shift+ Home	Extends selection to first character in a text box.
Shift+ End	Extends selection to last character in a text box

Editing Keys

Key(s)	Function
Backspace	Deletes the character to the left of the cursor. Or, deletes selected text.
Delete	Deletes the character to the right of the cursor. Or, deletes selected text.

Help Keys

Key(s)	Function
F1	<p>Gets Help and displays the Help Index for the application. If the Help window is already open, pressing F1 displays the "Using Windows Help" topics.</p> <p>In some Windows applications, pressing F1 displays a Help topic on the selected command, dialog box option, or system message.</p>
Shift+F1	<p>Changes the pointer to  so you can get Help on a specific command, screen region, or key. You can then choose a command, click the screen region, or press a key or key combination you want to know more about.</p> <p>(This feature is not available in all Windows applications.)</p>

Send Mail... (Files Menu)

Purpose:

Send Mail opens a file selector which allows you to select the file you wish to send and transmit it using the Microsoft Send file functionality.

Directions:

To send a file:

- (1) Choose "Files/Send Mail..."
- (2) Use the dialog box to select the file to send
- (3) The file is sent via the mail

See Also: [Files Menu](#)

Menu Keys

Key(s)	Function
Alt	Selects the first menu on the menu bar.
Letter key	Chooses the menu, or menu item, whose underlined letter matches the one you type, when a menu has focus.
Alt+Letter key	Pulls down the menu whose underlined letter matches the one you type.
Left or Right Arrow	Moves among menus of the main menu bar.
Up or Down Arrow	Moves among menu items within a drop-down menu.
Enter	Chooses the selected menu item.

System Keys

The following keys can be used from any window, regardless of the application you are using.

Key(s)	Function
Ctrl+Esc	Switches to the Task List.
Alt+Esc	Switches to the next application window or minimized icon, including full-screen programs.
Alt+Tab	Switches to the next application window, restoring applications that are running as icons.
Alt+PrtSc	Copies the entire screen to Clipboard.
Ctrl+F4	Closes the active window.
F1	Gets Help and displays the Help Index for the application. (See Help Keys)

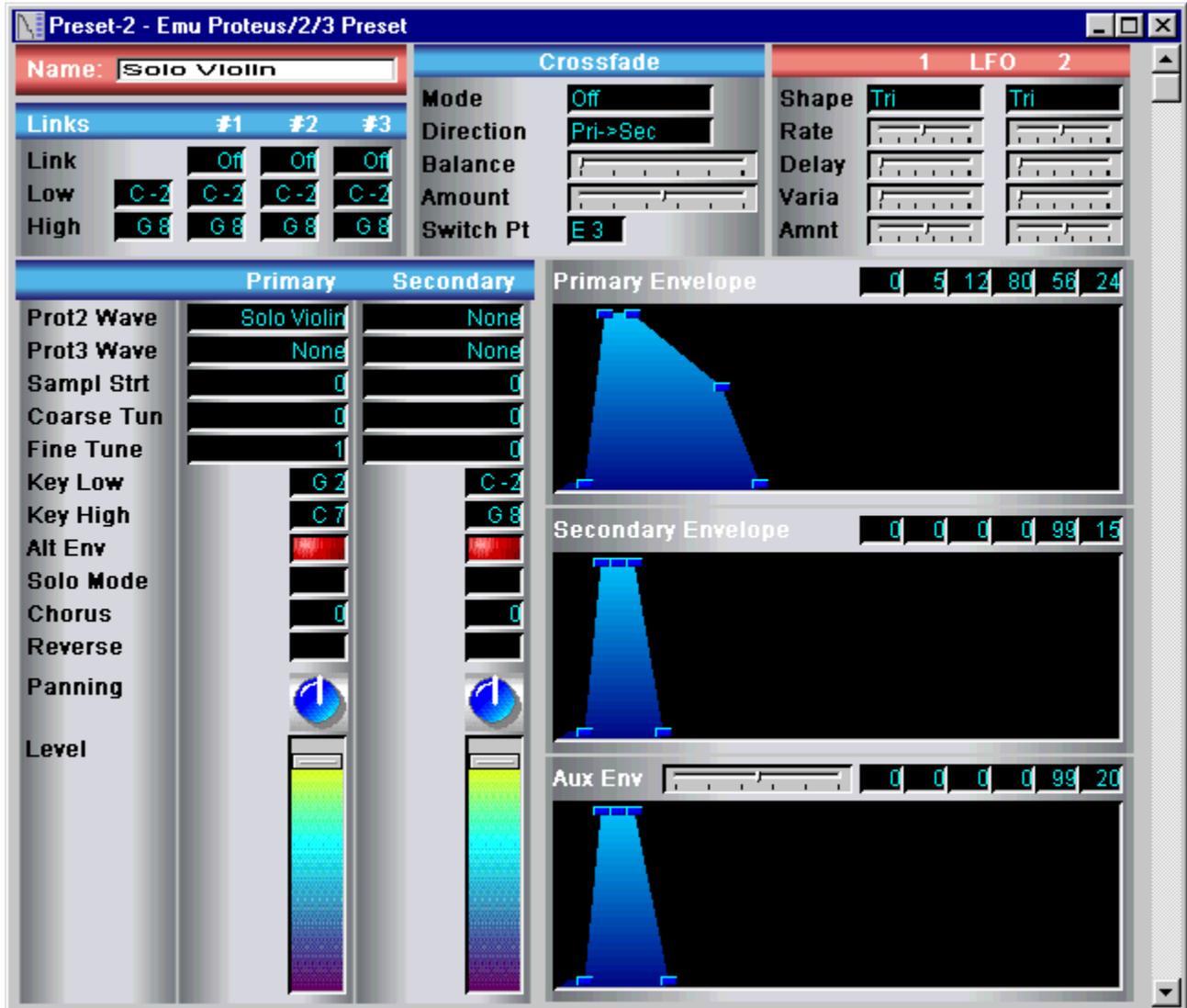
Text Selection Keys

Key(s)	Function
Shift+Left or Right Arrow	Selects text one character at a time to the left or right.
Shift+Down or Up	Selects one line of text up or down.
Shift+End	Selects text to the end of the line.
Shift+Home	Selects text to the beginning of the line.
Shift+Page Down	Selects text down one window. Or, cancels the selection if the next window is already selected.
Shift+Page Up	Selects text up one window. Or, cancels the selection if the previous window is already selected.
Ctrl+Shift+Left or Right Arrow	Selects text to the next or previous word.
Ctrl+Shift+Up or Down Arrow	Selects text to the beginning (Up Arrow) or end (Down Arrow) of the paragraph.
Ctrl+Shift+End	Selects text to the end of the document.
Ctrl+Shift+Home	Selects text to the beginning of the document.

Window Keys

Key(s)	Function
Alt+Spacebar	Opens the Control menu for an application window.
Alt+Hyphen	Opens the Control menu for a document window.
Alt+F4	Closes a window.
Alt+Esc	Switches to the next application window or minimized icon, including full-screen programs.
Alt+Tab	Switches to the next application window, restoring applications that are running as icons.
Alt+Enter	Switches a non-Windows application between running in a window and running full screen.
Arrow key	Moves a window when you have chosen Move from the Control menu. Or, changes the size of a window when you have chosen Size from the Control menu.

Patch Editor



About the Patch Editor

Patch Editors (the "Editors") are the core of the program's parameter editing capabilities. The editors allow you to change the values of individual parameters contained within the Patch data such as volume levels, panning, envelopes, etc. Each Editor is custom designed to match the architecture of the instrument. An M1 Patch Editor will look significantly different from the Proteus' Preset Editor shown above. Editors allow you to radically change or tweak sounds, design new multi-timbral configurations, Performances, or Combis, set up drums, modify an instrument's System Setup, or change just about any instrument parameter accessible via SysX.

Editing occurs through the use of a "template". The template contains all the information necessary to modify a Patch. Templates are stored on disk and are automatically read into the computer's memory when Editors are opened. The layout of the template is determined by the Patch's parameters and will include one or more of the following: graphic sliders, levels, envelopes, knobs, toggles, and image displays, as well as variable characters, string variables for entering names, and numeric displays.

Editors may also be displayed as floating editors. These editors can show an entire editor or only a portion there of and always sit on top of all other normal opened windows. This feature allows you to create small editors which can sit on top of a sequencer for immediate interactive editing. For more information on floating editors, please see: [Floating Editors](#).

For information on editing the parameters in a Patch editor, please see: [Editing Parameters](#)

For information on selecting parameters in a Patch editor, please see [Selecting Parameters](#)

For information on auditioning the Patch, please see [Setting the RMB Midi Channel](#)

For information on viewing Patch names in Combi/Multi Editors, see: [Viewing Patch Names](#)

For information on controlling an editor remotely, see [Controller Assign](#)

A Patch in a Patch Editor can be dragged and dropped between the Editor and other windows in the program. For more information, please see [Dragging and Dropping](#).

Note: The Patch Editor information has been included in the Midi Quest Jr. help file so that anyone considering upgrading to the full Midi Quest package can explore the extra features.

The Patch Editor Menu

The Patch Editor contains functions to assist in editing your instrument's Patches and other data. These functions are all contained in the Sngl Edit menu. The Patch Editor window must be activated before you can see and access the Sngl Edit menu.

Undo	Undo the last edit to the Patch
Copy	Copy the currently marked Patch parameters
Paste	Paste the copied parms into the currently marked parms
Forced Fill	Force the copied parms into the currently marked parms
Clear Selected Parameters	Unselect any currently selected parameters
Copy Selected Parameters to...	Copy any currently selected parameters to a Bank or Library
Randomize/Full Randomize	Randomize all marked parameters
Randomize/Constrn Randomize	Randomize all marked parameters based on current value
Randomize/Morph	Create a Bank of new Patches by morphing 2 Patches
Organize/Restore	Reset all marked parameters to their original values
Organize/Restore All	Restore the Patch to its pre-edited state
Organize/Buffer Save	Save the edited Patch in 1 of 4 buffers
Organize/Buffer Load	Load a Patch from a buffer into the editor
Open/Floating Editor...	Opens a dialog to open a floating editor for the current editor
Open/Update Settings...	Open the Settings Dialog
Open/New SysX View	Put the Patch data into a SysX View Window
Template Edit/Enable Control Drag	When checked, controls can be dragged
Template Edit/Control and text...	Dialog to edit control position and text parameters
Template Edit/Border...	Dialog to edit control borders
Template Edit/Extras...	Dialog to edit editors extra parameters
Template Edit/Controller Assign...	Dialog to assign a controller # to a control
Template Edit/Assign Color To Selected...	Dialog to assign new colors to selected controls
Template Edit/Delete Selected Controls	Delete the currently selected controls
Template Edit/Delete Unselected Controls	Delete the currently unselected controls
Template Edit/Move Selected Controls	Dialog to move the currently selected controls
Template Edit/Save Template	Save the modified editor template to disk
Template Edit/Save Template As...	Save the modified editor template to disk with a new name
Template Edit/Save Template As Floating Editr..	Save the current editor template to disk as a floating editor
Fast Tips...	Open the help window for the Patch editor
Move To...	Move the Patch data into another window

Midi Send (Patch Editor)

Purpose:

Midi/Data to Instrument is used to transmit the Patch in the Patch Editor window.

Each time a Patch Editor is activated, the Patch is automatically sent to the instrument. This keeps the instrument "in sync" with the current Patch you are editing. This feature allows you to edit several Patches simultaneously without the instrument becoming confused.

Of course, if you use your instrument to make changes to your Patches without using your Patch Editor, the instrument and Patch Editor will be "out of sync". To correct this, you will need to resend the data from the Patch editor to the instrument using *Midi/Data to Instrument*.

Directions:

To send the Patch in a Patch Editor to an instrument:

- (1) Click on the Window containing the Patch you wish to transmit
- (2) Choose *Midi/Data to Instrument*

See Also: [Patch Editor](#)

Editing and Selecting Patch Parameters

Purpose:

The most important attribute of a Patch Editor is, of course, its editing capabilities. Patch Editors utilize a number of different methods of editing or changing the value of a parameter. These methods are: 'Grab and Drag', right click editing, direct numeric entry, incremental editing, and double clicking on a parameter. There are also specific methods for editing graphic envelopes and Patch names. Each of these methods is described below.

The first step is to choose the parameter you wish to edit. Just point your mouse over the parameter and click on it. The chosen parameter is encircled in red.

Grab and Drag Editing

'Grab and Drag' editing is probably the most popular method of editing a parameter. By clicking on the parameter and dragging the mouse, any value in the parameter's range can be chosen.

To edit a parameter with 'Grab and Drag' editing:

- (1) Choose the parameter you wish to edit
- (2) Hold the mouse down and drag it left/right to change the value of the parameter
- (3) As the mouse is dragged, the value of the parameter will change
- (4) When the desired value is reached, release the mouse button

Right Click Editing

Right click editing is used with numeric and text parameters.

If the parameter has a small value range then a popup menu will be displayed showing all of the parameter values. To select a new value for the parameter, simply click on the value you want. If the parameter has a wide range of values then right clicking on it will have the same effect as if you had double clicked on it. For the numeric parameter, you will see a new control with up/down controls which allows you to change the value. For text parameters, the Parameter Selector Dialog is displayed and it is used to change the parameter's value. For more information, please see the Double click section below.

Direct Numeric Editing

Direct numeric entry is useful when the precise value of a parameter is known. In this case, it can be typed directly into the editor. An example of this would be creating a new Patch by entering the values from a Patch sheet.

To give a parameter a particular value:

- (1) Choose the parameter you wish to edit
- (2) Type in the first digit of the desired value
- (3) The number will appear at the location of the current parameter
- (4) Finish entering the value and press <RETURN> or <ENTER>
- (5) Step 4 is absolutely necessary to enter the change into the data

Note: vertical and horizontal sliders do not accept direct numeric editing.

Incremental Editing

Use incremental editing to change the value of a chosen parameter using a computer keyboard instead of the mouse. The two keys '<' and '>' will decrement and increment the value of the parameter respectively. Hold either of these keys down to scroll through the range of possible values. For horizontal and vertical sliders, use the left and right arrow keys.

To change a parameter's value with incremental editing:

- (1) Choose the parameter you wish to edit
- (2) Use '<' key to decrease the value. Use '>' key to increase the value

Note: Use the left and right arrow keys for incremental editing of the vertical and horizontal sliders

Graphic Envelope Editing

All editing of Graphic Envelopes is performed using the mouse. In most instances there are also parameters in the editor showing each of the envelope points numerically. If you wish to edit the envelope values with the computer keyboard, perform the edits on these numeric values.

Each editable envelope point will be indicated with a box. While most envelope points can be edited in both the horizontal and vertical directions, Depending on the instrument's envelope hardware, some of the envelope points may be able to move in only the horizontal or only the vertical directions.

To change the shape of a graphic envelope:

- (1) select one of the envelope's corner boxes
- (2) Hold the mouse button down and drag the envelope into the desired shape.

Note: Each corner can only be dragged within the limits set by the instrument's envelope.

Editing a Patch Name

To edit a Patch name:

- (1) Select a character in the existing name
- (2) A cursor is displayed
- (3) Type in the new name and
- (4) Press <RETURN>. You MUST press <RETURN> to enter the new name into the data.

Double Click

You may double click on any parameter which displays a list of optional text selections (eg. On, Off, Both, Single). This will bring up a Parameter Selector Dialog. which will display the entire range of parameter values.

To change a text selection's value by double clicking:

- (1) Click on the different parameter values until you find the one you want
- (2) Press OK to accept the new value or
- (3) Press Cancel to return to the original value

If you double click on a Numeric value, you will get a new display with up and down arrows that you can use to increment and decrement the value of the parameter or you can use the computer keyboard to type in a new value.

See Also: [Patch Editor](#)

Selecting Parameters (Patch Editor)

Purpose:

Several of the functions provided in the Sngl Edit menu operate on the selected parameters in a Patch Editor. These include: Copy, Copy Selected Parameters to..., Paste, Forced Fill, Randomize, %Randomize, Morph and a number of the template editing functions.

Use *Sngl Edit/Clear All Selected Parameters* to unselect the selected parameters.

Directions:

To select a group of parameters:

- (1) Click the mouse above and to the left of the parameter group you wish to select
- (2) Holding the button down, drag the mouse around the parameters you wish to select
- (3) While dragging, a selector box will encircle these parameters
- (4) Release the mouse and each selected parameter will be encircled with a green border

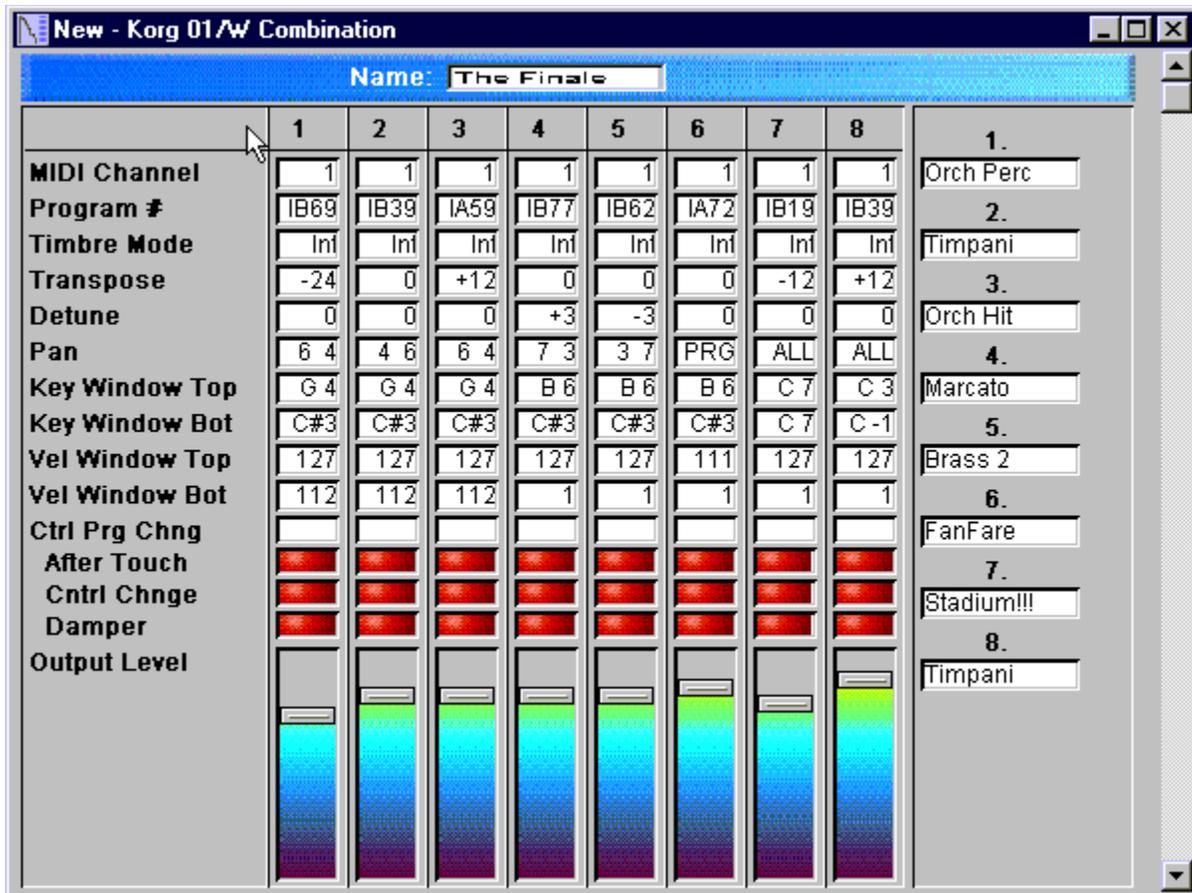
You may select additional parameters by repeating the above steps.

To select an individual parameter:

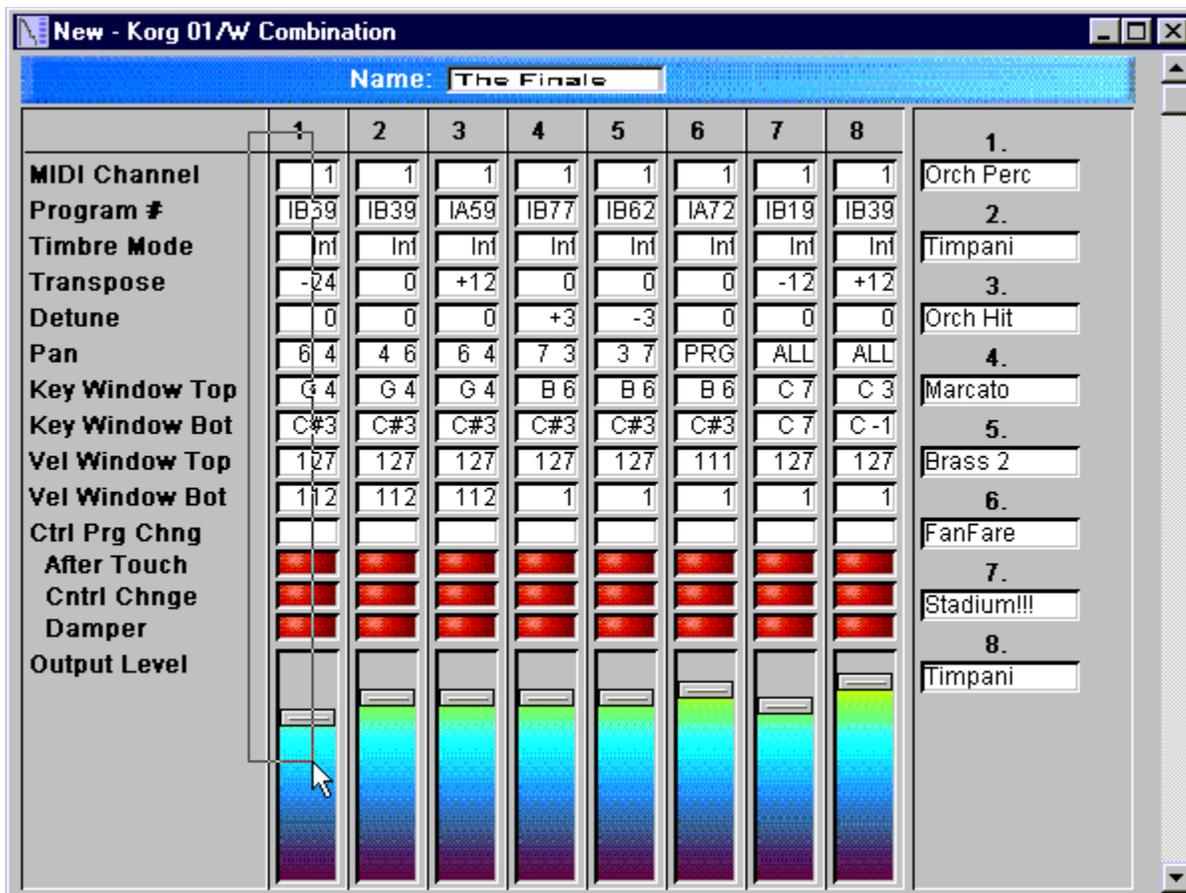
- (1) You may also toggle the select state of individual parameters by SHIFT-clicking on the parameter.
- (2) Each click will toggle the select state of the parameter.

Example:

- (1) Point the mouse above and to the left of the first parameter of the group to select



- (2) Hold down the left mouse button and drag so that the upper left corner of the parameters you wish to select are encircled
- (3) An encircling box is drawn to show which parameters will be selected



(4) Release the mouse button and the selected parameters will be encircled in green

New - Korg 01/W Combination

Name: **The Finale**

	1	2	3	4	5	6	7	8	
MIDI Channel	1	1	1	1	1	1	1	1	1.
Program #	IB69	IB39	IA59	IB77	IB62	IA72	IB19	IB39	Orch Perc
Timbre Mode	Inf	2.							
Transpose	-24	0	+12	0	0	0	-12	+12	Timpani
Detune	0	0	0	+3	-3	0	0	0	3.
Pan	6 4	4 6	6 4	7 3	3 7	PRG	ALL	ALL	Orch Hit
Key Window Top	G 4	G 4	G 4	B 6	B 6	B 6	C 7	C 3	4.
Key Window Bot	C#3	C#3	C#3	C#3	C#3	C#3	C 7	C -1	Marcato
Vel Window Top	127	127	127	127	127	111	127	127	5.
Vel Window Bot	112	112	112	1	1	1	1	1	Brass 2
Ctrl Prg Chng									6.
After Touch									FanFare
Cntrl Chnge									7.
Damper									Stadium!!!
Output Level									8.
									Timpani

See also: [Patch Editor](#)

Settings (Patch Editor)

Purpose:

Update Settings opens the Settings Dialog which is used to edit the basic settings for the Patch. Basic settings include: Comm Channel, MIDI channel, MIDI In Ports, MIDI Out Ports, Patch #, Cntrl #, and key words assigned to the Patch. Each Patch has its own set of basic settings which may be changed at any time.

For more information, please see the [Settings Dialog](#).

Directions:

To change the basic settings:

- (1) Choose *Sngl Edit/Open/Update Settings...*
- (2) A dialog containing the data file's Notes, Keys, Communication Channel, MIDI Channel, Port In, Port Out, Patch #, and Cntrl # will be displayed.
- (3) Make any desired changes to the basic settings.

Note: not available in MQ Jr.

See also: [Settings Dialog](#)

See also: [Patch Editor](#)

Move To (Bank)



Purpose:

Move To moves a complete Bank of Patches or individual Patches into any of the following destination windows: Bank Editor, Patch Editor, Group, DBase, or Library window.

Directions:

To move selected Patches into another window:

- (1) Select the Patches you wish to move
- (2) Choose the Move To button or *Bank Edit/Move To...*
- (3) Ensure that "Move All Patches in a Bank" is not checked in the Move To dialog
- (4) Highlight the destination window and press "OK"
- (5) The selected Patches are moved to the destination window

To move the entire Bank of Patches

- (1) Choose *Bank Edit/Move To...*
- (2) Ensure that "Move All Patches in a Bank" is checked in the Move To dialog
- (3) The entire Bank of Patches are moved to the destination window

Note: To move selected Patches to new locations in the same Bank Window, see Bank Editor/Paste.

See Also: Move To Dialog

See Also: Bank Editor

Copy (Patch Editor)



Purpose:

Copy copies the selected parameter. It is used in combination with Paste and Forced Fill.

For information and examples, see Paste and Forced Fill.

Directions:

To copy the selected parameters:

- (1) Select the parameters to copy.
- (2) Choose *Sngl Edit/Copy* to select the parameters for a subsequent Paste or Forced Fill operation.

See Also: Patch Editor

Constrained Randomize (Patch Editor)

Purpose:

Constrained Randomize is one of two methods you can use to randomize parameters in a Patch Editor. Constrained Randomize uses the current value of each selected parameter and adds or subtracts a random amount within the selected percentage limit. This usually provides a more subtle and contextual sound shift than the [Full Randomize](#) function.

To select the percentage limit, use the % Randomize parameter in the Properties [Tones Tab](#).

It is important to note that both randomization features only work on parameters you have selected. It does not work on unselected parameters in the Patch Editor including the active parameter (unless it is also selected). Click on '[Select Parameters](#)' to learn how to select parameters.

Both types of randomization will quickly provide you a variety of sound alternatives.

Directions:

To randomize selected parameters and create a new sound:

- (1) [Select](#) the parameters to randomize
- (2) Choose *Sngl Edit/Randomize/Constrained Randomize*

See also: [Full Randomize](#)

See Also: [Patch Editor](#)

Paste (Patch Editor)



Purpose:

Paste in combination with Copy moves selected parameters from one Patch to another. This can significantly decrease sound development time.

The Paste function performs intelligent parameter checking before it Pastes parameters. The intent is to ensure that the same parameter values are copied from one Patch to another. Therefore, when pasting Patch parameters from one Patch editor to another, you **must** select the identical parameters in both the source and destination Patches for the Paste function to work. To copy one or more parameter values to different parameters, use Forced Fill.

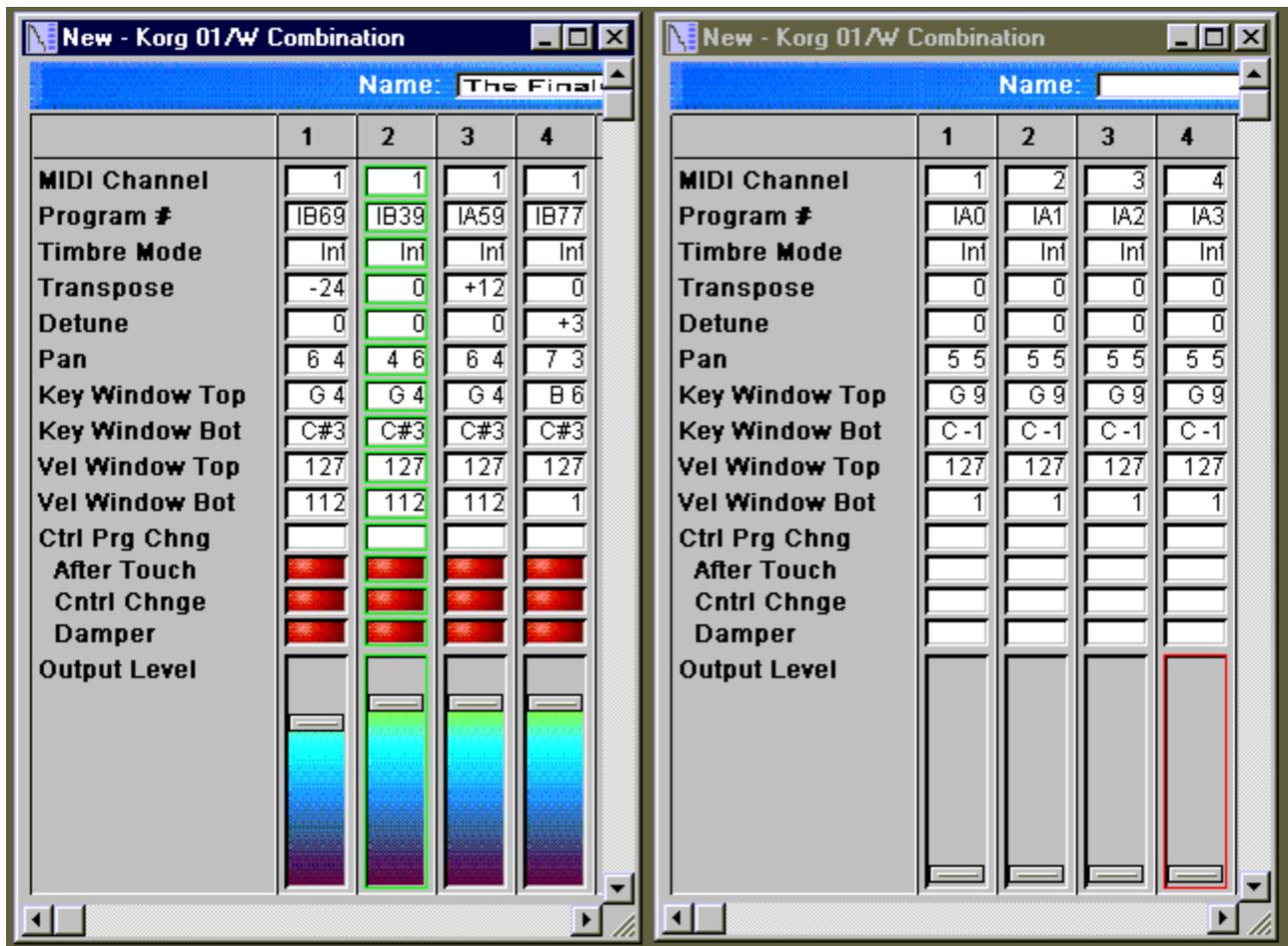
Directions:

To Paste selected parameter values in one Patch to the parameters in another Patch:

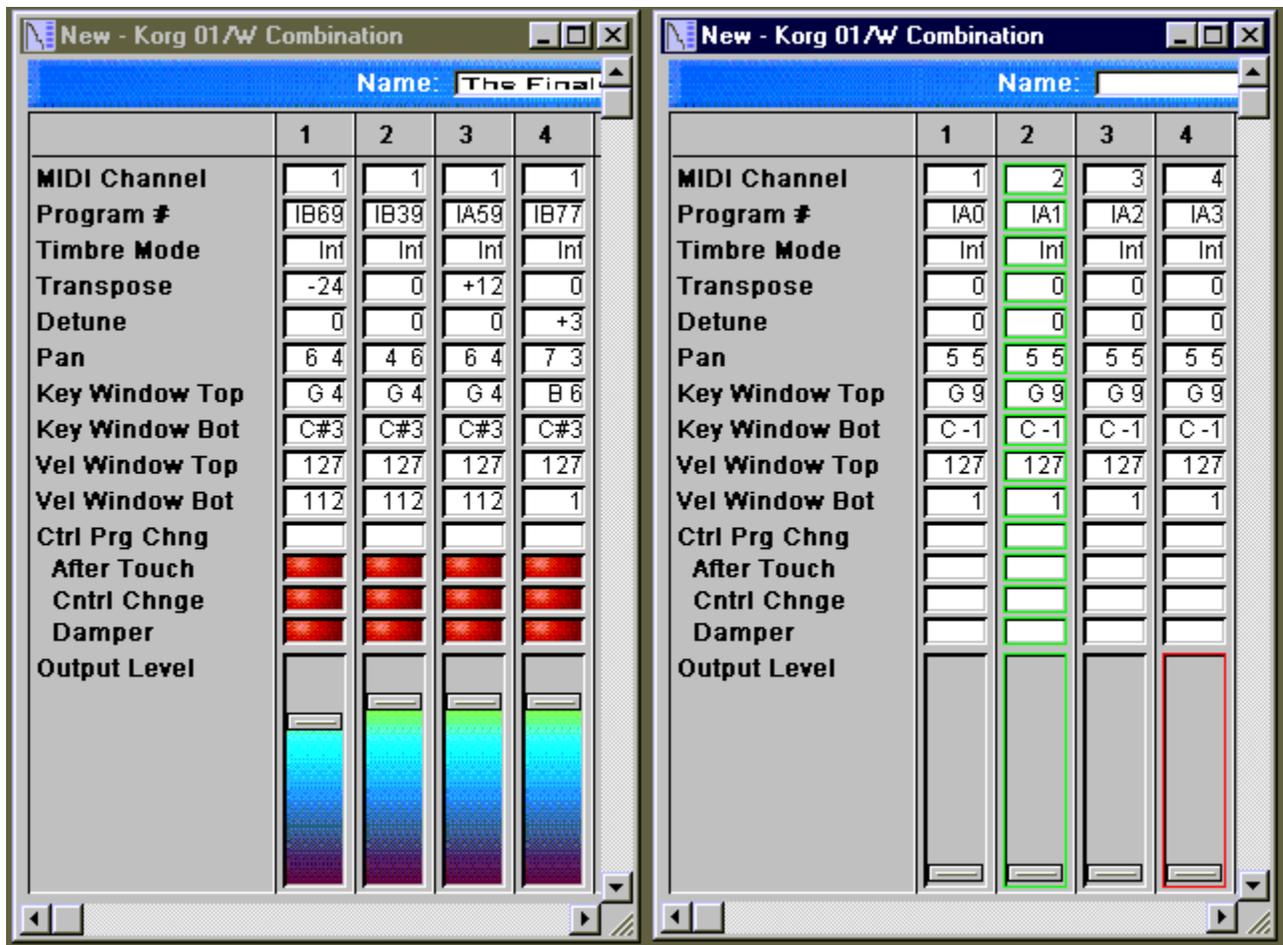
- (1) Select the desired parameters from the source Patch
- (2) Choose *Sngl Edit/C*opy.
- (3) Select the identical parameters in the destination Patch
- (4) Choose *Sngl Edit/P*aste

Example:

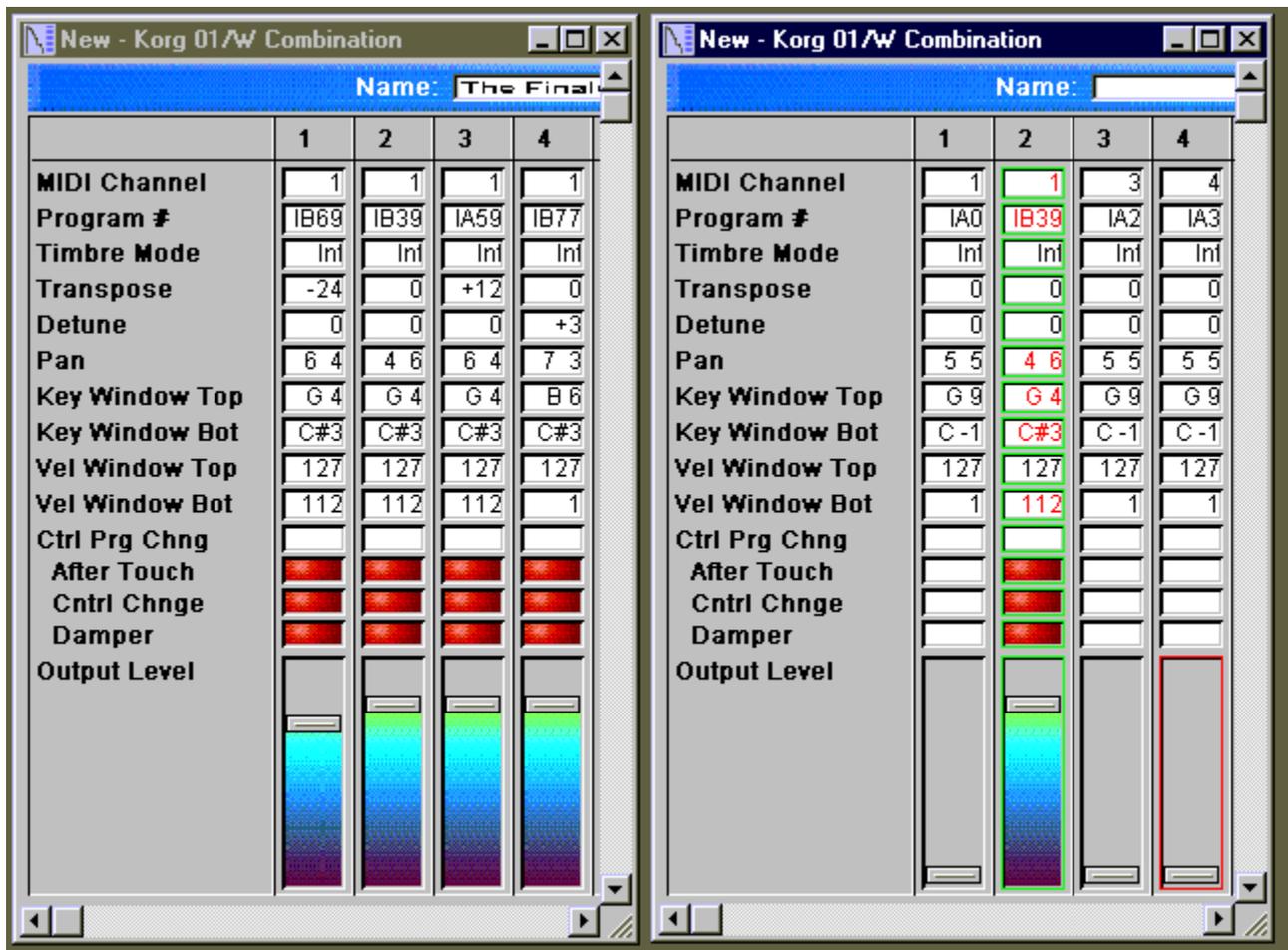
- (1) Open two editors of the same type in the program
- (2) Select the desired parameters in the left editor



- (3) Choose *Sngl Edit/Copy* while the left editor window is active
- (4) Select **the same** parameters in the editor on the right



- (5) Choose *Sngl Edit/Paste* while the right window is active
- (6) The parameters values are copied from the first window into the second



(7) Notice that the values of each of the parameters in the second column are the same

See Also: [Patch Editor](#)

Forced Fill (Patch Editor)

Purpose:

Using Copy and Forced Fill, you can easily move parameters of one Patch to a different location in the same Patch or to another Patch. For example, you can assign the envelope parameters in one of the Patch's sound sources, to other sound sources in the Patch. This can significantly decrease sound development time.

Unlike Paste, Forced Fill can be used to paste parameters into either the current editor or to a different editor of **any** type. Forced Fill does not perform any parameter type checking, so you can copy the chosen parameter values to any other currently selected parameters.

Directions:

To Force Fill the selected parameters:

- (1) Select the desired parameters from the source Patch
- (2) Choose *Sngl Edit/Copy*.
- (3) Select the parameters in the current editor or a different editor to Forced Fill into
- (4) Choose *Sngl Edit/Forced Fill*

Example:

To Force Fill the parameter of part 1 of an 01W Combination to part 5 of the same Combination.

- (1) Select the parameters to copy from:

New - Korg 01/W Combination

Name: **The Finale**

	1	2	3	4	5	6	7	8	
MIDI Channel	1	1	1	1	1	1	1	1	1. Orch Perc
Program #	IB69	IB39	IA59	IB77	IB62	IA72	IB19	IB39	2. Timpani
Timbre Mode	Inf	3. Orch Hit							
Transpose	-24	0	+12	0	0	0	-12	+12	4. Marcato
Detune	0	0	0	+3	-3	0	0	0	5. Brass 2
Pan	6 4	4 6	6 4	7 3	3 7	PRG	ALL	ALL	6. FanFare
Key Window Top	G 4	G 4	G 4	B 6	B 6	B 6	C 7	C 3	7. Stadium!!!
Key Window Bot	C#3	C#3	C#3	C#3	C#3	C#3	C 7	C -1	8. Timpani
Vel Window Top	127	127	127	127	127	111	127	127	
Vel Window Bot	112	112	112	1	1	1	1	1	
Ctrl Prg Chng									
After Touch									
Cntrl Chnge									
Damper									
Output Level									

- (2) Select *Sngl Edit/Copy*
- (3) Select the parameters to copy to:

New - Korg 01/W Combination

Name: The Finale

	1	2	3	4	5	6	7	8	
MIDI Channel	1	1	1	1	1	1	1	1	1. Orch Perc
Program #	IB69	IB39	IA59	IB77	IB62	IA72	IB19	IB39	2. Timpani
Timbre Mode	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	3. Orch Hit
Transpose	-24	0	+12	0	0	0	-12	+12	4. Marcato
Detune	0	0	0	+3	-3	0	0	0	5. Brass 2
Pan	6 4	4 6	6 4	7 3	3 7	PRG	ALL	ALL	6. FanFare
Key Window Top	G 4	G 4	G 4	B 6	B 6	B 6	C 7	C 3	7. Stadium!!!
Key Window Bot	C#3	C#3	C#3	C#3	C#3	C#3	C 7	C-1	8. Timpani
Vel Window Top	127	127	127	127	127	111	127	127	
Vel Window Bot	112	112	112	1	1	1	1	1	
Ctrl Prg Chng	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
After Touch	<input type="checkbox"/>	<input checked="" type="checkbox"/>							
Cntrl Chnge	<input type="checkbox"/>	<input checked="" type="checkbox"/>							
Damper	<input type="checkbox"/>	<input checked="" type="checkbox"/>							
Output Level	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

(4) Choose *Sngl Edit/Forced Fill*

(5) The following is the result of the Forced Fill

New - Korg 01/W Combination

Name: **The Finale**

	1	2	3	4	5	6	7	8	
MIDI Channel	1	1	1	1	1	1	1	1	1. Orch Perc
Program #	IB69	IB39	IA59	IB77	IB69	IA72	IB19	IB39	2. Timpani
Timbre Mode	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	3. Orch Hit
Transpose	-24	0	+12	0	-24	0	-12	+12	4. Marcato
Detune	0	0	0	+3	0	0	0	0	5. Orch Perc
Pan	6 4	4 6	6 4	7 3	6 4	PRG	ALL	ALL	6. FanFare
Key Window Top	G 4	G 4	G 4	B 6	G 4	B 6	C 7	C 3	7. Stadium!!!
Key Window Bot	C#3	C#3	C#3	C#3	C#3	C#3	C 7	C-1	8. Timpani
Vel Window Top	127	127	127	127	127	111	127	127	
Vel Window Bot	112	112	112	1	112	1	1	1	
Ctrl Prg Chng	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
After Touch	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Cntrl Chnge	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Damper	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Output Level	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

See Also: [Patch Editor](#)

Full Randomize (Patch Editor)

Purpose:

Full Randomize is one of two methods you can use to randomize parameters in a Patch Editor. Full Randomize sets each selected parameter to a random value within the valid range of the parameter as defined by the instrument.

It is important to note that both randomization features only work on parameters you have selected. It does not work on unselected parameters in the Patch Editor including the active parameter (unless it is also selected). Click on '[Select Parameters](#)' to learn how to select parameters.

Both types of randomization will quickly provide you a variety of sound alternatives.

Directions:

To randomize the selected parameters and create a new sound:

- (1) [Select](#) the parameters to randomize
- (2) Choose *Sngl Edit/Randomize/Full Randomize*

See Also: [Constrained Randomize](#)

See Also: [Patch Editor](#)

Clear Selected Parameters (Patch Editor)

Purpose:

Clear Selected Parameters deselects the selected parameters.

This removes the green borders that result from selecting parameters for pasting, copy to..., forced fill, and randomization functions.

Directions:

(1) Choose *Sngl Edit/Clear Selected Parameters*

You can also clear the selected parameters by left clicking on any non parameter space in the editor.

See also: [Selecting Parameters](#)

See also: [Patch Editor](#)

Restore (Patch Editor)

Purpose:

Restore returns the selected parameters to their original values.

After performing one or more edits, you may decide that some of the edited parameters should be restored to their original values. While it is possible to restore each edited parameter to its original value manually using the computer keyboard, it is easier to simply select each parameter you wish to restore and choose the restore option from the menus.

Directions:

- (1) Select the parameters to restore
- (2) Choose *Sngl Edit/Organize/Restore*

See Also: [Patch Editor](#)

Restore All (Patch Editor)

Purpose:

Restore All returns each of the selected parameters to their original, pre-edited state.

Directions:

(1) Choose *Sngl Edit/Organize/Restore All*

See Also: [Patch Editor](#)

Buffer Save (Patch Editor)

Purpose:

Buffer Save works in combination with [Buffer Load](#). Each Patch editor has four temporary buffers which are used to store "work in progress" by using Buffer Save. You may, for instance, create variations on a Patch and store them in the buffers. Once the various aural possibilities are created, they can easily be compared by using Buffer Load to move them from the buffer into the Patch Editor for auditioning or comparison.

Directions:

To store a Patch in a buffer:

(1) Choose one of the buffers in *Sngl Edit/Organize/Buffer Save*

See Also: [Buffer Load](#)

See Also: [Patch Editor](#)

Buffer Load (Patch Editor)

Purpose:

Buffer Load is used to retrieve a version of the Patch which you have previously stored in one of the four temporary buffers.

Buffer Load can be used in combination with Buffer Save to compare Patches while you are developing a new Patch. When you think the Patch is ready (but aren't sure), store the Patch in one of the four temporary buffers. Continue with your editing until you have another sound. Store it in a different buffer. You can then use Buffer Load to alternately load the two Patches to determine which one you prefer.

Directions:

To load a Patch stored in a temporary buffer:

(1) Choose *Sngl Edit/Organize/Buffer Load* to load a Patch from a buffer

Warning: The current Patch in the Patch Editor will be destroyed. If you do not want to lose it, store it to disk or in a temporary buffer.

See Also: Buffer Save

See Also: Patch Editor

Fast Tips (Patch Editor)

Purpose:

The "Fast Tips" help file is used to provide Patch Editor specific information such as the meaning of various short-forms and overall layout of the editor.

Directions:

To open Fast Tips:

(1) Choose *Sngl Edit/Fast Tips*

See Also: [Patch Editor](#)

Move To (Patch Editor)

Purpose:

Move To moves a Patch into the a different window.

In order to move the Patch in a Patch editor into a Bank Editor, DBase, or Library choose the Move To menu function. This will open a dialog window to select which window the Patch is directed to.

Directions:

To move the Patch into another window:

(1) Choose *Sngl Edit/Move To....*

See Also: [Move To Dialog](#)

See Also: [Patch Editor](#)

Viewing Patch Names (Patch Editor)

Purpose:

Multi style editors are capable of displaying the actual sound/Patch names referenced in most Multi style editors. To display the names however, it is necessary to have a Bank containing the names loaded into the program. If the Multi is being edited from a Group or DBase or the Multi is being edited from a Bank which in being edited from a DBase, the program first looks to the DBase for a Bank of the correct type to display the names. If there is no DBase involved in the editing, the Bank must be loaded in the program. With a correctly loaded Bank you will see for example, "Trumpet" instead of "INT 1" and "Thump Bass" instead of "INT 2".

Directions:

Option 1. Follow these steps to make sound names visible

- (1) Load an instrument group from the Driver List by double-clicking on the instrument bar's icon
- (2) Edit the Multi Bank from the newly created group
- (3) Edit a Multi from the Multi Bank.

Option 2. Follow these steps to make sound names visible:

- (1) Load the Bank containing the instrument's sounds (usually a Patch/Voice/Sound Bank).
- (2) Load a Multi out of the instrument. Along with the sound numbers will be a display of the actual sound names drawn from the Patch Bank loaded earlier

If the software can not find a loaded Patch Bank, it will display only the number of the selected sound.

Example:

This first graphic shows an 01/W Combi editor without a Patch Bank loaded. The right column only has the Patch reference numbers but not the actual names.

New - Korg 01/W Combination

Name: **The Finale**

	1	2	3	4	5	6	7	8	
MIDI Channel	1	1	1	1	1	1	1	1	1.
Program #	IB69	IB39	IA59	IB77	IB62	IA72	IB19	IB39	INT 169
Timbre Mode	Inf	2.							
Transpose	-24	0	+12	0	0	0	-12	+12	INT 139
Detune	0	0	0	+3	-3	0	0	0	3.
Pan	6 4	4 6	6 4	7 3	3 7	PRG	ALL	ALL	INT 59
Key Window Top	G 4	G 4	G 4	B 6	B 6	B 6	C 7	C 3	4.
Key Window Bot	C#3	C#3	C#3	C#3	C#3	C#3	C 7	C-1	INT 177
Vel Window Top	127	127	127	127	127	111	127	127	5.
Vel Window Bot	112	112	112	1	1	1	1	1	INT 162
Ctrl Prg Chng									6.
After Touch									INT 72
Cntrl Chnge									7.
Damper									INT 119
Output Level									8.
									INT 139

After loading an 01/W Patch Bank and redrawing the Combi editor, the following is displayed. The names of the Patches are now visible instead of the reference numbers.

New - Korg 01/W Combination

Name: **The Finale**

	1	2	3	4	5	6	7	8	
MIDI Channel	1	1	1	1	1	1	1	1	1.
Program #	IB69	IB39	IA59	IB77	IB62	IA72	IB19	IB39	Orch Perc
Timbre Mode	Inf	2.							
Transpose	-24	0	+12	0	0	0	-12	+12	Timpani
Detune	0	0	0	+3	-3	0	0	0	3.
Pan	6 4	4 6	6 4	7 3	3 7	PRG	ALL	ALL	Orch Hit
Key Window Top	G 4	G 4	G 4	B 6	B 6	B 6	C 7	C 3	4.
Key Window Bot	C#3	C#3	C#3	C#3	C#3	C#3	C 7	C -1	Marcato
Vel Window Top	127	127	127	127	127	111	127	127	5.
Vel Window Bot	112	112	112	1	1	1	1	1	Brass 2
Ctrl Prg Chng									6.
After Touch	<input type="checkbox"/>	FanFare							
Cntrl Chnge	<input type="checkbox"/>	7.							
Damper	<input type="checkbox"/>	Stadium!!!							
Output Level	<input type="checkbox"/>	8.							
									Timpani

See Also: [Patch Editor](#)

Setting the RMB Midi Channel in a Multi (Patch Editor)

Purpose:

This feature applies only to Multi style Patch Editors. Its purpose is to make it easier for you to audition chosen Sounds.

If you are working in any Patch Editor Window, just hold down the CTRL key and single-click the right mouse button to generate MIDI notes for auditioning. The type of auditioning is controlled in the Properties Tones Tab, where CTRL/clicking on the right mouse button will generate either a note, chord or sequence. If the Note (default) option is selected, then the position of the mouse in the screen determines the pitch of the note that is played. Moving the mouse from left to right produces higher pitches. Moving the mouse from the bottom of the screen to the top produces louder notes. This makes it easy to test a sound at various pitches and velocities. This option is not so good when developing drum kits because precise notes can not be accessed easily. In this case, open the Midi Controller Window to play notes to audition a drum kit under development.

If you use a 3 button mouse then instead of using the CTRL-click option, you can just press the middle mouse button to audition.

With most Multi style editors, one of the major purposes of the editor is to select different Patches for different MIDI channels. In order to provide reasonable auditioning capabilities, it is necessary to be able to change the MIDI channel used by the right mouse button. While the channel can be changed by selecting *Sngl Edit/Open/Update Settings...*, it is faster and easier to set this channel from the editor itself.

In cases where the Patch editor displays MIDI channels, selecting or editing one of those MIDI channels will automatically set the Right Mouse Button Midi channel to the selected channel. This makes auditioning the sound assigned to that channel much easier than going to the Settings Dialog to change the channel. If the editor has 16 parts with each part permanently assigned to a specific MIDI channel, clicking on the part number will automatically set the RMB (Right Mouse Button) to that channel. If you are using the built in MIDI Thru/mapping functionality, you need to audition at least one note for the MIDI Thru functions to pick up the new MIDI channel

See Also: [Patch Editor](#)

Sort All (Bank)

Purpose:

Sort All sorts all of the Patches in a Bank in alphabetically ascending order.

32-bit only: the 32 bit sort runs in a separate thread so you can proceed with other work while the sort is performed.

Directions:

- (1) Choose *Bank Edit/Organize/Sort All*
- (2) The Bank Editor will resort the Bank's contents and redisplay them

Before the sort:

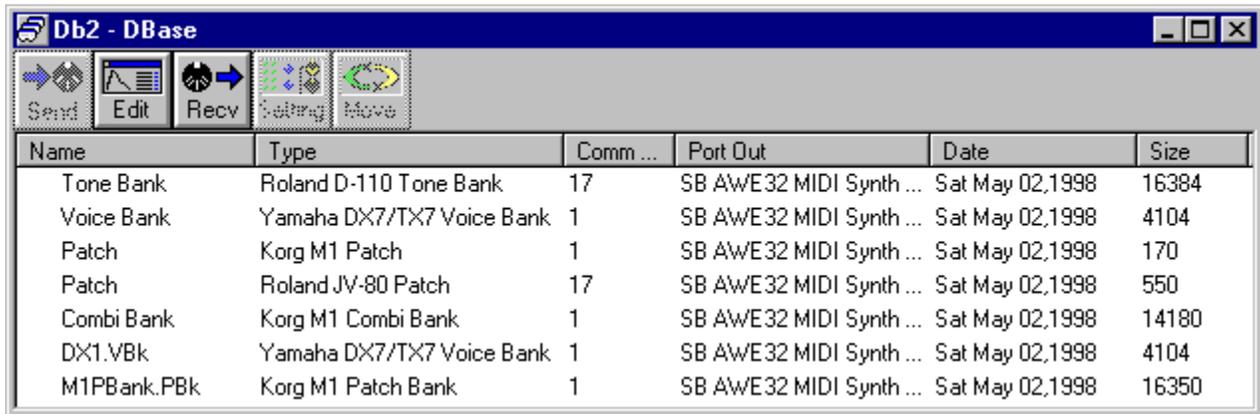


After the sort:



See also: [Bank Editor Window](#)

DBase Window



The screenshot shows a window titled "Db2 - DBase" with a toolbar containing icons for Send, Edit, Recv, Setting, and Move. Below the toolbar is a table with the following data:

Name	Type	Comm ...	Port Out	Date	Size
Tone Bank	Roland D-110 Tone Bank	17	SB AWE32 MIDI Synth ...	Sat May 02,1998	16384
Voice Bank	Yamaha DX7/TX7 Voice Bank	1	SB AWE32 MIDI Synth ...	Sat May 02,1998	4104
Patch	Korg M1 Patch	1	SB AWE32 MIDI Synth ...	Sat May 02,1998	170
Patch	Roland JV-80 Patch	17	SB AWE32 MIDI Synth ...	Sat May 02,1998	550
Combi Bank	Korg M1 Combi Bank	1	SB AWE32 MIDI Synth ...	Sat May 02,1998	14180
DX1.VBk	Yamaha DX7/TX7 Voice Bank	1	SB AWE32 MIDI Synth ...	Sat May 02,1998	4104
M1PBk.PBk	Korg M1 Patch Bank	1	SB AWE32 MIDI Synth ...	Sat May 02,1998	16350

Purpose:

The DBase is a simple yet powerful data storage facility which allows you to take ANY combination of Sound Quest data files and store them together in one place, the DBase. The DBase can hold, for example, an M1 Patch Bank, an M1 Combi Bank, 2 DX7 Voice Banks, a D-10 system setup dump, and a Wavestation Performance. It can, in fact, hold virtually any combination of SysX data that you can think of.

Unlike other Editor/Librarians, we have chosen not to limit the flexibility of this window as we have learned over time that different people like to organize their information in completely different ways. While Sound Quest originally envisioned this window would be used take system snapshots, some of our users prefer to use the DBase to store all of their Banks for one or more instruments. Others keep an entire collection of sequences stored in their DBases. The system snapshot, however, is still the most common use for the DBase.

A snapshot is a "picture" of the current state of one or more of your instruments or other devices in your MIDI setup. It contains all of the data necessary to restore the instrument(s) to the same state as at some other time. After the snapshot is taken, the DBase should contain all of your instruments' Banks along with any other sets of data which have been customized for the project. With this feature, it is possible to work on one project then take a snapshot of the instruments' settings before going on to a different project. When you return to the first project, in the afternoon, the next day, or a year from now, you will be able to send the SysX data back to your instruments and be confident that they will be properly configured for the piece.

After completing a piece of music, you should always create a DBase which has all of your instrument's relevant Patches, Banks, and other data stored in it. The structure can act as an isolation area so that the data isn't accidentally changed for use on another project.

In the Driver List window it is possible to create various Custom Sets of drivers. You can set up each Custom Set to contain a list of all of the different types of data utilized in a piece of music. At the end of the day, it's easy to select the custom set and use it to create a DBase to hold all of the relevant instrument settings for the project in one place. For more information, please see [Custom Sets](#).

A useful function, particularly if you keep your Library sorted, is the Quick Search. After selecting any patch in the Library, you can press any alphanumeric key on the computer keyboard the you will automatically jump to the next Patch that starts with the selected letter. If you know that the patch you are looking for starts with an H, this is an extremely fast method of getting there.

You may display simultaneously as many DBases on your screen as desired (limited only by available RAM). You may even cut or copy components of one DBase into another.

The DBase window supports drag and drop. Dragging and dropping any Midi Quest patch or bank file from Explorer into a DBase window will add the file to the existing list. DBases also support drag and drop for MIDIX files (.SYX). In this case, the file is automatically converted to Midi Quest format and added to the list.

For information on each column in the DBase, see [DBase Contents](#)
To send the entire contents of a DBase to the instruments, see [Sending a DBase](#).
To send one or more data files in a DBase, see [Sending Data Files](#).

DBase Menu

The DBase contains 16 menu functions with which to organize the Patches, Banks, and other data used by your various instruments. These functions are all contained in the DBase menu. The DBase window must be activated before you can see and access the DBase menu.

Cut	Cut the selected Data File(s) to the DBase clipboard
Copy	Copy the selected Data File(s) to the DBase clipboard
Paste	Paste the Data Files in the clipboard to the current DBase
Clear	Remove selected Data Files from the DBase
Select All	Select all entries in the DBase for processing
Open/Editor	Open the Data File into a Patch or Bank Editor
Open/SysX View	Open the Data File into a SysX View Window
Open/Update Settings...	Open the Update Settings Dialog for the selected data file
Open/New SysX Message...	Create a custom SysX message that is added to the DBase
Sort By	Sort the DBase by one of 5 methods
Disk/Load Data File	Load a Data File from disk into the DBase
Disk/Save Data File	Save a Data File in the DBase to disk as a separate file
Disk/Link a Data File	Link a Data File from disk into the DBase
Disk/Save Data File & Link	Save a Data File to disk and link the DBase to it
Disk/Resolve Links	Make any linked Data Files into files stored in the DBase
View Mode/Columns	Display the DBase using columns
View Mode/Names	Display the DBase using just names
View Mode/Small Icons	Display the DBase using small icons
View Mode/Large Icons	Display the DBase using large icons
Receive From Instrument	Add data for the Driver List drivers into the DBase
Move To...	Add the selected Data File to a different window

Sending a DBase (DBase)

Purpose:

Sending a DBase sends each data file in a DBase to its respective instrument.

One of the major advantages of using the DBase is that it can store a complete snapshot of your MIDI system.

If you create a DBase for each of your projects, you can easily jump back and forth between them. To return to a particular project just send its respective DBase. The DBase will properly configure your entire system.

Directions:

To send all of the data stored in a DBase to your instruments:

- (1) click on the Window containing the DBase you wish to send
- (2) Choose *Midi/DBase/Group to Instrument*

See Also: [DBase Window](#)

Sending Data Files (DBase)



Purpose:

Click once on the Send button to send selected data files in a DBase to their respective instruments.
Double click on the Send button to send all of the data files in the DBase.

Directions:

To send one or more files in a DBase to an instrument(s):

- (1) Select the data file(s) to send
- (2) Select 'Send' button or choose *Midi/Current Data to Instrument*

See Also: [DBase Window](#)

Cut (DBase)



Purpose:

Cut is one of two methods of placing data files into the DBase Clipboard. Copy is the other.

Cut removes the currently selected data files from the DBase and attaches them to the DBase Clipboard. The data files are now available to paste into the same or a different DBase.

This is useful for removing data files from a DBase where they are no longer necessary. The files can then be pasted to a different DBase which is used as a storage location for unused files.

Note: Before adding the data files to the DBase Clipboard, the program deletes any data files currently stored there.

Directions:

To Cut data and place it into the DBase clipboard

- (1) Select the data file(s) you wish to cut
- (2) Choose *DBase/Cut*

See Also: [DBase Window](#)

Copy (DBase)



Purpose:

Copy is one of two methods of placing data files into the DBase Clipboard. Copy makes a duplicate of the selected DBase data files. It attaches the copies to the DBase Clipboard for subsequent pasting into the same or a different DBase.

Copy provides an easy way of duplicating a data file before editing the copy. If you wish to create a new Patch or Bank based on an existing one, copy the data file to the DBase Clipboard using *DBase/Copy*. Next, use *DBase/Paste* to paste the copy of the data file back into the DBase. The copied data file may now be edited without destroying the original.

Note: Before adding the data files to the DBase Clipboard, the program deletes any data files currently stored there.

Directions:

To Copy a data file and place it in the DBase clipboard:

- (1) Select the data file(s) to copy
- (2) Choose *DBase/Copy*

See Also: [DBase Window](#)

Paste (DBase)



Purpose:

Paste copies data files in DBase Clipboard into the active DBase.

Before data files can be pasted into a DBase, data files from the current DBase or another DBase must be Cut or Copied. This places the data files in the DBase clipboard. Once in the clipboard, the data files can be pasted to the same DBase or a different one.

The primary use for Cut/Copy/Paste is to move data files between DBases.

Directions:

To paste the contents of the DBase Clipboard into a DBase:

- (1) Cut or Copy one or more data files from a DBase
- (2) Click on the DBase Window to paste the data into
- (3) Choose *DBase/Paste*

See Also: [DBase Window](#)

Clear (DBase)

Purpose:

Clear removes data files from the DBase which are no longer needed. Maybe you replaced the trashy piano you were using in a project with a great new Grand Piano sound. Well, get that old piano out of the DBase.

Warning: when data is cleared from a DBase it is permanent, the files can not be retrieved.

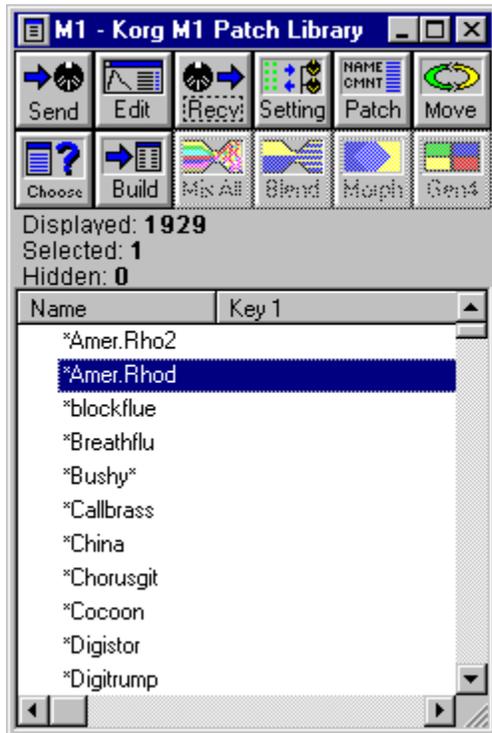
Directions:

To clear unwanted data files from a Data Base:

- (1) select the file(s) you wish to clear
- (2) Choose *DBase/Clear*

See Also: [DBase Window](#)

Library



About the Library

The Library is used to create a master listing of Patches for your instrument.

If you have a large number of Patches, the ideal place to store and organize them is in a Library. The advantage of the Library is that it can store a virtually unlimited number of Patches. This allows you to keep all of your Patches in one place.

Use the Library as a source area when you are looking for Patches for a project. If all of your Patches are kept in a Library, your Patches will be easy to find and you will be saved from having to hunt through large numbers of Banks on disk.

The Library has another advantage. You can attach keywords to each Patch in a Library. These keywords can be used to identify the tonal or other quality of the Patch. For example, if you code all of your bass Patches with the keyword BASS, you can ask the Library to only show you all of your BASS Patches. You will then have a complete list of all your basses and can select the one that is most appropriate. In the long run, you will find this to be a real time saver.

Each Library can hold only one type of Patch. For example, for the M1, you will need to create a separate library for Patch Patches, Multi Patches, and Global Patches.

You cannot create a Library of Banks. For instance, you can create a Patch Library but not a Patch Bank Library. However, by using Move To or Drag and Drop, you can easily move all of some of the Patches in a Bank into a Library. You do not need to transfer them one at a time. When this transfer occurs, the program automatically adds the bank's name and the position of the patch into the patch's keys in the Library. This allows you to rebuild the bank in the future if you wish.

It is possible to have as many Libraries as desired displayed simultaneously on the screen (limited only by available RAM).

A useful function, particularly if you keep your Library sorted, is the Quick Search. After selecting any patch in the Library, you can press any alphanumeric key on the computer keyboard the you will automatically jump to the next Patch that starts with the selected letter. If you know that the patch you are looking for starts with an H, this is an extremely fast method of getting there.

Individual Patches in the Library can be reordered as follows: (1) Choose the Patch to move. (2) Hold down the ALT key on the keyboard. (3) Use the up and down arrow keys to move the Patch to the desired location.

Library Menu

The Library contains 28 menu functions with which to organize the Patches and other data used by your various instruments. These functions are all contained in the Library menu. The Library Window must be activated before you can see and access the Library menu.

<u>Cut</u>	Cut the selected Patches to the Library clipboard
<u>Copy</u>	Copy the selected Patches to the Library clipboard
<u>Paste</u>	Paste the Patches in the clipboard to the current Library
<u>Clear</u>	Remove selected Patches from the Library
<u>Select All</u>	Select all entries in the Library for processing
<u>Open/Editor</u>	Open the Patch into an editor
<u>Open/SysX View</u>	Open the Patch into a SysX View Window
<u>Open/Build Bank</u>	Create a new Bank with the selected Patches
<u>Open/Update Library Settings..</u>	Open the Update Settings Dialog for the Library
<u>Open/Update Patch Parm...<u></u></u>	Open the Patch Info Dialog for the selected patch
<u>View/View Mode/Columns</u>	Display the library using the columns view
<u>View/View Mode/Patch Names</u>	Display the library using patch names only
<u>View/View Mode/Small Icons</u>	Display the library using small icons
<u>View/View Mode/Large Icons</u>	Display the library using large icons
<u>View/Patch Names Only</u>	Show the Library with names only (toggle)
<u>View/Setup Columns...<u></u></u>	Allow the order and width of the columns to be set
<u>View/Choose...<u></u></u>	Open a dialog to choose Patches based on given criteria
<u>View/Hide Selected</u>	Hide the selected Patches from view in the Library
<u>View/Show All</u>	Make any hidden Patches visible in the Library
<u>View/Delete Hidden</u>	Delete all patches which are currently hidden
<u>View/Swap Hidden/Displayed</u>	Swap the hidden/displayed state of patches in the library
<u>Keys/Sort</u>	Sort the Library by one of 3 methods
<u>Keys/Number Library</u>	Number the using one of the keyword columns
<u>Keys/Move Keywords...<u></u></u>	Move keywords from one column to another
<u>Keys/Merge Keywords</u>	Merge all keywords in all selected atches
<u>Keys/Relationships</u>	assign short forms to particular key words and sequences to keywords
<u>Keys/Apply Default Keys</u>	Process the selected patches to assign keywords
<u>Disk/Load Patch/Bank</u>	Load a Patch from disk into the Library
<u>Disk/Save Patch</u>	Save a Patch in the Library to disk as a separate file
<u>Disk/Link a Data File</u>	Link a Patch or Bank from disk into the Library
<u>Disk/Save Data File & Link</u>	Save a Patch to disk and link the Library to it
<u>Disk/Resolve Links</u>	Make any linked Patches into files stored in the Library
<u>Disk/Load All Banks&Patches</u>	Automatically build a library from existing banks and patches
<u>Randomize/Mix All</u>	Create a new Bank by randomly mixing all selected Patches
<u>Randomize/Blend</u>	Create a new Bank by mixing two Patches
<u>Randomize/Gen 4...<u></u></u>	Create a new Bank by taking portions of 4 Patches
<u>Randomize/Morph</u>	Create a new Bank by incrementally moving from one Patch to a second
<u>Receive from Instrument</u>	Load a Patch directly from the instrument into the Library

Move To...
Send Data to Instrument

Add the selected Patch to a different window
Send the first selected Patch to the instrument

Select All (DBase)

Purpose:

Select All is a quick way to select the entire DBase.

Directions:

Choose *DBase/Select All*.

See Also: [DBase Window](#)

Open Editor (DBase)

Purpose:

Open Editor opens any editable data file in the DBase. If the selected data file is a Bank, a Bank Editor window is opened to edit the data file. Otherwise, a Patch Editor Window is opened to edit the data file. For additional information on editing Banks and Patches, see the [Bank Editor Window](#) and the [Patch Editor Window](#).

To store the edited data file as a new entry in the DBase use the Move To option in the Bank Editor Window's Bank Edit menu or the Patch Editor's Sngl Edit menu to move the data file into the DBase.

Directions:

To edit a Patch or Bank from a DBase:

- (1) Select the data file to edit
- (2) Choose *DBase/Open/Editor*.

See Also: [DBase Window](#)

Update Settings (DBase)



Purpose:

Update Settings opens the Settings Dialog which is used to edit the basic settings for the selected DBase data file. Basic settings include: Comm Channel, MIDI channel, MIDI In Ports, MIDI Out Ports, Patch #, Cntrl #, and key words assigned to the data file. Each data file has its own set of basic settings which may be changed at any time.

For more information, you should see the [Settings Dialog](#).

Directions:

To change the basic setting for the selected data file:

- (1) Select a data file
- (2) Choose *DBase/Open/Update Settings...*
- (3) Make the desired changes to the basic settings

See also: [Settings Dialog](#)

See also: [DBase Window](#)

Open SysX View (DBase)

Purpose:

Open SysX View copies the selected Patch's SysX data into the SysX View Window. From there, the data can be viewed in its raw hexadecimal form. This feature is useful for viewing data while developing and testing instrument drivers in MIDI QUEST.

Directions:

To view the data file in hexadecimal format:

- (1) Select the data file to view
- (2) Choose *DBase/Open/SysX View*

See also: [SysX View Window](#)

See also: [DBase Window](#)

Sort (DBase)

Purpose:

Each DBase can be sorted by: Manufacturer, Instrument Type, File Size, Time, and File Name.

32-bit only: in list display mode to sort on any column, click on the column header and the datafiles in the Database will be sorted in ascending order based on that criteria.

Manufacturer sorts the DBase alphabetically by company name and instrument name (eg. Yamaha TG500). Sorting by manufacturer makes it is easy to see what data is stored for each instrument.

Instrument/Type sorts alphabetically by the type of data file. Instrument/Type sorts will show which types of data have been most frequently stored in the DBase.

File Size sorts the DBase by the size of the data file (smallest to largest). File Size sort shows at a glance the whether the DBase is biased towards Patches or Banks.

Time sorts the data files by the date they were created or last modified. Time sort is useful for determining which data files were most recently modified and which have remained unchanged the longest.

File Name sorts the data files alphabetically by file name. If you are hunting for a particular data file and just can't seem to find it, try sorting the DBase by name. You should be able to spot it more easily when the DBase is in alphabetical order.

Directions:

To perform a sort:

(1) Choose the type of sort from the *Data Base/Sort* menu.

See Also: [DBase Window](#)

Load Data File (DBase)

Purpose:

Load Data Files loads one or more data files that have already been saved to disk into a DBase.

If a new DBase is being built, some of the files required may be currently stored on disk. Instead of opening the files as editors and transferring them into the DBase, the files can be loaded directly in from disk using Load Data File.

In some cases, data files for certain instruments are not editable. Because these files can not be opened as editors, there is no means available to transfer the data into the DBase from within the program. Use this feature to load the data directly from disk.

Directions:

To load one or more data files from disk into a DBase:

- (1) Choose *DBase/Disk/Load Data File*
- (2) Use the File Selector to select the data files you wish to load
- (3) Hold down the CTRL key to select multiple files then press the OK button

See Also: [DBase Window](#)

Save Data File (DBase)

Purpose:

Save Data File saves a DBase data file to disk.

This is similar to [Save Data File and Link](#). Both functions save the selected data file to disk, however, with Save Data File and Link the data file in the DBase becomes a linked disk file. The data file itself is no longer stored in the DBase.

Directions:

To save a DBase data file to disk:

- (1) Select the data file you wish to save
- (2) Choose *DBase/Disk/Save Data File*
- (3) Use the File Selector to name and save the data file

See Also: [DBase Window](#)

Receive From Instrument (DBase)



Purpose:

Receive From Instrument and the Recv button loads data files from one or more instruments directly into an existing DBase.

Select from the [Driver List](#) the drivers for the data files that need to be loaded in the DBase. Next, return to the DBase and press the Recv button. The DBase will load a data file for each driver you selected from the Driver List.

This feature allows a complete or partial system snapshot to be taken. One of the most useful organizational features of the DBase is that it can be used to collect and store all of the settings from all of the instruments in your MIDI system. Effectively, a MIDI snapshot. This allows all of the various instrument data files for a piece of music to be stored together in one place.

Directions:

To add data files from one or more instruments to a DBase:

- (1) Select one or more drivers from the Driver List
- (2) Select the `Recv` button or *DBase / Receive From Instrument*
- (3) The program will load data files into the DBase for each driver selected from the Driver List

See Also: [DBase Window](#)

Cut (Library)



Purpose:

Cut is one of two methods of placing Patches into the Library Clipboard. Copy is the other.

Cut removes the selected data files from the Library and attaches them to the Library Clipboard. The data files are now available to paste into the same or a different DBase.

This function is useful for removing Patches from a Library where they are no longer necessary or for moving them to a different Library.

Note: Before adding the data files to the Library Clipboard, the program deletes any data files currently stored there.

Directions:

To cut Patches and place them in the Library clipboard:

- (1) Select the Library Patches to cut
- (2) Choose *Library/Cut*

See also: [Library Window](#)

Copy (Library)



Purpose:

Copy is one of two methods of placing Patches into the Library Clipboard. Copy makes a duplicate of the selected Library Patches. It attaches the copies to the Library Clipboard so that they may be pasted into a Library.

Copy provides an easy way to duplicate a Patch before editing the copy. If you wish to create a new Patch based on an existing Patch, copy the Patch to the Library Clipboard using *Library/Copy*. Next, use *Library/Paste* to paste the copy of the Patch back into the Library. The copied Patch may now be edited without destroying the original.

Note: Before adding the data files to the Library Clipboard, the program deletes any Patches currently stored there.

Directions:

To copy Patches and place it in the Library Clipboard:

- (1) Select one or more Patches to copy
- (2) Choose *Library/Copy*

See also: [Library Window](#)

Paste (Library)



Purpose:

Paste is used to add Library Clipboard Patches into the active Library.

Before Patches can be pasted into a Library, Patches from the current Library or another Library must be Cut or Copied. This places the Patches in the Library Clipboard. Once in the clipboard, the patches can be pasted to the same Library or a different one.

When moving Patches from one Library to another, the Libraries must contain the same type of data. For example, an M1 Patch can not be cut or copied to the Library Clipboard and then pasted into a D-10 Tone Library.

Directions:

To paste the contents of the Library clipboard into a Library:

- (1) Cut or Copy Patches from the current library or another library
- (2) Activate the Library window to receive the Patches
- (3) Choose *Library/Paste*

See also: [Library Window](#)

Clear (Library)

Purpose:

Clear is used to remove entries from a Library which are no longer needed. For example, maybe you replaced the trashy piano you were with a great new Grand Piano Patch. You know you will never use the old piano sound again. You can remove the old piano sound by selecting it and choosing *Library/Clear*.

Use this function with care. Remember, the Library is intended to be a resource area where all of your Patches are kept. Consider carefully before removing the Patch since it can not be retrieved.

Warning: when data is cleared from a Library it is permanent, the files can not be retrieved.

Directions:

To clear an unwanted Patch(es) from a Library:

- (1) Select Patch(es) you wish to clear
- (2) Choose *Library/Clear*

See also: [Library Window](#)

Select All (Library)

Purpose:

Select All is a quick way to select the entire Library.

Directions:

To select all Patches in a Library:

(1) Choose *Library/Select All*.

Open Editor (Library)

Purpose:

Any Patch for which there is a Patch Editor can be edited from the Library. For additional information on editing Patches, see the [Patch Editor Window](#).

To store the edited Patch as a new entry in the Library, first give the Patch a new name. Then use [Move To](#) in the Patch Editor's Sngl Edit menu to add the Patch to the Library.

Directions:

To edit a Patch from a Library:

- (1) Select the Patch to edit
- (2) Choose *Library/Open/Editor*

See also: [Library Window](#)

Print XReference (Bank)

Purpose:

Print XReference should only be used with Multi Banks. These are Banks containing Patches which contain references to sound Patches in a sound Bank. An example of this would be a Korg Combi Bank. Each Combi in the Combi Bank has references to 8 Patches in a Patch Bank.

As of the release of this software, Midi Quest and Solo Quest are the only Editor/Librarian packages that intuitively recognize which Patch locations in a sound Bank are not already referenced by a Multi and therefore are available for use by a new Patch.

You will love this new feature as when selected, it looks at each Multi Patch in the Bank and determines which sound Patches each Multi Patch is referencing. This information is then printed out on the currently selected default printer.

Note: Print XReference is only available in Midi Quest and Solo Quest.

Directions:

(1) Choose *Bank Edit/Print XReference*.

Open SysX View (Library)

Purpose:

Use this menu item to move the Patch's SysX data into the SysX View Window. From there, the data can be viewed in its raw hexadecimal form. This feature is useful for viewing data while developing and testing instrument drivers in MIDI QUEST.

Directions:

To view data in hexadecimal format:

- (1) Select the Patch to view
- (2) Choose *Library/Open/SysX View Window*

See Also: [SysX View Window](#)

See Also: [Library Window](#)

Build Bank (Library)



Purpose:

Build Bank creates a new Patch Bank using Patches selected in a Library. If you want to send more than one patch at a time to your instrument, you must do so by sending a bank. With this feature you can build a bank using Patches you have selected in the Library.

The Library is ideal for storing all of an instrument's Patches. Use it to find the sound that best suits a particular part in a piece. Once the Patches for the piece have been chosen, they can be selected and built into a Bank. The Bank can then be transferred into the instrument's permanent memory whenever necessary. Your sequencer will then be able to use Patch change commands to access the Patches. The newly created Bank should be added to the DBase containing the music's other data files.

Patches are added to the Bank in the order they are selected. Only as many Patches as will fit in a Bank will be added even if more are selected. If there are fewer Patches selected, the remaining entries in the Bank are left empty.

When a bank is moved into a Library, both the name of the bank and the patches position in the bank are stored in keys 8 and 7 respectively. With this information, you could later rebuild the bank by using the Choose function to hide all patches but those of a specific bank and then sorting the patches based on the number in key 7.

Note: in order for the Bank to be created, the Bank driver which corresponds to the Library Patches must be in the Driver List. For instance, an M1 Patch Library would require the M1 Patch Bank driver to be in the Driver List.

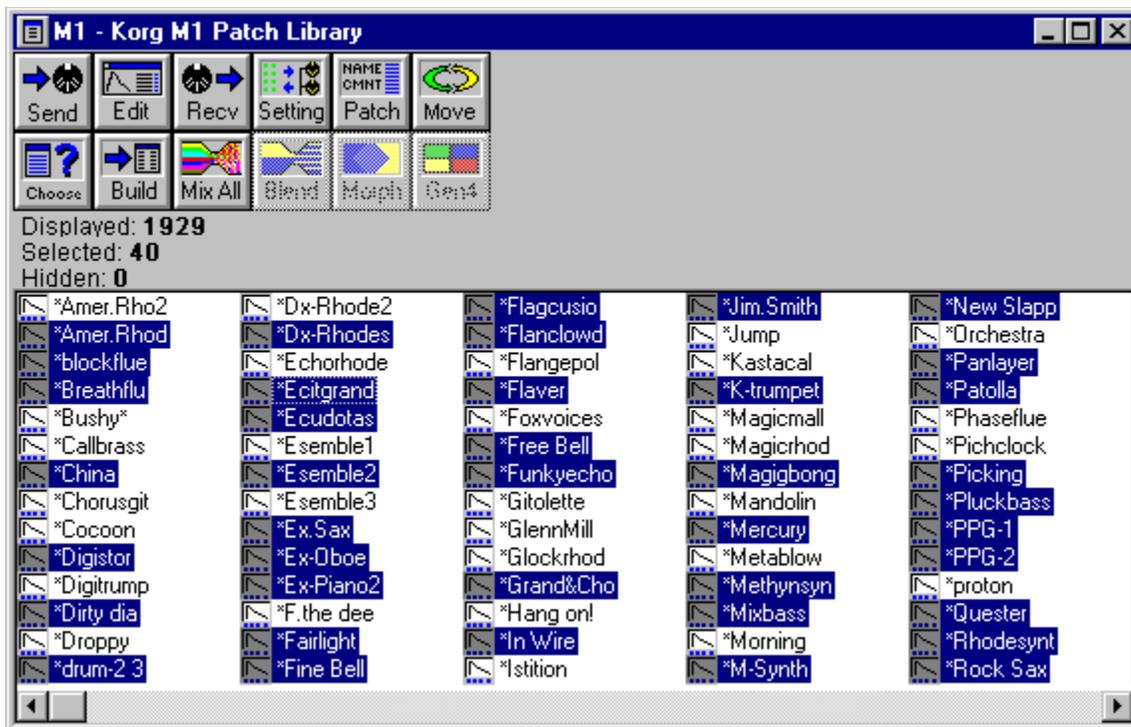
Directions:

To build a Bank from a Library of Patches:

- (1) Select all of the Patches destined for the Bank
- (2) Choose *Library/Open/Build Bank*

Example:

- (1) Select the Patches to place in the new Bank



(2) Choose *Library/Open/Build Bank* and the following Bank is created:



See also: [Library Window](#)

Update Settings (Library)



Purpose:

Update Settings opens the Settings Dialog which is used to edit the basic settings for all of the Library's Patches. Basic settings include: Comm Channel, MIDI channel, MIDI In Ports, MIDI Out Ports, Patch #, Cntrl #, and key words assigned globally to the Library. Each Library has its own set of basic settings which may be changed at any time.

These basic settings are global to **all** of the Patches in the Library. While it is possible to assign keys in this window, the program currently makes no use of them.

For more information, you should see the [Settings Dialog](#).

Directions:

To change the Basic settings for the Library:

- (1) choose the 'Settings' button or *Library/Open/Update Settings...*
- (2) Make any necessary changes to these settings

See also: [Settings Dialog](#)

See also: [Library Window](#)

Choose... (Library)



Purpose:

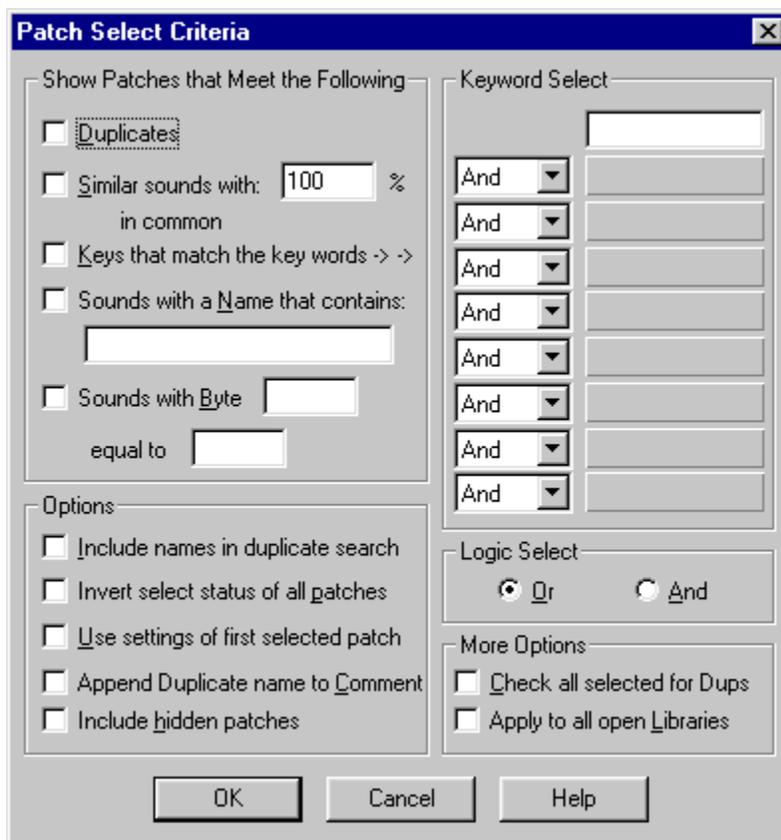
Choose opens a dialog that allows Patches in the Library to be chosen based on criteria set in the dialog. The criteria include: duplicates, similar sounds, name, keys, and byte value. Patches which meet the criteria are displayed in the Library Window. All Patches which do not meet the criteria are hidden from view.

The 32-bit program version will execute the Choose function in a separate thread. This makes it possible to continue working while the Choose function executes.

Directions:

(1) Select the 'Choose' button or choose *Library/View/Choose...*

The Choose Dialog



The Choose dialog is divided into four sections. Each section controls the way that the program searches for Patches within the Library. The sections are: Show Patches that meet the following: (the Basic Criteria), Options, Keyword Select, and Logic Select. Only Patches which meet the all criteria in the dialog are subsequently shown in the Library. All other Patches are hidden from view.

Show Patches that Meet the following: (the Basic Criteria)

The Basic Criteria fields are used to select which tests are performed on the library. There are five different tests. Each will be explained below. Any combination of the tests may be performed on the Library at one time. The Patches must meet all of the criteria in order to be displayed.

Duplicates

If 'Duplicates' is chosen, the program will compare every Patch with every other Patch in the Library. If any of the Patches are identical, one of them will be flagged for display. The purpose of the function is to weed out duplicates which have been placed in the Library either intentionally or unintentionally.

After the search is complete, the program will also open a DISPLAY ONLY window which shows the names of the Patches that are identical along with the filename and date information. This window is for information purposes only.

Similar Sounds with ____% in common

If this option is selected, the program will search for the Patches which have the given percentage of parameters in common. The purpose of this test is to test and weed out Patches which are so similar that there may be no auditory difference between the two. Remember that if Duplicates and Similar Patches are chosen at the same time, duplicates will over-ride and only those Patches which match exactly will be shown.

Keys that match the Keywords

If this option is selected, the program will compare the selected keywords in the Choose Dialog against the keywords assigned to the Patch. If the criteria are met, the Patch will be displayed. Keywords are words to describe and quantify the sound and should be one of the most commonly used criteria. This function allows you to quickly search and find Patches that meet a given criteria. Maybe you are looking for a BASS sound or more specifically a BASS sound with a SLAP quality. If you choose the Keys option then you

This function is extremely useful for finding Patches which have certain aural qualities when you are searching for a particular type of sound.

Remember that for this option to work, you must have previously assigned keys to your Patches. While this may seem like a lot of up front work, in the long run it can save you a substantial amount of time when you start looking for a particular type of sound. These descriptions are intended to be general so you can find a certain type of Patch in the future. Try not to get too fancy with your choice of keywords or you may never successfully search and find a sound. Above all, be consistent. in your use of keys. When you add a key, ensure that there is not already a synonym for the new key you plan to add.

Sounds with a name that matches: _____

If this option is selected, the program will search the Library for Patches that have a partial or full match with the text entered here. The test is context sensitive so Brass is considered to be different from BRASS as far as the program is concerned. This function is useful for finding Patches which have similar names. Maybe you want to find all of the Patches with the word STRING in them.

Sounds with Byte X equal to Y

If this option is selected, a technical test takes place to show only those Patches where byte X currently has the value Y. You will need to be intimately familiar with your instrument's SysX in order to make use of this option. For example on a DX7 if you are looking for all of the Patches which use an algorithm of 8 and you knew that the algorithm is stored in byte 134. You could set X = 134 and Y = 7 and the program will find all of the Patches with the desired algorithm. This can be an extremely useful function if you have the technical know how.

Logic Select

This option can be set to either And or Or. It works in conjunction with the Criteria when more than one criteria is selected. If more than one criteria is selected and Or is chosen, then Patches will be shown in the Library if they meet one or more of the chosen criteria. If And is chosen, a Patch must meet ALL of the chosen criteria in order for it to be displayed.

Options and More Options

The Options block has five additional options which control how other tests are performed. Any of these options may be selected as desired.

Include Names in Duplicate Search

Normally when performing a Duplicates search, the program does not test the name as the intent is to find Patches which are identical in sound even if the name is different. Select this option to have the program treat Patches with different names as different Patches.

Invert Select status of all Patches

This is the equivalent of a global NOT command. When this is checked, the program will display all of the Patches which did NOT meet the criteria you set out.

Use Settings of first selected Patch

This function came about as a testing method in which the program tested only one Patch against all others to see if there were any duplicates. If this option is checked, you should have preselected one Patch in the Library. The program will then use the parameters of that Patch for its tests instead of the information entered in the dialog window.

Append Duplicate Name to Comment

When a duplicate search is run, the intent is to remove patches which are duplicates. This function adds the name of the duplicate patch to the comment area of the patch that is retained. This provides a reminder of what other patches sound identical to the currently selected one.

Include Hidden Patches

Frequently you will be performing multiple searches one after the other. When checked, this option automatically adds any hidden patches into the list so they may be searched as well. This saves you from having to choose *Library/View/Show All* before each use of the Choose dialog.

Check All Selected for Dups

When checked, when the Duplicate check function is selected, the program will compare only selected patches against the rest of the library. This is much faster than cross checking all of the patches in the

library and is useful when you wish to check only a specific group of patches

[Apply to all open Libraries](#)

When checked, the software not only applies your options to the current library but to all other open libraries as well. This method is ideal for finding, say, all of your Bass patches in all of your instruments. Just open the appropriate libraries and perform a keyword search for "Bass". Each library will now only list the bass sounds.

Keyword Select

The final area of the dialog is the Keyword Select. The information in this area will only be used if the [keys that match the key words](#) criteria option is selected. Click on each of the raised boxes to select a keyword that you wish to use in order to find particular Patches. Remember, that for this feature to work you must have already coded each of your Patches with keywords. There is also space to enter one keyword by hand if you find this faster.

For example, if you wish to find all of the Patches that are bowed strings so that you can audition them and in your Library you have coded each of these Patches with "String-Bow" then check "keys that match the key words" and click on the first raised box to display the keywords. Choose the keyword "String-Bow" and finally press the OK button. The program will search through the Library and show only those with the "String-Bow" keyword.

Remember that the order of the keywords is not important. The program will check each of test keywords against those assigned to the Patch.

See also: [Library Window](#)

Hide Selected (Library)

Purpose:

Hide Selected hides Patches from view without actually removing them from the Library. This function is ideal if you are only interested in looking at a portion of a Library.

Hide Selected is also used by the Choose Dialog. Patches that do not match the selected criteria are hidden while those that do are displayed.

Directions:

- (1) Select the Patches to hide from view
- (2) Choose *Library/View/Hide Selected*

See also: [Show All](#)

See also: [Library Window](#)

Show All (Library)

Purpose:

Show All displays any Patches that you have hidden by using Hide Selected or the Choose dialog. If any Patches in the Library have been hidden, the 'Hidden' indicator will show greater than 0. To show the hidden Patches choose 'Show All'. The Patches will be added to the bottom of the Library.

Directions:

(1) Choose *Library/View/Show All*

See also: Library Window

Sort... (Library)

Purpose:

Opens the Sort Dialog to select the criteria to sort the Library with.

32-bit only: in list display mode to sort on any column, click on the column header and the datafiles in the Library will be sorted in ascending order based on that criteria.

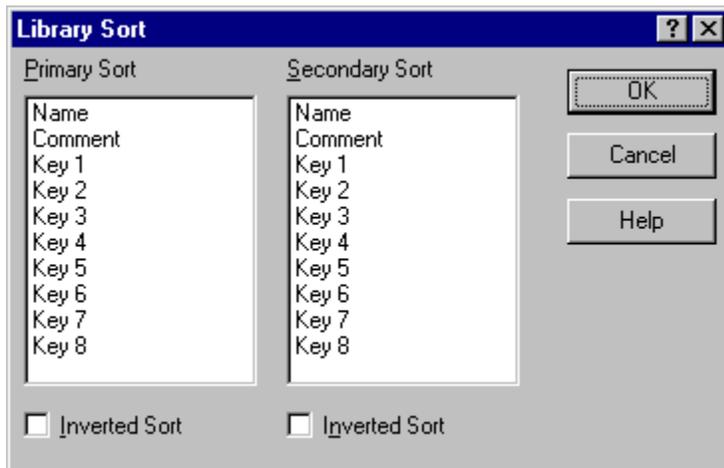
32-bit only: The 32-bit program version will execute the Sort function in a separate thread. This makes it possible to continue working while the Sort function executes.

Directions:

To sort the Library:

- (1) Choose the type of sort from the *Library/Sort by/Number / Sort...* menu
- (2) Use the Sort Dialog to set up the sort
- (3) Press the OK button to run the sort

The Sort Dialog



The Sort Dialog allows you to set up a two level sort selecting from ten different criteria (Name, Comment, and Keys 1 through 8). The sorting is performed alphabetically but you can choose inverse order as well by checking "Inverted Sort". Offering a Secondary sort means that if there are a number of patches which have the identical primary sort value, the Secondary sort parameter will be used to order the items. If you do not wish to use a Secondary Sort, don't select an item from the list.

To perform the sort, press the OK button or to cancel, press the Cancel button.

See also: [Library Window](#)

Load A Patch/Bank (Library)

Purpose:

Load a Patch/Bank adds data files that have already been saved to disk to a Library.

If a new Library is being built, some of the files required may be stored on disk. Instead of opening the disk files as editors and transferring them into the Library, the files can be loaded directly in from disk using Load Patch/Bank. When loading from disk, any combination of Patches and Banks may be selected but they must be of the same type as the Library or they will not load. If a Bank is selected, each Patch in the Bank will be separated and loaded into the Library.

In some cases, data files for certain instruments are not editable. Because these files can not be opened as editors, there is no means available to transfer the data into the Library from within the program. This feature provides this capability by loading the data directly from disk.

Directions:

To load a multiple Patches and Banks from disk into a Library:

- (1) Choose *Library/Disk/Load Patch*
- (2) Use the File Selector to select one or more Patches or Banks to load
- (3) To be loaded into the Library, the data file(s) must contain the same type of data as the Library.

See Also: [Library Window](#)

Save Patch (Library)

Purpose:

Save Patch saves a Library Patch to disk.

This is similar to [Save Patch and Link](#). Both function save the selected data file to disk, however, with Save Patch and Link the data file in the Library becomes a linked disk file. The data file itself is no longer stored in the Library.

Directions:

To save one Library Patch to disk:

- (1) Select the Patch to save
- (2) Choose *Library/Disk/Save Patch*
- (3) Use the File Selector to name and save the Patch

See also: [Library Window](#)

Mix All (Library)



Purpose:

Mix All randomly picks parameters from all selected Patches and creates a new Library of 32 Patches.

The probability of the parameter coming from each of the selected Patches is equal. Therefore, if 2 Patches are selected, the chance is 50% that the parameter value will come from either Patch. If four Patches are selected, there is a 25% chance that the parameter will be selected from each Patch, and so on.

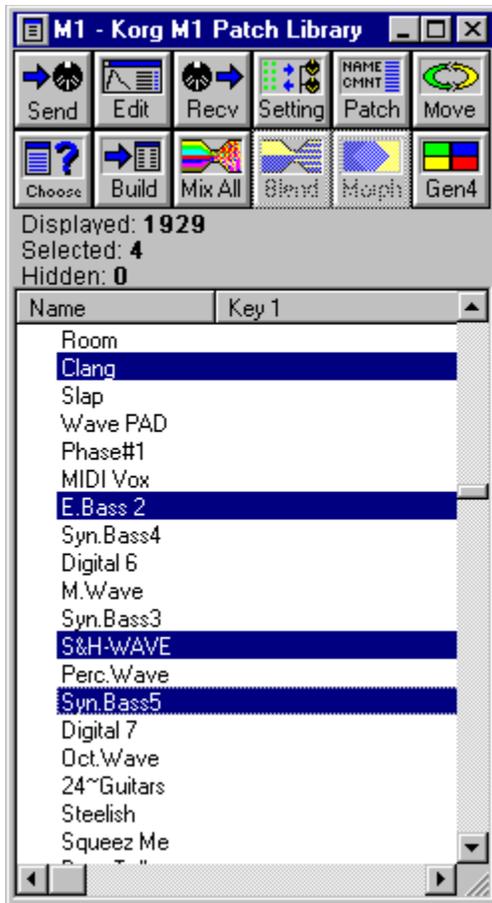
Directions:

To Mix Patches:

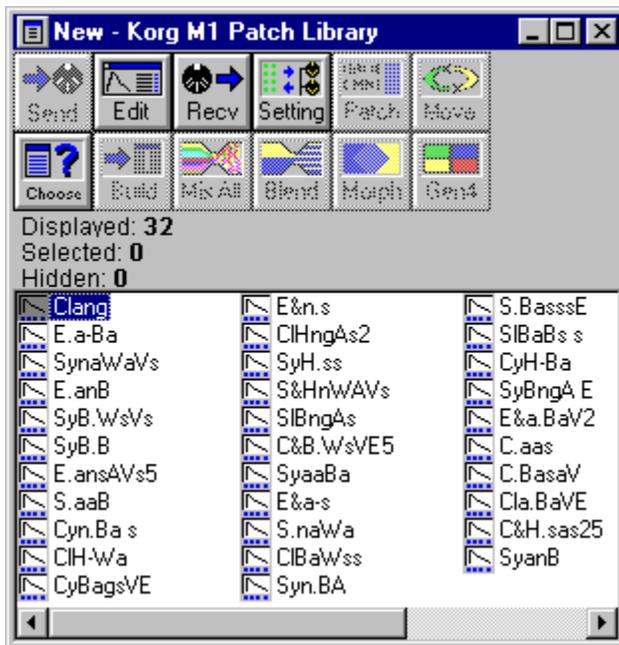
- (1) Select the Patches you wish to mix
- (2) Choose the 'Mix All' button or *Library/Randomize/Mix All*
- (3) Why not try auditioning your new Patches?

Example:

- (1) Select two or more Patches to mix



- (2) Choose the Mix button
- (3) A new Library opens containing 32 new Patches



See also: [Library Window](#)

Receive from Instrument (Library)



Purpose:

Recv loads a Patch from an instrument directly into an existing Library.

In most instances, the Library will load the Patch which is currently in use by the instrument.

In some instances due to instrument limitations, you can not load the current Patch and must choose a Patch from a specific memory location in the instrument. If you find that you are always loading the same Patch regardless of which Patch is selected in the instrument, your instrument probably falls into this category. In this case, you should follow these steps to load a particular Patch:

- (1) Select the 'Settings' button to display the Settings dialog.
- (2) In the lower left corner of the dialog will be the 'Patch #' parameter.
- (3) Use this parameter to select the Patch to load
- (4) Press the OK button to return to the Library.
- (5) Select the 'Recv' button or choose *Library/Receive from Instrument*

Directions:

To load a Patch directly from the instrument and add it to the Library:

- (1) On the instrument, select the Patch to load (if appropriate)
- (2) Select the 'Recv' button or choose *Library/Receive from Instrument*

See also: Library Window

Sending a Patch (Library)



Purpose:

One of the primary functions of the Library is to send Patches to an instrument so they can be heard. After all, the Library will likely contain most of the Patches you have collected for your instrument.

If *Options/Properties/Library Tab/Audition* is set on, selecting a Patch will automatically send it to the instrument. The Patch may then be auditioned using the program's extensive auditioning utilities.

If *Options/Properties/Library Tab/Audition* is set off, then the Patch can be sent to the instrument manually by following the directions below.

Directions:

To send a Patch to the instrument manually:

- (1) Select the Patch to send
- (2) Select the 'Send' button or choose *Midi/Send Data File*

See also: [Library Window](#)

Common Errors

Common Errors are divided into four sections:

[Template Load Errors](#)

[MIDI Communication Errors](#)

[Disk Access Errors](#)

[Macro Errors](#)

[Sound Quest's .ini file format](#)

Template Load Errors (Common Errors)

There are four possible errors loading a template. If there is an error indicating that the file is not found:

(1) The editor template has not been installed

This error is possible if the editor template has not been installed, for instance, you obtain a data file from your friend and load it into the program. The file will load into a DBase or into a Bank Editor if it is a Bank, even if the instrument's drivers and templates have not been installed. However, any attempt to open an editor will fail because the necessary control files are not there.

(2) There is no editor file for the type of data selected

It is possible for an editor to not be on the disk for two reasons. First, the editor has been intentionally or unintentionally erased. Without the editor on the disk, an error will result

Alternately, a driver could have been developed and a Patch editor name assigned without an editor ever having been developed for the instrument. In this case, the program will look for the editor on the disk and then fail because the editor is again not there.

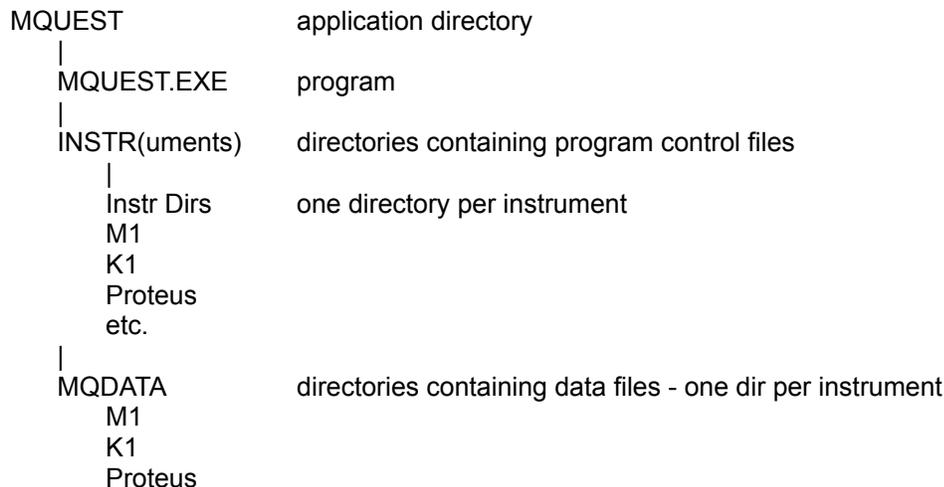
(3) The path to access the instrument editors is incorrect

When the program is installed, a file called SQ.INI is created in your windows directory. In this file is a path to the program's INSTR directory. If you have installed to the C: drive and used the suggested directory names, the path would be C:\MQUEST\INSTR. If the program is subsequently moved to a different drive or the installed directory renamed this path will become invalid. To correct this problem either reinstall the program so that the correct path is automatically created or edit the SQ.INI file with the NOTEPAD program included with windows and correct the path reference.

(4) There is insufficient memory available to load the editor

This error should be extremely rare under Windows. If it does occur you will need to free memory by closing other programs or closing some of the windows currently open in the program.

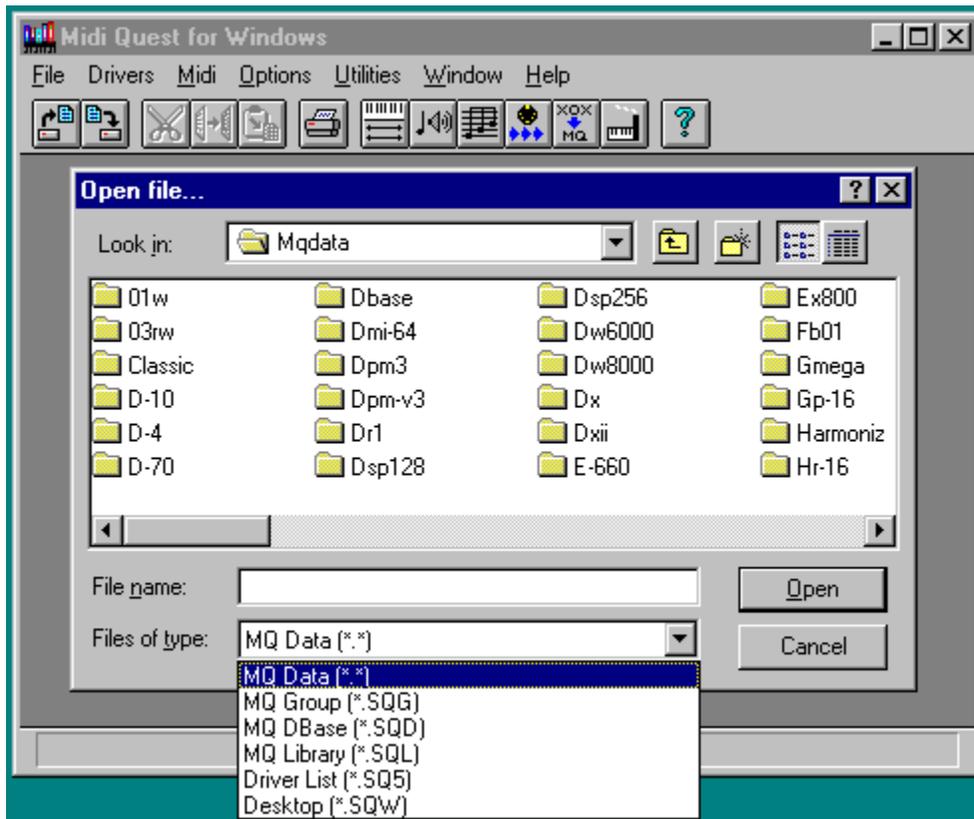
The program files must be located as follows:



etc.

See also: [Common Errors](#)

File Selector (Common Elements)



The File Selector is used to load, name, and save files. It is opened automatically after choosing *Files/Open...* and *Files/Save As...* along with data file load and save options in the Group, Library and DBase menus.

When *Files/Save As...* is selected from a Bank Editor, Patch Editor, Group, DBase, or Library Window, the File Selector dialog allows the data to be saved in a number of different formats depending on the type of data file. The format that the data is saved in is selected from the "Save File as Type..." in the lower left side of the dialog. When loading a file, use the similarly located "List Files of Type".

The following chart shows which alternate file formats can be saved by each type of editor:

<u>Window</u>	<u>Bank Editor</u>	<u>Patch Editor</u>	<u>Group/DBase Window</u>	<u>Library Window</u>
MID	Y	Y	Y	-
SYX	Y	Y	Y	-
TXT names	Y	-	-	Y
Full ASCII text	Y	Y	Y	-
INI Cake names v1/2	Y	-	-	-
INS Cake names v3	Y	-	-	-

MID = Standard MIDI File format

SYX = MIDIX Format - Cakewalk compatible

TXT = Only the name of the Patch is saved in ASCII text

ASCII = All SysX saved to disk as an ASCII text file
Cakewalk names - Names only in Cakewalk v 1.0/2.0 format
Cakewalk names v3 - Names only in Cakewalk v3.0 (and higher format)

.MID (Standard MIDI File) Format

Save your file in .MID format when the data is to be exported to a sequencer or other program which is capable of extracting SysX data from a Standard MIDI file.

.SYX (MIDIX) Format

Save the file in .SYX format when the data is to be exported to a sequencer or other program which is capable of loading MIDIX data. eg. Cakewalk.

When saving in .SYX format from a Bank, a dialog is displayed which gives you the option of two different saves. You can save the entire bank as a single MIDIX file, or you can save the selected patches in the bank and individual MIDIX files. If you opt to save selected patches individually, The patches are auto numbered in order to create unique files.

.TXT Format (1)

Save the names in .TXT format when you wish to import the names of a Bank or Library into a word processor for subsequent editing or formatting before printing.

When saving Patch names from a bank then after entering the name of the text file to use and pressing OK, you will be given the option of having the patch names separated by a <RETURN> in which case each patch will appear on a separate line in a word processor. Alternately, you can choose to have the data TAB separated. This is useful if you plan to import the names into a database. In this case, each name will fill a record entry.

When a bank is saved, the numeric patch numbers are also added after the names.

.TXT Format (2)

Save the entire SysX file in .TXT format. Useful when you would like to export the entire file in an ASCII legible format for inclusion or printing from another file.

.INI Format (Cakewalk v1.0/2.0)

The Windows version allows a Bank of Patch names to be written directly into the Cakewalk PATCHES.INI file so they can be viewed directly from within Cakewalk. The INI option will only work for Cakewalk v1.x and v2.x. To write the Patches into the .INI file, choose the .INI option, move into the WinCake directory, and double click on PATCHES.INI. The program will **add** the Bank's Patch Names into the file. It will **not** overwrite any other information in the file.

.INS Format (Cakewalk v3.0 and higher)

The Windows version also allows a Bank of Patch names to be written directly into the Cakewalk MASTER.INS file so they can be viewed directly from within Cakewalk. This will only work for Cakewalk

v3.0. To write to Cakewalk v1.x or 2.x, use the .INI format discussed above. Patch names can be written two different ways.

These are:

- 1) Writing Patch Names directly into the MASTER.INS file
 - a) choose the .INS option
 - b) using the file selector, move into Cakewalk's directory
 - c) double click on the MASTER.INS file
 - d) the program will **ADD** the Bank's Patch Names into the file. It will **NOT** overwrite any other information.
 - e) you now need to quit Cakewalk and rerun it then create an Instrument using the saved bank
 - f) It is important to note, that after adding the bank of names, you must create the instrument yourself and associate the names with the instrument yourself. If you would like the entire instrument created, please follow the instructions below

- 2) Writing Patch Names into a separate .INS file
 - a) choose the .INS option
 - b) enter a name for the file and press OK
 - c) once the file is saved, for older versions of Cakewalk you must use a text editor to integrate the file into the MASTER.INS file
 - d) there should be instructions in the Cakewalk manual for accomplishing this
 - e) If you have a copy of Cakewalk v9.0 Cakewalk can automatically create an instrument for you. In Cakewalk, follow these steps:
 - i) Choose *Options/Instruments*
 - ii) Click the **Define** button in the **Assign Instruments** dialog
 - iii) Click the **Import...** button in the **Define Instruments and Names** dialog
 - iv) Select the .ins file you just created in Midi Quest and load the instrument

Midi Communication Errors (Common Errors)

These errors are all generated while attempting to either transmit or receive System Exclusive information. Before attempting to deal with the particular error, you should verify this information:

1. Communications channels for data and instrument match.
2. SysX communication is ENABLED for the instrument.
(It is possible to shut off SysX on many instruments)
3. Check that the Patch Bay is properly mapped to the instrument.
...When first starting with the program we strongly suggest bypassing the Patch bay
4. The Patch Bay is NOT routing SysX output from the computer back to the computer
5. All requirements & warnings in Fast Tips have been met.
6. Connect the instrument DIRECTLY to the computer.
(ie. No merge boxes, mappers, Patch bays, etc. are connected)
7. Retry the communication.

If after checking all of these possibilities an error is still being generated, it will be necessary to check out the particular error. The "Possible Causes" listed below will not include those already covered!

ERROR! SysX Missing Bytes!

This is probably the most common error and can result from a number of different problems including those unique to a particular computer.

Note: If you get this error after pressing the OK button while the number of bytes received in the MIDI I/O Window is 0, it is not a true SysX Missing Bytes error. The error should pop up by itself after the program has requested data from the instrument and the instrument has transmitted data back to the computer.

On Windows systems, this problem can be caused by one of a number of possibilities:

1) You are using old MIDI drivers for your interface which do not receive SysX properly. Contact your MIDI card manufacturer to ensure that your drivers are up to date. (Particularly for Sound Blaster, Voyetra, MidiMan, and CMS cards.)

2) You have two drivers installed in your Windows Driver List which are both attempting to access the same MIDI interface. This is guaranteed to cause problems. Remove one of the drivers from your Windows MIDI Driver List and reboot your system. For example, you may have installed both the Music Quest MIDI drivers and the Roland MPU401 drivers for your Music Quest MIDI interface. This should not be done and you should remove the Roland drivers which are inferior.

3) Most 386SX systems do not run fast enough to receive large SysX dumps in Windows 386 mode. Run Windows in Standard (Win /s) and this should cure your problems. An additional suggestion along the same vein is that on some PC systems we have seen improved interface performance on interrupts other than 2/9. Try setting up your interface on interrupts 5 or 7 and see if this improves performance.

4) A less common cause of this problem on PC systems is that there are two pieces of hardware attempting to use the same piece of hardware. The most common example of this is an MPU interface and a bus mouse both attempting to use interrupt 2 (the default for both pieces of hardware). You must change the interrupt on one of the two pieces (preferably the mouse).

5) A second cause of this problem is occasionally that the data inside your instrument is corrupt (damaged) and so it is not transmitted correctly. The computer will see this as an incorrect transmission. To test this problem, perform a complete memory reset on your instrument if this is possible and try the

dump again. Alternately, attempt to download different types of data from the instrument to determine which data is corrupt. For instance, you may be able to save all of your Patches individually even though the Bank will not download as a whole.

ERROR! No EOF transmission found!

This error is a result of the program receiving a MIDI dump which is larger than expected. This error usually occurs when the program requires the dump to be manually transmitted from the instrument and the wrong type of data is transmitted.

The error may also result from ROM upgrades made to an instrument. In this case the data transmitted may not match the format for which the program was tested. This usually requires a ROM upgrade to the instrument.

Finally, this error is also displayed when program has not received any data and "Finish" is selected in the MIDI I/O Window.

ERROR! Incorrect Company ID!

This error results from incorrect data being transmitted to the program. Check your system to ensure the correct data is sent.

See also: [Common Errors](#)

Disk Access Errors (Common Errors)

All Disk Access Errors are generated while attempting to save or read data from disk.

File NOT Found!

This error results when attempting to load a file from disk which does not exist in the selected directory. To correct the problem:

- (1) verify that the current directory actually exists
- (2) verify that the file is in the selected directory
- (3) verify that the name of the file is correct.

File READ Error!

A File READ Error only results when the selected file is corrupted. Save your current work and use a disk recovery to attempt to recover your work.

Note: you should always backup your work onto a separate disk to avoid losing valuable work because of hardware failure.

Not an IFF File!

Not a Sound Quest Data File!

Not a Sound Quest DBase File!

Not a Sound Quest Library File!

This error occurs when File/Open... is used to select a file which does not match the type selected in "List Files of Type" in the lower left of the File Selector. To load a DBase, ensure you select the file type DBase. To load a Library, ensure that the file type Library is selected. Alternately, it is possible that the file being loaded is not a Sound Quest file and so will fail to load.

Not an IFF Driver File!

The selected file does not contain a driver for a Sound Quest "QUEST" product. Select a Sound Quest driver file.

Directory NOT Found!

This error occurs when the path entered the File Selector's "Path" does not exist. To correct this, enter a valid path in the selector.

File WRITE Error!

This error will only occur if there is insufficient space on the disk to save the file or the disk is damaged. Save the data on another disk.

File CREATION Error!

This error can be caused by a number of different problems:

- (1) the name of the file is unacceptable to the computer's DOS
- (2) the data is being saved to a directory which does not exist
- (3) the file being written to has been given READ ONLY status

Determine which of the problems it is and correct it.

See also: [Common Errors](#)

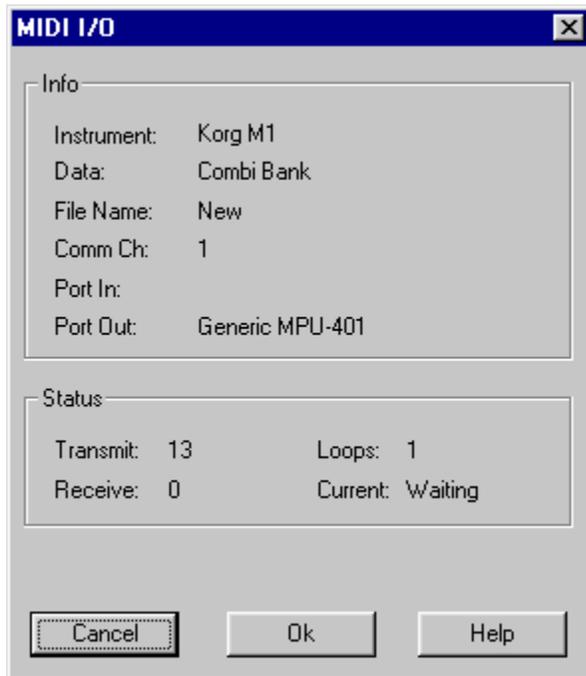
Midi Menu (Common Elements)

Purpose:

This menu controls sending MIDI data from the program to the instruments. The menu contains the following options:

<u>Current Data To Instrument</u>	Send the Data File in the active window to the instrument
<u>Current DBase to Instrument</u>	Send the DBase in the active window to the instruments
<u>Data from Disk to Instrument</u>	Send a Data File from disk to an instrument
<u>DBase from Disk to Instrument</u>	Send a DBase from disk to an instrument
<u>MIDIx from Disk to Instrument</u>	Send a MidiX format file to an instrument
<u>Play Chord</u>	Play the chord defined in the Tones Window
<u>Play Sequence</u>	Play the sequence defined in the Tones Window
<u>Play Last Note</u>	Play the last note played
<u>Play Note</u>	Play one of the first 8 notes in the Tones Window
<u>Midi In Ports</u>	Choose the Midi In Ports available to the program
<u>Midi Out Ports</u>	Choose the Midi Out Ports available to the program

Midi I/O Window (Common Elements)



The MIDI I/O Window automatically opens during computer/instrument SysX transfer. The window displays the following information:

Make & Model - the type of data being received or transmitted

File Name - the name of the file being transmitted

Comm Ch - the Communication channel being used to transfer the data

Port In - the MIDI Port to receive the data

Port Out - the MIDI Port to transmit the data

Transmit - # of bytes transmitted and # of times data transmission has occurred

Receive - # of bytes in the last data reception

Wait - indicates that MIDI QUEST is waiting to receive data

Cancel - select to cancel the current operation

OK - select OK when manual transmission of instrument data is complete

In most cases, the program will send the instrument a message requesting the data from it and the instrument will return the requested data. If it is necessary for you to transmit the data from the instrument manually, it will be indicated in the Fast Tips help.

If the program sits in "Wait" for a significant period of time and you are not required to manually transmit the SysX data, there has probably been some sort of MIDI communication error. Read the Fast Tips file for the instrument Driver you are using, double check your MIDI cables, ensure that the ports and communication channel are correctly set, and try again. If you continue to have problems, check the Common Errors/[Midi Communication Errors](#) portion of the manual along with the following:

See Also: [Midi In Ports](#)

See Also: [Midi Out Ports](#)

See Also: [Settings Dialog](#)

Common Elements

Common Elements are the global elements of the program. They include: the File Selector which is used to load and save files from disk; the MIDI I/O Window which is used while transferring data to and from instruments; the Move To dialog which is used to move data files from one window to another; the Drag and Drop facilities which also move data files between windows, and the options in five menus:

Files

Midi

Options

Utilities

Windows

Files Menu (Common Elements)

The File Menu contains the options normally found in a program. They include the following options:

<u>New</u>	Create a new Data File, DBase, Library, or Driver List
<u>Open</u>	Open an existing file from disk
<u>Close</u>	Close the active window
<u>Save</u>	Save the contents of the active window
<u>Save As...</u>	Save the contents of the active window with a new name
<u>Print</u>	Print the contents of the active window
<u>Print Setup</u>	Select the printer to print to
<u>Save Desktop As...</u>	Save a list of all of the currently open windows for future loading
<u>MRU</u>	Most Recently Used file list - quickly load a recently used file
<u>Exiting</u>	Exit the program

See also: [Common Elements](#)

New (Files Menu)

Purpose:

New opens a new window of the selected type. The menu item contains a sub-menu which allows one of four types to be selected: Group/Editor, Library, DBase, and Driver List.

Group/Editor - this option is enabled if the Driver List Window is open and a new Group or Editor can be created for the selected Instrument Bar or Driver. If an Instrument bar is selected then a new Group is created, otherwise, an "Editor" for the selected driver is displayed.

Library - this option is enabled if the Driver List Window is open and a data type for which a Library can be created is selected.

DBase - this option creates a new DBase.

Driver List - this option opens a Driver List if there is no Driver List open. Alternately, it clears the contents of the current Driver List.

Directions:

- (1) Choose *Files/New*
- (2) Choose the type of data to create

See Also: [Editor Window](#)

See Also: [Bank Window](#)

See Also: [Group Window](#)

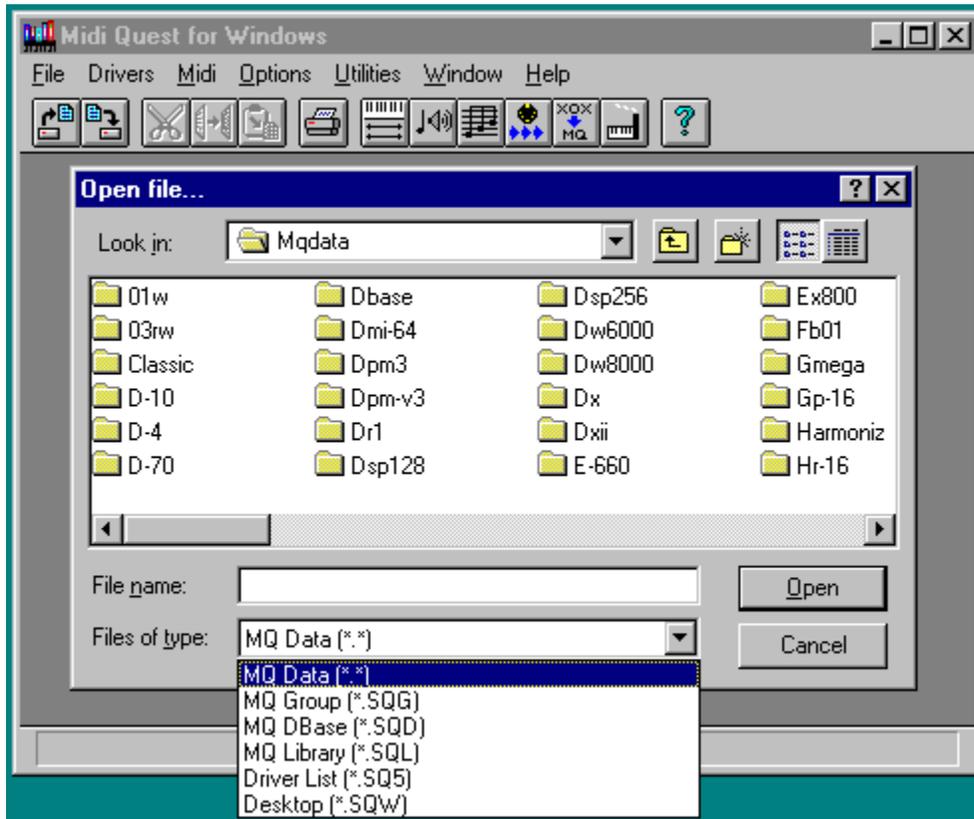
See Also: [DBase Window](#)

See Also: [Library Window](#)

See Also: [Files Menu](#)

See Also: [Common Elements](#)

Open (Files Menu)



Purpose:

Open displays a File Selector to open a disk file. The file type can be a data file (Patch or Bank), DBase, Group, Library, Driver List or Desktop. If the standard Sound Quest extension is not used for the file type, the correct type must be chosen in the "List Files of Type" selector in the lower left corner of the [File Selector Dialog](#). Failure to do so will result in an error when loading the file.

The extensions for files are as follows: Groups (SQG), Libraries (.SQL), DBases (.SQD), Driver List (.SQ5), any other extension is considered to be a Bank or single data file.

Directions:

- (1) Select the 'Open' button or choose *Files/Open...*
- (2) Select the type of data to load using the "List Files of Type" selector (if necessary)
- (3) Choose the file to load.

See Also: [Files Menu](#)

Save (Files Menu)



Purpose:

Save saves the contents of the active window and overwrites the current file on disk.

If the data in the active window has a file name assigned, the current changes will be saved to disk overwriting the existing file. If the data in the window has not yet been named, the program will display a File Selector Dialog to assign one.

Depending on the type of window that is currently active, this menu item will read: *Save Drv List*, *Save Editor*, *Save Bank*, *Save DBase*, *Save Library*, or *Save SysX View*.

Directions:

(1) Select the 'Save' button or choose *Files/Save*

See Also: [Files Menu](#)

Save As (Files Menu)

Purpose:

Use Save As... to assign a new name and/or directory to the data in the active window

Depending on the type of window that is currently active, this entry will read: *Save Drv List As...*, *Save Editor As...*, *Save Bank As...*, *Save DBase As...*, *Save Library As...*, *Save Group As...* or *Save SysX View As...*

For more information on saving data in various file formats including those for **Cakewalk, MIDIX and Standard MIDI file**, please see: [File Selector Dialog](#).

Directions:

- (1) Choose *Files/Save As...*
- (2) Use the [File Selector Dialog](#) to select the directory and name of the file
- (3) If necessary, use the "Save File as Type" list to choose the format of the file

See Also: [Files Menu](#)

Close (Files Menu)

Purpose:

Close closes the active window. If the contents of the active window has been changed, the program will prompt you to save the changes.

Directions:

(1) Choose *Files/Close*

See Also: [Files Menu](#)

Print (Files Menu)



Purpose:

Print prints the contents of the active window. This option is only active when the contents of the window can be printed.

Directions:

(1) Select the 'Print' button or choose *Files/Print*

See Also: [Files Menu](#)

Current Data to Instrument (Midi Menu)



Purpose:

Current Data To Instrument sends one or more Data Files from the active window. The menu item corresponds to the 'Send' button found in the Group, DBase, Library, and Bank Editor Windows. It is used to send data files from the window to the appropriate instruments.

This menu item is context sensitive and is only available when the active window has a data file it can send. In the case of a Group or DBase, choosing the option will send all currently selected data files.

Directions:

To send data from an active window to an instrument:

- (1) Activate the window containing the data to send
- (2) Choose *Midi/Current Data to Instrument*

See Also: [Midi I/O Window](#)

See Also: [Midi Menu](#)

Current DBase/Group to Instruments (Midi Menu)

Purpose:

Current DBase/Group to Instruments sends an entire Group or DBase from the active window.

This menu item is context sensitive and is only available when the active window is a DBase or Group. This option transmits the entire DBase or Group regardless of whether items are selected or not.

Directions:

To send an entire DBase or Group from an active window to an instrument(s):

- (1) Activate the window containing the DBase or Group.
- (2) Choose *Midi/Current DBase/Group to Instruments*

See Also: [Midi I/O Window](#)

See also: [Midi Menu](#)

Data from Disk to Instrument (Midi Menu)

Purpose:

Data from Disk To Instrument sends a data file directly from disk to an instrument. This function is similar to *Midi/Current Data to Instrument* except that the data is sent from a disk file instead of from the active window.

If it is only necessary to transmit a particular data file to an instrument when no editing is planned, this is the fastest way to load and transmit a data file.

Directions:

- (1) Choose Midi/Data from Disk to Instrument.
- (2) Use the File Selector to choose the file to load.

See Also: [Midi I/O Window](#)

See also: [Midi Menu](#)

DBase from Disk to Instruments (Midi Menu)

Purpose:

DBase from Disk to Instruments sends an entire Group or DBase directly from disk to one or more instruments. This function is similar to *Midi/Current DBase/Group to Instrument* except that the data files are sent from a disk file instead of from the active window.

If it is only necessary to transmit a particular Group or DBase in order to configure your system and no editing is planned, this is the fastest way to load and transmit the data.

Directions:

- (1) Choose *Midi/DBase/Group from Disk to Instrument*.
- (2) Use the File Selector to choose the DBase or group to load.

See Also: [Midi I/O Window](#)

See also: [Midi Menu](#)

MidiX from Disk to Instrument (Midi Menu)

Purpose:

MidiX from Disk to Instrument sends a MidiX format file directly from disk to an instrument.

With this function you can use the program to transmit Patches stored in MIDIX format to an instrument. These files may have been obtained from other sources such as BBSs or from 3rd party sound manufacturers.

While it may not be possible to directly open and edit these files with the program, it is always possible to transmit the data to the instrument. If the the program is capable of storing or editing the instrument's data, you can use the drivers to reload the data. Additionally, any MIDIX format file can be transmitted regardless of whether the program supports the instrument or not.

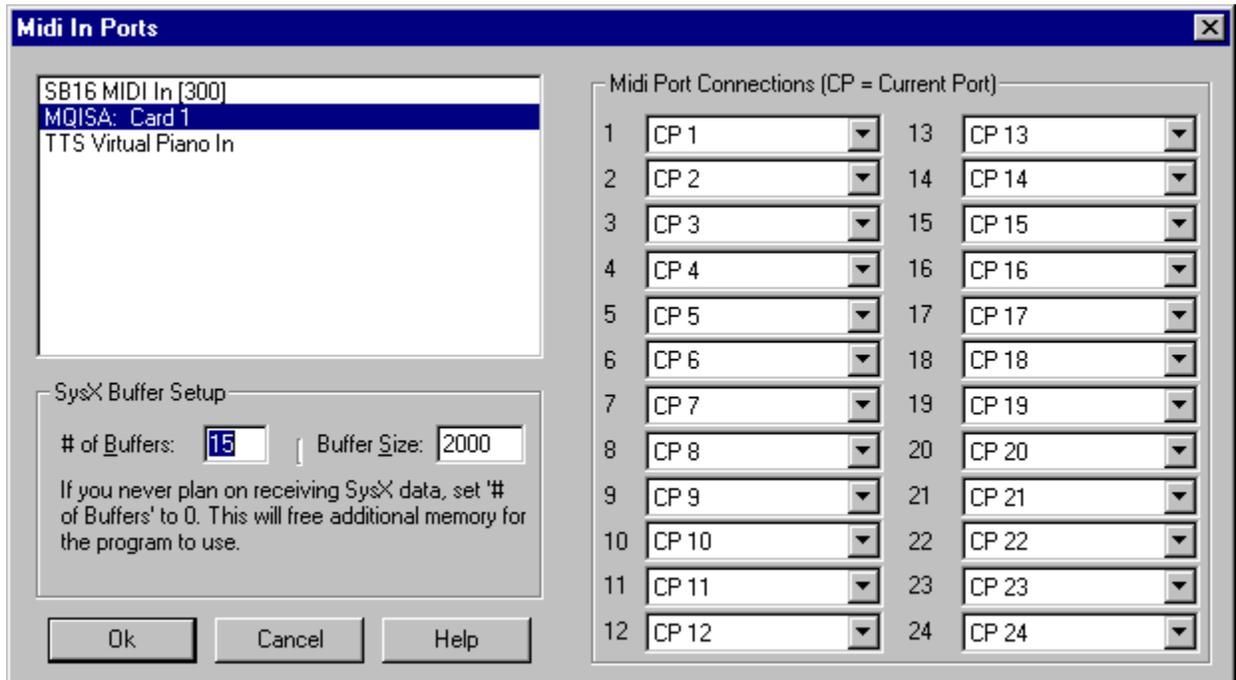
Directions:

- (1) Choose *Midi/MidiX from Disk to Instrument*.
- (2) Use the File Selector to choose the MidiX file to load and transmit

See Also: [Midi I/O Window](#)

See also: [Midi Menu](#)

Midi In Ports (Midi Menu)



Purpose:

Midi Out Ports selects the MIDI Out Ports available to the program. If you are using a 16-bit program, the selected ports will be used by all Sound Quest 16-bit programs. If you are using a 32-bit program, the selection is made just for the current program.

For the program to transmit MIDI data, it must do so through open MIDI OUT Ports. You should only open ports you plan to use with the program. This will reduce the likelihood of experiencing a number of possible MIDI communications problems. Additionally, you should not open ports to the FM portions of sound cards or any other specialty driver such as those used to transfer SYMPTE timing data unless you are using them with the program.

To select the MIDI OUT ports to use with the program, highlight the ports you would like to have open from the list on the left hand side of the dialog. (Hold down the CNTRL key to make individual selections.)

Midi Port Connections

Midi Port Connections adds MIDI Port mapping capabilities. If you are only using a single MIDI port, there is no need to learn about, or make use of, Midi Port Connections so feel free to proceed to the next topic.

This program has support for up to 32 Midi Out connections. By default, the program automatically connects the first program MIDI OUT port to the first selected physical MIDI OUT port (as select above). The second program port is connected to the second physical port and so on. As long as you don't frequently make changes to the MIDI interfaces you use, this system will work well. If you do make changes then there is the possibility for problems. Sound Quest programs store their port assignments by number so if you intentionally or inadvertently change the order of your MIDI ports, you may find that MIDI data is no longer being sent where you intended it to go. Midi Port Connections was designed to solve

this problem. By offering the option of locking certain ports by name so that even if the MIDI port order changes, your MIDI connections will still remain correct.

Go to the Midi Port Connections portion of the dialog and click on the drop down list beside #9. In that list you will see each of your MIDI ports listed below the default settings. Choose one of the available MIDI ports. You have now locked port 9 to the selected MIDI port. If you now assign any type of MIDI or SysX data to this port, your data will always be sent to the correct MIDI port even if the order of your MIDI ports changes in the future.

When selecting ports in the program, you are selecting a port by name when it is contained in braces such as "[MQISA: Card 1 Out 1]" while a direct port connection will simply appear as "MQISA: Card 1 Out 1". Remember that if you choose a MIDI port directly and then add a new MIDI interface, you may find that your MIDI port assignments are no longer correct.

Directions:

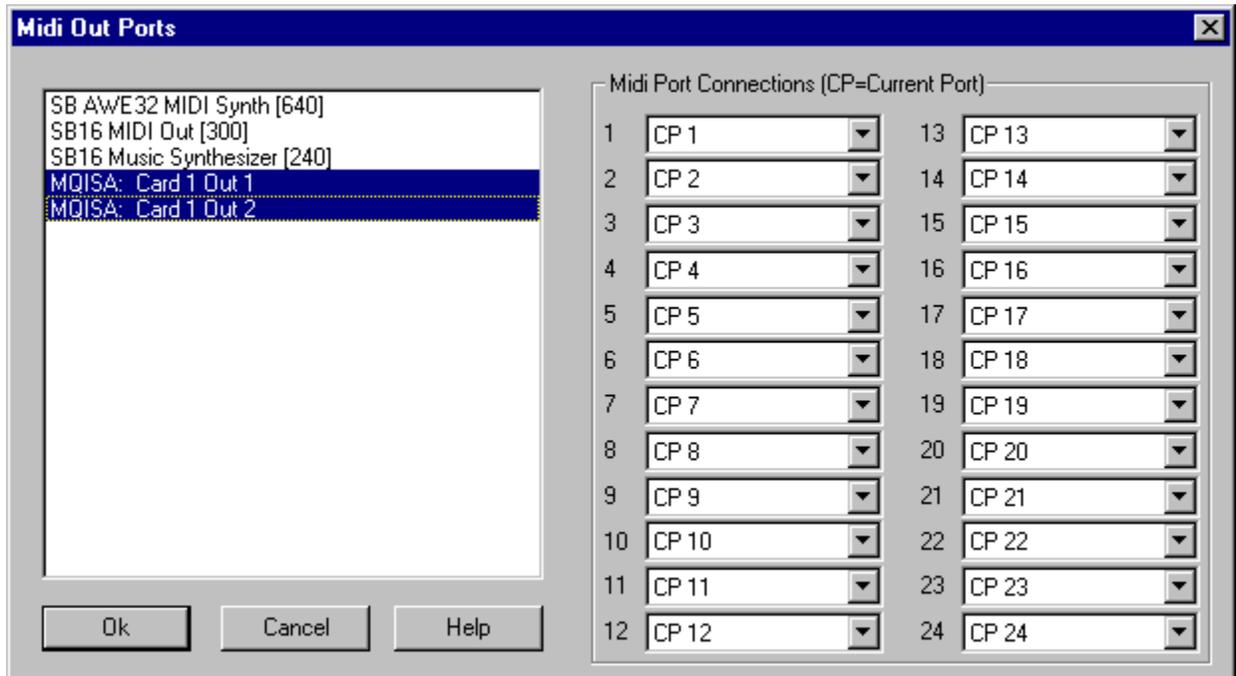
To select or change the selection of Midi In Ports:

- (1) Choose *Midi/Midi In Ports*
- (2) Select the MIDI Ports that the program is to use

See also: [Midi Out Ports](#)

See also: [Midi Menu](#)

Midi Out Ports (Midi Menu)



Purpose:

Midi Out Ports selects the MIDI Out Ports available to the program. If you are using a 16-bit program, the selected ports will be used by all Sound Quest 16-bit programs. If you are using a 32-bit program, the selection is made just for the current program.

For the program to transmit MIDI data, it must do so through open MIDI OUT Ports. You should only open ports you plan to use with the program. This will reduce the likelihood of experiencing a number of possible MIDI communications problems. Additionally, you should not open ports to the FM portions of sound cards or any other specialty driver such as those used to transfer SYMPTE timing data unless you are using them with the program.

To select the MIDI OUT ports to use with the program, highlight the ports you would like to have open from the list on the left hand side of the dialog. (Hold down the CNTRL key to make individual selections.)

Midi Port Connections

Midi Port Connections adds MIDI Port mapping capabilities. If you are only using a single MIDI port, there is no need to learn about, or make use of, Midi Port Connections so feel free to proceed to the next topic.

This program has support for up to 32 Midi Out connections. By default, the program automatically connects the first program MIDI OUT port to the first selected physical MIDI OUT port (as select above). The second program port is connected to the second physical port and so on. As long as you don't frequently make changes to the MIDI interfaces you use, this system will work well. If you do make changes then there is the possibility for problems. Sound Quest programs store their port assignments by number so if you intentionally or inadvertently change the order of your MIDI ports, you may find that MIDI data is no longer being sent where you intended it to go. Midi Port Connections was designed to solve

this problem. By offering the option of locking certain ports by name so that even if the MIDI port order changes, your MIDI connections will still remain correct.

Go to the Midi Port Connections portion of the dialog and click on the drop down list beside #9. In that list you will see each of your MIDI ports listed below the default settings. Choose one of the available MIDI ports. You have now locked port 9 to the selected MIDI port. If you now assign any type of MIDI or SysX data to this port, your data will always be sent to the correct MIDI port even if the order of your MIDI ports changes in the future.

When selecting ports in the program, you are selecting a port by name when it is contained in braces such as "[MQISA: Card 1 Out 1]" while a direct port connection will simply appear as "MQISA: Card 1 Out 1". Remember that if you choose a MIDI port directly and then add a new MIDI interface, you may find that your MIDI port assignments are no longer correct.

Directions:

To select or change the selection of Midi In Ports:

- (1) Choose *Midi/Midi Out Ports*
- (2) Select the MIDI Ports that the program is to use

See also: [Midi In Ports](#)

See also: [Midi Menu](#)

Driver List Window



Purpose:

The Driver List Window is the main control center for obtaining data from your instruments. It can load and direct SysX data into the various storage and editing windows: Patch Editor, Bank Editor, Group, DBase, Library, and SysX View as well as save the data directly to disk.

Loading data from disk is also easiest from this window because of the additional information it can provide. Basically choosing the driver for the data you wish to load and then selecting any of the options in the *Drivers/Open* sub menu will automatically open a file selector in the correct directory to load the desired data.

Because it is the main control center, you will be able to access to virtually all of the program's features from the the Driver List via the regular and the right mouse button speed menus.

This window opens automatically when the program starts up and the last Driver List in use is loaded.

For more information on the contents of the Driver List Window see: [Contents](#)

Driver List Menus

The Driver List contains a wide range of menu functions to organize the Patches and other data used by your various instruments. These functions are all contained in the Drv List menu. The Driver List window must be activated before you can see and access the Drv List menu.

<u>Add Instruments</u>	add on or more instruments to the driver list
<u>Remove Drivers</u>	remove the selected drivers from the list
<u>Remove All Drivers</u>	clears the list completely
<u>Select All Drivers</u>	selects all drivers in the list
<u>Update Settings...</u>	modify the settings for the selected driver
<u>Fast Tips</u>	get info on the selected driver
<u>Build Driver List</u>	automatically build a new Driver List
<u>Instrument To/Group/Editor</u>	obtain the selected data from the instrument
<u>Instrument To/Disk</u>	obtain the selected data and save directly to disk
<u>Instrument To/DBase</u>	obtain the selected data and place in a new DBase
<u>Instrument To/SysX View</u>	obtain the selected data and place in a SysX View
<u>Open/Group/Editor</u>	select a group of editor to load
<u>Open/Library</u>	select a library to load
<u>Open/DBase</u>	select a DBase to load
<u>Open/SysX View</u>	select a Bank or Patch file and place it in a SysX View
<u>Open/MidiX</u>	import SysX data from a MIDIX file
<u>Open/SMF</u>	import SysX data from a standard MIDI file
<u>Open/Import</u>	import SysX data from headerless data file
<u>New/Group/Editor</u>	create a new Group, Bank or Patch
<u>New/Library</u>	create a new Library for the selected data type
<u>New/DBase</u>	create a new DBase
<u>Utilities/Midi Controller Window</u>	open the MIDI Controller Window
<u>Utilities/Sequencer Window</u>	open the Sequencer Window
<u>Utilities/Midi Monitor Window</u>	open the MIDI Monitor window
<u>Utilities/File Conversion Window</u>	open the File Conversion window
<u>Utilities/Driver Creator Window</u>	open the Driver Creator window
<u>Open Driver List</u>	open the Driver List window

New Library (Driver List)



Purpose:

New Library creates a new Library for a particular type of data. Since a Library can contain only one type of data, you must first choose the driver for the type of Library you wish to create. Libraries can only be created for single data types. The New Library button and menu functions are disabled if a Bank driver is selected.

Directions:

To add a new Library:

- (1) Select the driver for which a new Library is to be created
- (2) Only drivers for single data types such as a Patch can be used to create a Library
- (3) Select the New Lib button or choose *Drivers/New/Library*

See Also: [Library Window](#)

See Also: [Driver List Window](#)

New Group/Editor (Driver List)



Purpose:

New Group/Editor opens a new Group or Editor window. The new Group or Editor will have all of its parameter values set to 0. This function will work with instrument groups, and with most Bank drivers and single drivers for which there are editors available.

This function works in two different modes depending on whether an Instrument Bar or a driver is selected. If an Instrument Bar is selected, clicking on the 'NEdit' button will create a new Group. If a particular driver is selected, a new Bank or Patch editor is created. If neither of these editors is available for the selected driver, the data is placed in a SysX View Window.

Directions:

To add a new Editor window:

- (1) Select the Instrument Bar or driver for the data you wish to add
- (2) Choose *Drivers/New/Group* or *Drivers/New/Editor* depending on the Driver List entry you have selected

See also: [Driver List Window](#)

New DBase (Driver List)

Purpose:

New DBase creates a new DBase.

Remember that adding new entries to an existing DBase is as easy as highlighting drivers in the Driver List Window and then pressing the 'Recv' button in the DBase Window.

Description:

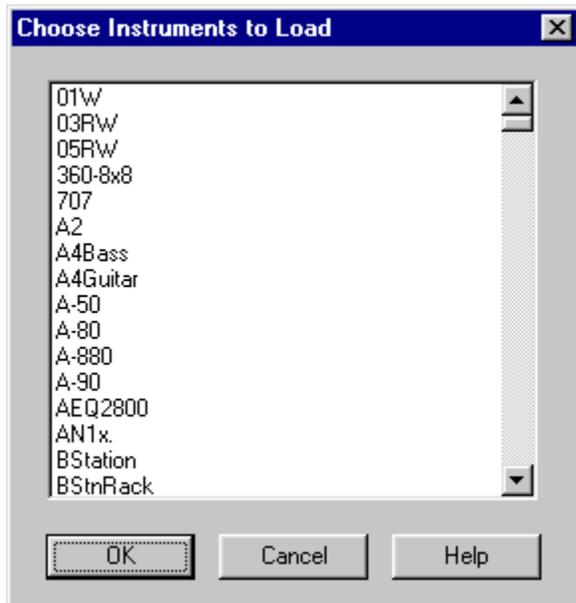
To open a new DBase:

(1) Choose *Driver List/New/DBase*

See Also: [DBase Window](#)

See Also: [Driver List Window](#)

Add Instrument (Driver List)



Purpose:

Add Instrument adds an instrument to the Driver List. This function only adds instruments to the Driver List which have been previously installed on the hard drive by the program's installer.

This function is useful in the following circumstances: (1) it allows instruments which have been previously removed from the Driver List to be reinstalled. (2) A Driver List can be created which has multiple copies of the same instrument. For instance, you could install four sets of TX81Z drivers if you had four TX81Zs. (3) It allows you to add drivers for a new instrument without losing your current Driver List.

The assumption of Add Instrument is that the drivers being added has already been installed on the hard drive using the program's installer. If you have downloaded support for a new instrument from Sound Quest's BBS or received the files on disk, it is necessary to exit the program, run the program's installer, and use it to install the new instrument(s) onto the hard drive. Once this has been done, choose Add Instrument to display a list of the available instruments and choose the new instruments to add to the current Driver List.

Directions:

To add instrument to the Driver List:

- (1) Choose *Drv List/Add Instrument...*
- (2) Select the instrument to add to the list. More than one instrument may be selected
- (3) Click on OK to add the instrument drivers

See also: [Driver List Window](#)

Remove Drivers (Driver List)

Purpose:

Remove Drivers removes one or more drivers from the Driver List. This allows you to remove entire instruments which are not in use from your list. It also allows you to remove particular drivers for an instrument if you wish. For large, sophisticated instruments, removing the drivers you are not using will make the program much more manageable.

Note: Once any individual driver is removed from an instrument in the Driver List, the only way to put the driver back into the list is to add the entire instrument using [Add Instrument](#).

Directions:

To remove one or more drivers from the Driver List:

- (1) Select the drivers you wish to remove from the Driver List
- (2) If an Instrument is selected, all of its drivers will be removed
- (3) Choose *Drv List/Remove Drivers*

See also: [Driver List Window](#)

Remove All Drivers (Driver List)

Purpose:

Remove All Drivers removes all of the Instrument Bars and drivers in the Driver List. This option would most commonly be used immediately before building a new Driver List.

Remember that the program does support multiple Driver Lists so you can have multiple configurations that can be read from disk at the appropriate time. This is particularly advantageous for studios where a separate Driver List can be kept for each client's instruments and their respective MIDI Port and channel settings.

Directions:

To remove all drivers from the Driver List:

(1) Choose *Drivers/Remove All Drivers*

See also: [Driver List Window](#)

Select All Drivers (Driver List)

Purpose:

Select All Drivers selects all of the drivers in the Driver List.

Directions:

To select all drivers in the Driver List:

(1) Choose *Drivers/Select All Drivers*

See also: [Driver List Window](#)

Update Settings... (Driver List)



Purpose:

Update Settings changes the basic settings of a individual driver or all of the drivers in an instrument. If an Instrument Bar selected when Update Settings is chosen, all of the drivers in the instrument will receive the changes. If a driver is selected when Update Settings is chosen, changes are made only to that driver.

In both cases, choosing Update Settings displays the Settings Dialog. This dialog is used to make changes to the basic parameter settings of not only drivers in a Driver List but also Patch Editors, Bank Editors, Groups, DBases, and Libraries.

It is through this dialog that basic settings for proper instrument communication are set. See the [Settings Dialog](#) for additional information on the purpose and editing options of each of the basic parameters.

Directions:

To change the basic settings:

- (1) Select an Instrument or individual driver in the Driver List
- (2) Choose *Drv List/Update Settings...*
- (3) Make the desired changes to the basic settings and press OK

See also: [Settings Dialog](#)

See also: [Driver List Window](#)

Fast Tips (Driver List)



Purpose:

Fast Tips are available for most drivers in the Driver List. They are used to provide specific information on how to set up an instrument to receive and transmit data as well as provide special information, warnings, and tricks which are relevant to that instrument. The file will often include information on:

- how to enable SysX transmissions
- the name of the parameter equivalent to Sound Quest's "Comm Channel"
- requirements for successful communication between the instrument and program
- what type of data the driver stores
- the range and purpose of the Pch# parameter if it is used
- any known problems with instrument's ROMs
- if there are problems, what minimum ROM version should be used (if possible)

The information is displayed as a scrollable text window which will appear in the right hand corner of the window

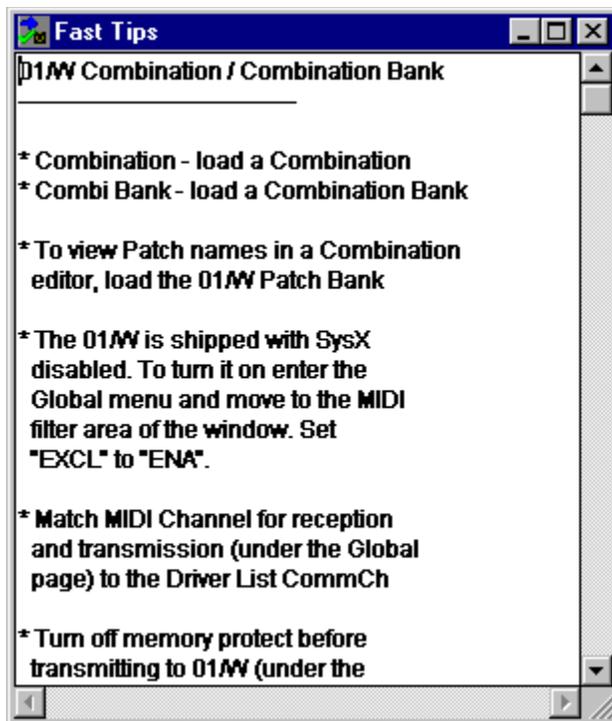
Directions:

To view a driver's Fast Tips:

- (1) Select the driver you wish to obtain information for
- (2) Choose *Drivers/Fast Tips*.

Example:

The following is an example of a Fast Tips Window for the 01/W Combination driver



See also: [Driver List Window](#)

Build Driver List (Driver List)

Purpose:

Build Driver List automatically creates a Driver List containing the drivers for all currently installed instruments. This is the fastest way to create a Driver List.

Note: Building a new Driver List automatically removes the current Driver List and its settings from the window. The new Driver List should be saved under a different name if you plan to retain the original Driver List for future use.

Directions:

To build a Driver List containing all installed instruments:

(1) Choose *Drivers/Build Driver List*

See also: [Driver List Window](#)

Instrument to SysX View (Driver List)

Purpose:

Instrument to SysX View loads data from the instrument directly into a SysX View window. This feature is used while developing and testing new drivers and templates for the program.

Directions:

To load data from your instrument and put it in a hexadecimal display:

- (1) Select the data from the list in the Driver List Window
- (2) Choose *Drivers/Instrument to/SysX View*

If there are any problems, an error will be displayed and you can check Fast Tips for help, otherwise, a SysX View window will open with the data.

See also: [SysX View Window](#)

See also: [Driver List Window](#)

Instrument to Group/Editor (Driver List)



Purpose:

Instrument to Group/Editor is one of the most commonly used in the program. It loads SysX data out of the instrument and into a Group, Bank or Patch Editor window.

This function works in two different modes depending on whether an Instrument Bar or a driver is selected in the Driver List. If an Instrument Bar is selected, clicking on the 'Edit' button will create and load a new Group. If a driver is selected, the data is loaded from the instrument directly into a Bank or Patch editor. If neither Bank nor Patch editors is available, the data is placed in a SysX View Window.

When an Instrument Group is loaded, it contains data for each of the instrument's drivers that are part of the group. The instrument drivers that have both 'Grp' and 'Auto' designators are loaded directly from the instrument. Those drivers designated with just 'Grp' are added to the Group but not loaded from the instrument. These drivers typically load less frequently utilized data types which are also part of the internals of the instrument. For more information please see Groups.

If there is an error while loading data, please check [Fast Tips](#) for the driver and the [Midi Communication Errors](#).

Directions:

To load data from your instrument and place it in a Group, Bank or Patch Editor:

- (1) Select the type of data you wish to load from the Driver List Window (ie: Yamaha DX7 Voice Bank)
- (2) If an instrument is chosen, select the 'Edit' button, double click on the instrument icon in the Driver List or choose *Drivers/Instrument To/Group* **or**
- (3) If an instrument driver is chosen, select the 'Edit' button or choose *Drivers/instrument To/Edit*.

If there are any problems an error will be displayed and you can check Fast Tips for help. Otherwise the data is displayed in an Editor Window.

Note: if an instrument driver is chosen which is not editable, the received data will be placed in a SysX View window.

See Also: [Group Window](#)

See Also: [Patch Editor Window](#)

See Also: [Bank Editor Window](#)

See also: [Driver List Window](#)

Instrument to Disk (Driver List)



Purpose:

Instrument to Disk loads data directly from your instrument directly to disk. If you are just interested in saving SysX data without viewing or editing it, this is the fastest and easiest method. This function is only active when specific drivers are selected. It will not work when an Instrument Bar is selected.

If there is an error, please check [Fast Tips](#) for the driver and the [Midi Communication Errors](#).

Directions:

To load data from your instrument and save it directly to disk:

- (1) Select the data from the list in the Driver List Window
- (2) Select the 'Disk' button or choose *Drivers/Instrument to/Disk*
- (3) If there are any problems, an error is displayed and you can check Fast Tips for help
- (4) A File Selector is displayed
- (5) Use the selector to give the data a name and select `Ok' to save the data to disk.

See also: [Driver List Window](#)

Instrument to DBase (Driver List)



Purpose:

Instrument to DBase loads data files from one or more instruments directly into a new DBase.

This feature allows a complete or partial system snapshot to be taken of your system. One of the most useful organizational features of this program is that it can be used to collect and store all of the settings from all of the instruments in a MIDI system. Effectively, a MIDI snapshot. This allows all of the various pieces of SysX data for each instrument necessary for a piece of music to be stored together in one place.

The process begins with selecting the drivers for the different types of data that need to be loaded into the Driver List. Once this is done, select the DBase button. The program will create a new DBase and load data for each selected driver into it.

If there is an error, please check [Fast Tips](#) for the driver and the [Midi Communication Errors](#).

Directions:

To open a new Data Base and load instrument data into it:

- (1) Select one or more drivers from the Driver List Window
- (2) Choose *Drivers/Select All Drivers* to take a complete system snapshot
- (3) Select the 'DBase' button or choose *Drivers/Instrument to/DBase*

See also: [DBase Window](#)

See also: [Driver List Window](#)

Open Group/Editor (Driver List)

Purpose:

Open Group/Editor loads an instrument Group or data file from disk into a Group or Editor window. If the file is a Patch or Bank data file from disk into a Patch or Bank Editor respectively. This is similar to choosing *Files/Open...* except that when chosen from the Drivers menu, the File Selector is automatically placed in the correct directory with the correct extension for loading a Group or data file.

This menu item is bi-modal.. If an Instrument Bar is selected, 'Group' is the first item displayed in the *Drivers/Open* sub-menu. If a driver is selected, 'Editor' is displayed as the first item of the *Drivers/Open* sub-menu.

Directions:

- (1) Select the Instrument Bar or driver for the type of data you wish to load
- (2) Choose *Drivers/Open/Group* or *Drivers/Open/Editor*
- (3) Use the File Selector to load the desired data file

See Also: [Group Window](#)

See Also: [Patch Editor Window](#)

See Also: [Bank Editor Window](#)

See Also: [Driver List Window](#)

Open Library (Driver List)

Purpose:

Open Library loads a Library from disk.

This function is similar to choosing *Files/Open...* with the added advantage that the displayed File Selector automatically opens to the correct directory with the Library .SQL extension.

Directions:

- (1) Select the driver for the Library type you wish to load
- (2) Choose *Drivers/Open/Library*
- (3) Use the File Selector to load the desired Library

See Also: [Library Window](#)

See Also: [Driver List Window](#)

Open DBase (Driver List)

Purpose:

Open DBase loads a DBase from disk.

Its function is similar to choosing *Files/Open...* with the added advantage that the displayed file selector will automatically be opened to the DBase directory with the DBase .SQD extension.

Directions:

- (1) Choose *Drivers/Open/DBase*
- (2) Use the file selector to load the desired DBase

See Also: [DBase Window](#)

See Also: [Driver List Window](#)

Open SysX View (Driver List)

Purpose:

Open SysX View loads a Patch or Bank from disk into a SysX View Window.

This function is similar to choosing *Files/Open...* with the added advantage that the displayed File Selector automatically opens to the correct directory with the correct extension for the type of data file you wish to load.

Directions:

- (1) Select the driver for the type of data you are loading
- (2) Choose *Drivers/Open/SysX View*
- (3) Use the file selector to load the desired data file

See Also: [SysX View Window](#)

See Also: [Driver List Window](#)

Sound Checker

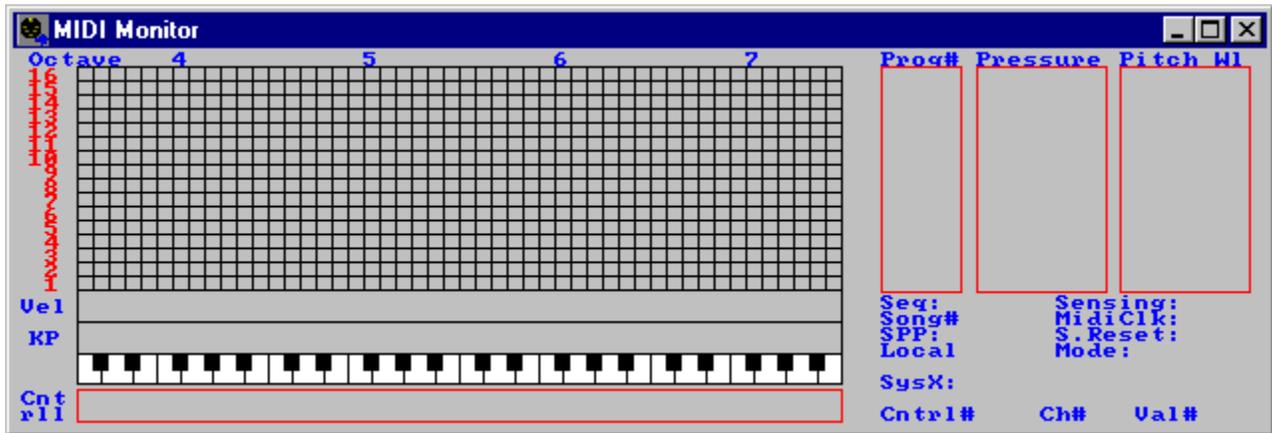
Purpose:

The Sound Checker provides an integrated environment in which to develop, test, and audition Patches. The Sound Checker is comprised of four tools:

- the MIDI Sequencer Window - to play Standard MIDI Files
- the MIDI Monitor Window - to view MIDI Port activity
- the MIDI Controller Window - to transmit any standard MIDI message

With these tools, it is no longer necessary to return to a sequencer to see if the sound being developed is 'right'.

Midi Monitor Window (Sound Checker)



Purpose:

The MIDI Monitor displays MIDI data arriving at the MIDI IN Ports. The MIDI data can be viewed as either graphics or text.

The text mode is ideal for viewing the exact MIDI output of your instrument. This can assist you in learning about MIDI. It is also extremely useful while sorting out unusual problems in your MIDI system. The text mode is also useful for viewing SysX dumps and is invaluable when creating your own drivers for MIDI QUEST or MIDI QUEST Jr.

In graphic mode, the current status of four octaves of MIDI notes along with the other standard MIDI messages are displaying in the window. The displayed octave range can be selected from the menus.

Using the Midi Monitor

To open the MIDI Monitor Window, choose *Utilities/MIDI Monitor Window*, or *Drivers/Utilities/MIDI Monitor Window*, or press the Midi Monitor button on the button bar.

The MIDI Monitor has 5 basic options. Each option is selected from the menus:

Base Octave chooses the octaves which are visible while the window is in graphic mode.

Monitor Midi Out is only visible while the window is monitoring the Midi Inputs. Choose this item to monitor the output of the program's Standard MIDI File sequence player.

Monitor Midi In is only visible while the window is monitoring the Standard MIDI File sequence player. Choose this item to have the window monitor the computer's MIDI INputs.

Switch to Text Mode is only visible while the window is in graphic mode. Choose this item to switch the Midi Monitor Window to text mode

Switch to Graphic Mode is only visible while the window is in text mode. Choose this item to switch the Midi Monitor Window to graphic mode.

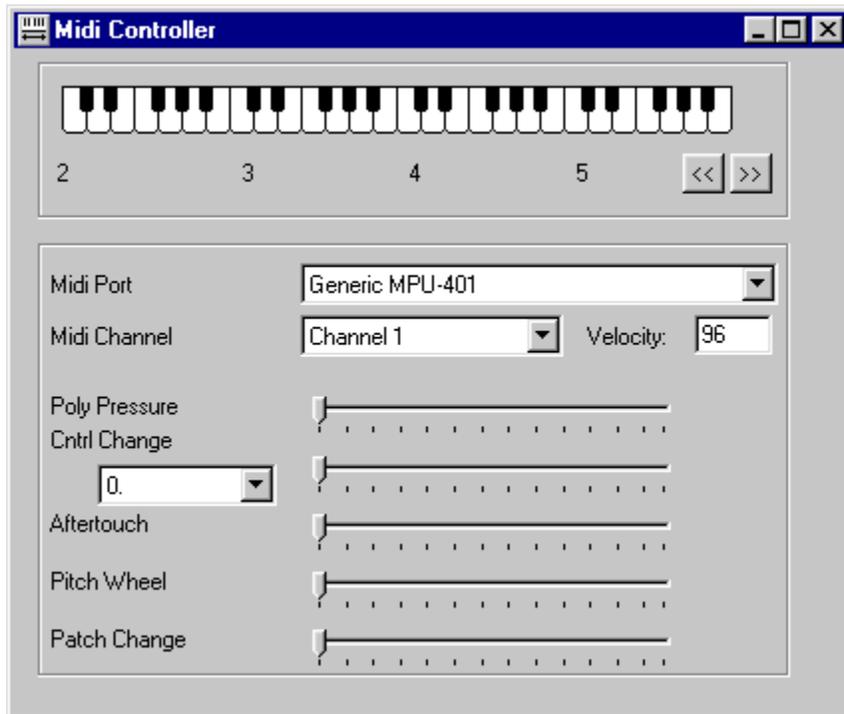
Filter MIDI Real Time Messages is a toggle item. Check it to block the display of MIDI real time

messages. This is particularly useful in text mode when MIDI's active sensing message (FE) needs to be blocked.

See also: [Sound Checker](#)

See also: [Utilities Menu](#)

Midi Controller Window (Sound Checker)



Purpose:

The MIDI Controller Window window is a utility window which plays no part in the actual editing or storage of SysX data. It is used to generate standard MIDI messages: Notes, Poly Pressure, Control Changes, Program Changes, After Touch, and Pitch Wheel Changes. Should the need arise, this window provides you with a complete range of MIDI control and auditioning capabilities.

For example, if a drum sound assigned to a particular note (in an instrument with drum capabilities) is being edited, click on that note to audition it

This window can be used in conjunction with the Standard MIDI File Sequence player. You can have a sequence playing and easily experiment with adding controller data to the sequence from the comfort of the computer.

The Midi Controller Window automatically tracks and retains the MIDI channel and Port used by the last active Patch, Bank, DBase, Group, and Library Window so that auditioning Patches using the correct MIDI Port and channel is easy.

Using the MIDI Controller Window

To open the MIDI Controller Window, choose the 'Midi Controller' button or *Utilities/MIDI Controller Window*. When the Drivers List Window is active, *Drivers/Utilities/Midi Controller Window* can also be chosen.

The MIDI Controller Window contains a graphic keyboard. Clicking the mouse on any of the notes will

play that note on the selected MIDI channel and Port. If the mouse button is held down, dragging across the keyboard will play the different notes.

The window also contains six sliders which are used to set the MIDI channel, send a Patch change message and generate continuous MIDI messages:

MIDI Port: selects the port that the MIDI messages are sent to.

MIDI Channel: selects the channel for the MIDI messages.

Vel: sets the velocity level for notes that are played on the keyboard

Poly Key Pressure: transmit pressure levels for the last key played on the keyboard. This slider will automatically snap back to 0 when released, just like a real keyboard.

Control Change: uses the first parameter to select a controller number (ie. Volume = 7, Breath Controller = 2), and the second to transmit controller levels.

After Touch: uses the slider to continuously transmit after touch data. This slider will automatically snap back to 0 when released, just like a real keyboard.

Pitch Wheel: uses the slider to continuously transmit pitch wheel data. This slider will automatically snap back to 0 when released, just like a real keyboard.

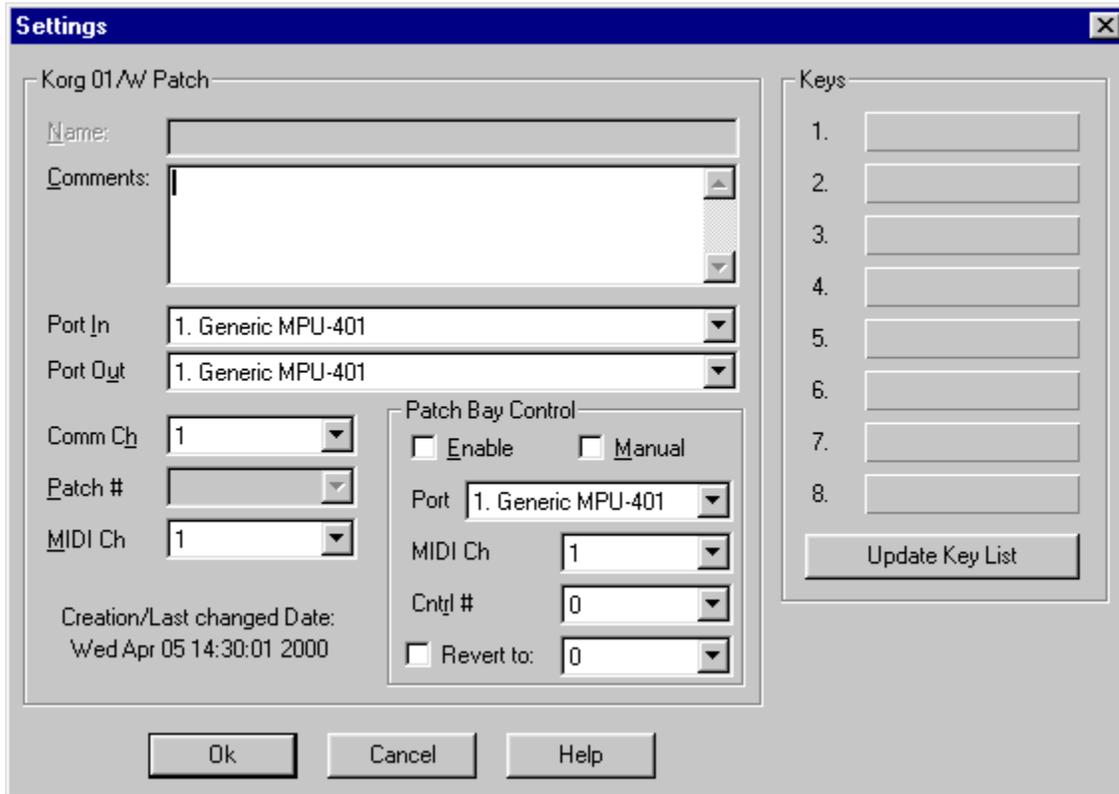
Program Change: uses the slider to select a program change. The change is sent when the mouse button is released.

The arrows (<<, >>) are used to select the octave range of the graphic keyboard.

See also: [Sound Checker](#)

See also: [Utilities Menu](#)

Settings Dialog (Common Elements)



About the Settings Dialog

The Settings Dialog contains the basic settings for each data file. The port and channel settings in this window are critical for the correct transfer of the SysX data between the computer and the instrument and must be set correctly.

Dialog Parameters:

Name: this is the name of the file. The parameter will only be active when the dialog is opened from a DBase window. Normally, the name of a file is set by choosing *Files/Save As...*

Comments: a comment can be assigned to any data file and may contain any pertinent information you want to include about the data such as what the file is used for, who created it, and so on.

Port In: this drop down menu lists the open MIDI IN Ports. Select the port to which the MIDI OUT cable of the instrument is currently connected. If this menu is empty, the program will not be able to communicate properly with any of your instruments. Please read Port Out below for additional information on MIDI Port problems. See Also: [Midi In Ports](#).

Port Out: this drop down menu lists the open MIDI OUT Ports. Select the port to which MIDI IN cable of the instrument is currently connected. If this menu is empty, the program will not be able to communicate

with any of your instruments either because you have not installed any MIDI drivers in windows or because another program is using the MIDI Port and the port is not shareable. If you have been running other Windows MIDI programs successfully, make sure all of these applications are closed and rerun the program. If no other MIDI programs have been run on the computer, you probably need to install MIDI drivers using the Control Panel in the Main Group of Window's Program Manager. See Also: [Midi Out Ports](#)

Comm Ch: the Communication Channel is the channel that the instrument uses to communicate SysX data on. This channel MUST match the channel that the instrument is using or nothing will happen because the instrument will ignore the messages sent by the computer! The Comm Channel is not always the MIDI channel but may be. On some older instruments this channel is not used at all. Please check the instrument's drivers [Fast Tips](#) for information on how to correctly set the Comm Channel.

Midi Ch: the Midi Channel should be set to match the basic MIDI channel of the instrument. The MIDI Channel is used largely for auditioning purposes and should match the instrument so that auditioning from Banks and Patch Editors can be heard. This channel should be set correctly since a small number of instruments use it while transferring SysX data.

For Roland multi-part equipment, the MIDI channel used should be the same as that for part #1 of the instrument.

Patch #: the Patch # is used to select one Patch or Patch Bank from a range of Patches or Patch Banks. For example, the "Casio CZ Patch" driver will load 1 of up to 32 different Patches. The Patch # selects which Patch is loaded. In these instances, it also determines which Patch or Patch Bank location the data is sent to. By changing this parameter, it is then possible to load a Patch from location 2 and save it in location 7 of the instrument.

Keywords: Each data file can have up to 8 keywords which describe the Patch. These keywords are currently used in the Library's [Choose Dialog](#) so that Patches meeting a given set of criteria can be selected for auditioning or editing. If all of your Patches are properly keyed, it is easy to find all of the bass Patches, bass Patches that are synthesized or just the pizzicato violins from all of the violins. While it may take some time to initially set up this information, it will save you a substantial amount of time in the long run since you will be able to find certain types of Patches very easily.

When the Settings Dialog is opened from the Driver List, the keywords are disabled. Keywords can only be assigned to data files, not to drivers in the Driver List.

To add or edit keys for a Patch, click the left mouse button in the raised area beside a key number. Upon clicking, a pop-up menu will appear which lists all of the currently available keys. Simply point to the desired key and release the mouse.

Update Key List: select this button to open the [Keys Dialog](#) where keys can be added and removed from the list.

Patch Bay Control

Patch Bay Control is used to automate control of your Patch bay. If Patch Bay Control is enabled, then before each data transmission, the software will send a program change to the Patch bay. The program change will be sent on the selected MIDI channel with the "Cntrl #" being the actual patch number sent.

For example, if your Patch bay is responding to Patch changes on MIDI channel 15 and Patch 10 connects the MIDI OUT of the computer to the MIDI IN of the instrument AND the MIDI In of the instrument to the MIDI OUT of the computer then in this section, the MIDI channel should be set to 15 and the Cntrl # for this driver should be set to 10. Port should be set to the MIDI port to which the patch bay is

connected.

Enable: when enable is checked, patch bay control is enabled. The control number, described above is then used to control the patch bay.

Port: the MIDI port that the patch change command is sent to in order to properly configure the patch bay

MIDI Ch: the MIDI channel that the patch bay is monitoring for patch changes.

Cntrl#: enter the number of the patch that will connect the computer to the instrument via the patch bay

Revert to: when checked, the associated patch change is sent after doing a bulk dump. This allows the patch bay to be configured so that an instrument can receive SysX data and then returned to a default configuration.

Manual: in some cases, the patch bay is not capable of responding to patch change commands. In this case, you can check this option. Instead of sending a patch change command, Midi Quest will display a message prompting you to manually configure the patch bay so that the computer can communicate with the appropriate instrument.

Options Menu

Purpose:

The Options Menu contains parameters which determine how the overall program operates. If Save Preferences is checked, these settings are automatically saved by the program for future use.

The menu items are:

- Save Preferences - when checked all Preferences settings are saved on program exit
- Preferences... - opens a tabbed dialog where all program settings may be altered
- Set Patch Editor Colors - sets the colors used in the Patch editors
- Open Toolbar - open a new toolbar

Save Preferences (Options Menu)

Purpose:

Save Preferences, when checked, saves all program settings to disk. The next time the program is started, this file is read and settings recalled. Save Preferences saves:

1. All Options menu settings.
2. All Tones Window parameters.
3. All Default Window parameters.
4. The file name of the file containing the list of drivers to load.
5. All currently loaded Standard MIDI Files.

Directions:

To toggle Save Preferences:

- (1) Choose *Options/Save Preferences*

Default: On

See also: [Options Menu](#)

Library Views (Library)

32 bit only for 16 bit [click here](#)

A Library has four different views: Columns, Names, Small Icons, Large Icons. These are identical to the view options you have in Explorer.

To select the default view, the view that each Library is opened with, see the [Library Tab](#) in the Properties dialog.

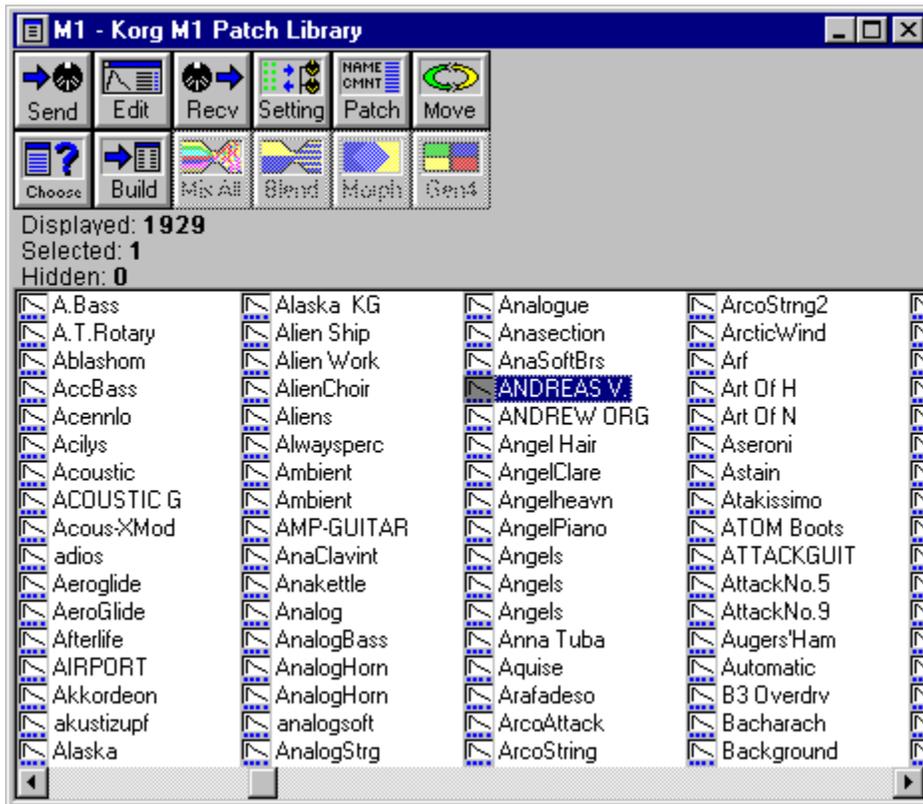
To change the view of the Library once the window is open, choose *Library/View/Display* and select the desired view.

Columns (Default)



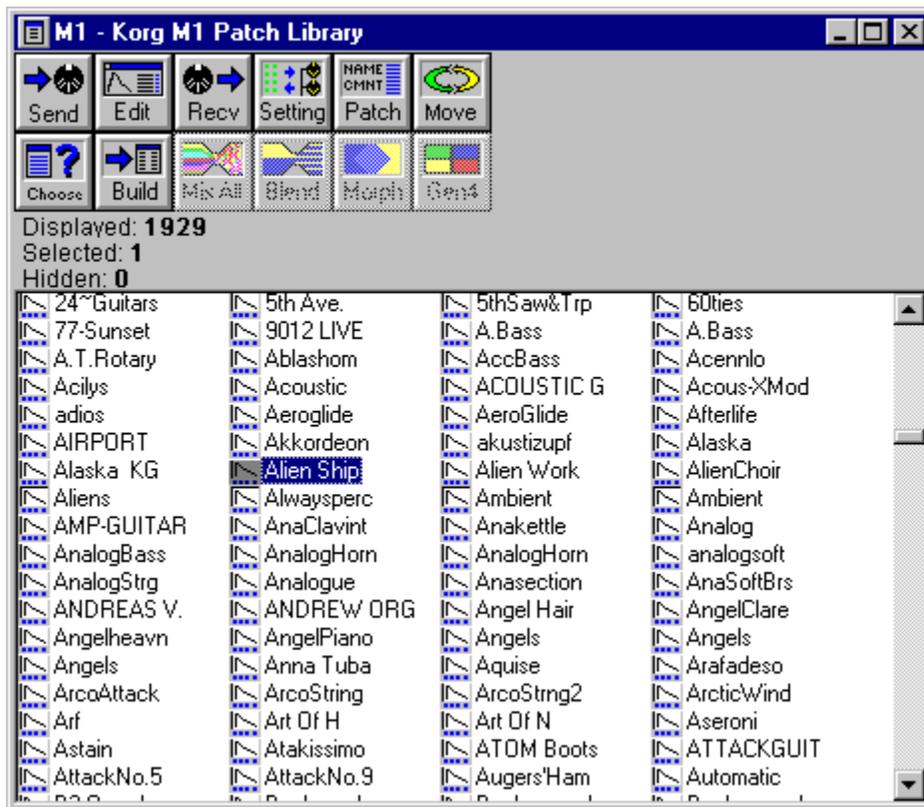
Shows the patches by name including columns for the listing of keys and comments.

Names



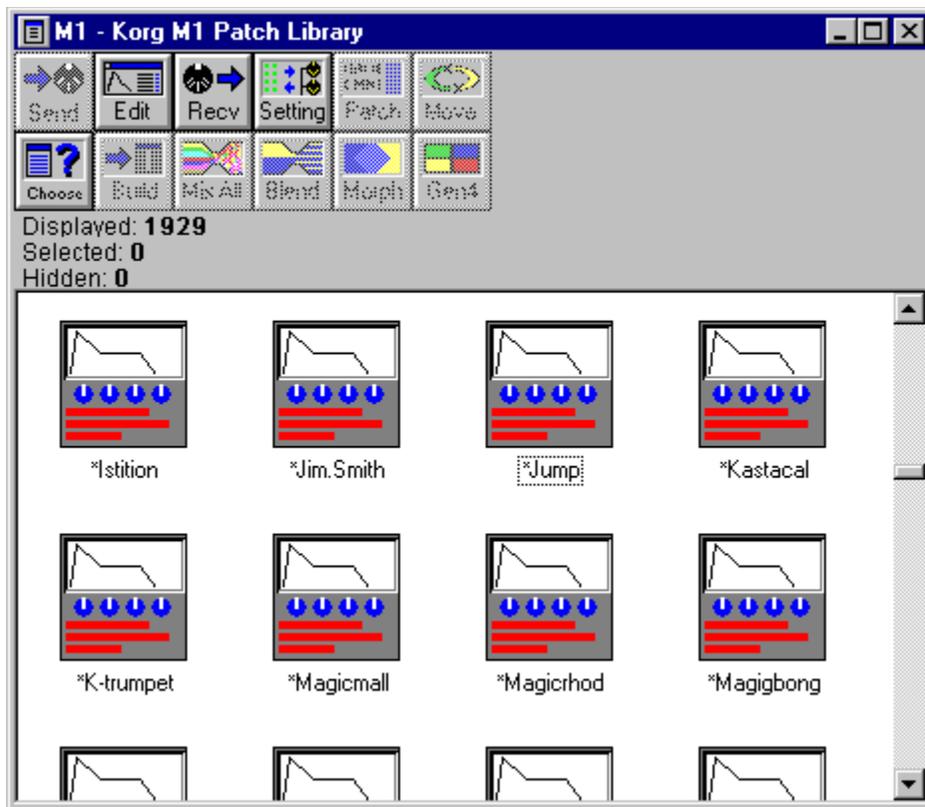
Lists names only in vertical columns with a small identifying icon

Small Icons



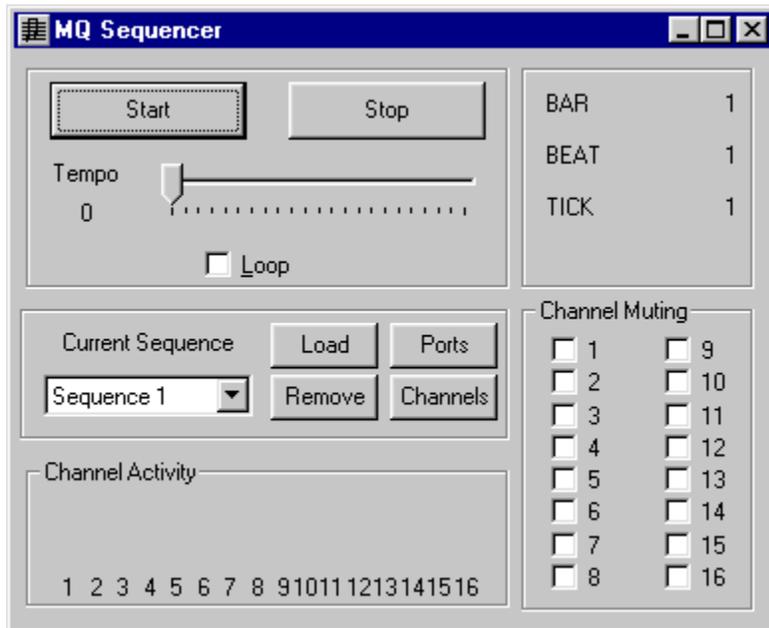
Lists names only in horizontal columns with a small identifying icon

Large Icons



Lists name only in horizontal columns with large identifying icon

Sequencer Window (Sound Checker)



Purpose:

The Sequencer window is a simple Standard MIDI File Sequence player. It is designed so that virtually all program functions can be performed while the sequencer plays. For example, Patches can be edited or auditioned, files loaded and saved, or new Patches created.

The window is capable of holding ten different SMF sequences and will automatically reload them when the program is next run. The channel activity area has a level meter to display activity for each active MIDI channel. The BAR, BEAT, and TICK parameters provide feedback on the sequence player's current location in the piece.

This sequencer, although not sophisticated, is extremely useful if you are using Windows MIDI drivers which do not permit multi program access. In such cases, it is not possible for your sequencer to play while the program is auditioning, editing, or performing other MIDI related functions. With this sequence player, all of this is possible.

Using the Sequencer Window

The controls within the Sequencer Window are:

Start: start playing the current sequence. If there is no sequence loaded, an error is displayed.

Stop: stops playing the current sequence.

Tempo: sets the tempo of the piece. If this is not set, the program will use the tempo map within the sequence itself. If a tempo is set, the sequence will remain locked to that tempo.

Loop: when checked the playing sequence will loop indefinitely or until stopped manually

Current Sequence: selects one of the ten sequences.

Load: loads a new standard MIDI file into the selected sequence

Remove: removes the standard MIDI file from the selected sequence

Ports: each channel in the sequence can be assigned to a different MIDI Port. This button displays a dialog which will assign each channel to a port. By default, all channels are set to the first open MIDI Port.

Channels: each track in the sequence can be assigned to a specific MIDI channel. This button displays a dialog which will assign each track to a particular MIDI channel. By default, there is no channel assignment.

Channel Muting: each MIDI channel may be individually muted so that only particular channels may be auditioned. When a particular channel box is selected, that channel is muted.

Directions:

To load and play a standard MIDI file sequence:

- (1) If necessary, save your piece in standard MIDI file format from your sequencer
- (2) Click on Load to display a File Selector
- (3) Choose a standard MIDI file and press OK to load it
- (4) Click on the Play button

See Also: Sequencer Ports

See Also: Sequencer Channel Map

See Also: Sound Checker

Utilities Menu (Common Elements)

The Utilities menu contains items which open all of the windows for the Sound Checker and other program Utilities.

The following menu items are included:

<u>Midi Controller Window</u>	- opens a window which to generates all types of standard MIDI events
<u>Sequencer Window</u>	- open a Standard MIDI File player window
<u>Midi Monitor Window</u>	- open a window to monitor MIDI input or output streams
<u>File Conversion Window</u>	- open a window to perform file conversion from other formats
<u>Driver Creator Window</u>	- open a window which is used to create/modify SQ drivers
<u>Add to Cakewalk Toolbar</u>	- add the program to Cakewalks toolbar for easier access
<u>Uninstall Preparation...</u>	- prepare to uninstall - remove elements which the uninstaller can not

File Conversion Window (Utilities Menu)



Purpose:

The File Conversion Window is used to convert files stored by other software packages into Sound Quest file format (ie. Dr.T D-50 Patches). The process is not a simple one and will probably take a fair amount of time to setup so unless you have a substantial number of files to convert.

Converting MidiX Files - If you know that the file(s) are in MIDIX format, please move to the next page for instructions on converting these files.

Converting SMF Files - If you know that the file(s) containing the SysX data is in SMF (Standard MIDI File) format then please move on 2 pages for instructions on converting these files.

Converting No Header/Unison Files - If you know the type of data that the file contains and the data is stored with out header information then please move on 3 pages for instructions on converting these files.

Directions:

Note: The following instructions should be used to convert files you have obtained that are not stored in Sound Quest's file format. If you know that the SysX data you

Warning! To use this window you should:

1. Have a knowledge and understanding of hexadecimal.
2. Be able to spot relevant SysX data in a file.
3. Be able to create complete SysX messages.
4. Be able to read and understand the System Exclusive implementation charts of your instrument.

The File Conversion Window is used to convert files from other manufacturer's formats into Sound Quest's file format so that they can be used by the software.

The steps to converting a file are as follows:

1. Open the File Conversion Window.
2. Load the Sound Quest Driver for the data into the window.
3. Determine the file format of the old file to convert.
4. Write a macro that reads & converts the old file.
5. Select "Convert File to SQ" or "Convert Directory to SQ".
6. Load the new data file to ensure proper conversion.
7. "Save" the new conversion macro to disk for later use.

NOTE: With an existing conversion macro replace 3,4,5 with 8.

8. "Open" an existing file conversion macro.

Step 1. Opening the File Conversion Window.

Use the following steps to access the File Conversion Window:

1. Bring down the "Utilities" menu.
2. Select "File Conversion Window" (item 8).

Step 2. Loading the Sound Quest Driver

The Sound Quest Driver must be loaded into the File Conversion Window because this information is an integral component of every Sound Quest Data File. It also tells the File Conversion Window what kind of data is being converted. A driver can be obtained from two locations, directly from disk or from the Driver List Window.

Loading the Driver from disk:

1. Select on the File Conversion Window to activate it.
2. Display the Convert menu.
3. Select the "Load Driver From Disk" item.
4. Use the displayed File Selector to select the driver.
5. Select OK to load the driver.
6. The driver name will appear below "Convert to:".

Loading the Driver from the Driver List:

1. Display the Driver List Window (if necessary).
2. Select the Driver to load into File Conversion.
3. Select on the File Conversion Window to activate it.
4. Display the Convert menu.
5. Select the "Load Driver From List" option.
6. The selected driver will appear below "Convert to:".

After the driver has been loaded, it will automatically fill in the "Extension" text area with its extension.

Step 3. Determining the File Format

The Sound Quest format file stores much more than just the System Exclusive information. It also stores comments, file search codes, and information on how the data is to be transmitted. Sound Quest's file format also stores complete MIDI System Exclusive messages unlike many other manufacturers which only store the data.

With the release of v6.0, the File Conversion Window can now attempt to write the conversion macro for you. Please keep in mind however that this is only the program's guess at what is correct and it may in fact not be. In the end, you must determine whether the macro is correct or not. To have the program attempt to create the macro, press the "**Build Sample Macro**" button. The program will display a file selector. Use this selector to choose one of the files which needs to be converted. The program will load and analyze the file. If it thinks it has found the SysX component in the file, it will build the appropriate macro. If not, it will give you a warning message that you will need to write the macro yourself. This can be difficult but the following steps should assist you.

The format of another manufacturer's file can be determined in one of these ways:

1. The format is printed in the manufacturer's manual.

2. Use the "Load File to SysX View" menu item to load the file and search for the data within the file.
3. Use some other program or utility you own to view the file and analyze its contents.

Assuming that you must determine the manufacturer's file format, here is one of the best ways to do it:

1. Load file into MIDI QUEST using "Load File to SysX View"
2. Search for names stored within the file.
3. Compare the position of the name(s) in the file to where they are stored in the data and work backwards to determine where the data starts.
4. This method assumes that the SysX data has not been "converted" in any way.

Step 4 - Write a Macro Conversion

If the new automated macro builder was successful, then you will already have a macro written for you at this point. If the software was unable to analyze the data to be converted, you will need to write your own macro.

Clicking the mouse BELOW the "Conversion Macro" title will activate a cursor where the file conversion macro is entered. When the conversion process is started, MIDI QUEST automatically opens the selected data file for reading AND allocates the memory required for a file using this driver.

If you are lucky enough to know that the file you want to convert is in MIDIX format, just press the "Create MIDIX Macro" button and the macro will be created for the data. Otherwise, continue with the process of creating a conversion macro.

Remember, this whole process is not intelligent with very few checks to verify that what you are asking to do is acceptable. Its very easy to crash the computer by giving the system incorrect information.

Every conversion macro will vary according to the specific manufacturer and the type of data which is being converted but as a general rule the steps will occur in this order:

1. Advance the disk file to where the SysX data is (DS n).
2. Read the SysX data from old file -> new file (DR n x).
3. Add SysX header if necessary (FA 6 0 {F0 43 0 0 1 1B}).
4. Add SysX tail if necessary.

NOTE: the bracketed commands show the relevant macro commands which accomplish the required function where n and x are some numeric value.

It may, of course, be necessary to perform multiple disk seeks and reads as well as a wider range of formatting. Obviously it will be necessary to read the section describing the macro language extensively.

WARNING: Make sure that the DR macro call reads data into the allocated memory and NOT beyond or nasty problems can result. The same warning holds true for the FA macro command.

Step 5. Convert the File

To convert the file, select either "Convert File to SQ" or "Convert Dir to SQ". The software takes the following steps:

1. Opens a File Selector
2. Choose the file to convert and Select "Ok"
3. A data file is allocated for the Conversion Driver
4. The selected disk file is opened and prepared.
5. The macro is executed.
6. The Sound Quest file is saved to disk with the extension currently in the "Extension" box in the window.
7. If directory processing Loop to #3 for all files.
note: even for directory processing, you must select at least one file with the file selector

If you wish to convert only those files with a particular extension, enter this extension in the "Ext Select" entry area. For instance, if you only wished to convert files with the ".DDF" extension, you would enter "*.DDF".

WARNING: If a directory of files is being converted, ALL of the files in that directory MUST be of the same type.

WARNING: If there are two or more files with the same name but different extensions, all of the converted files will be lost except the last one converted. The files should be renamed before the conversion process is started.

Step 6 - Load the New File to Verify

After the files have been converted, the new files should be loaded to ensure that the process proceeded properly. This is particularly important when you have just written your own conversion macro.

Take the following steps:

1. Bring down the "Windows" menu.
2. Select the "Edit Window" item.
3. A File Selector will appear.
4. Select the newly created data file and press "Ok".
5. Verify that the data has been converted properly.
6. Transmit the data to your instrument.
7. Verify the data aurally.

Congratulations, you have successfully converted file formats.

Step 7 - Save the Conversion Macro

Once the conversion macro has been tested, you can save it to disk for later use. When the macro is saved to disk, the Driver file in use is also saved with the macro and is automatically loaded with the macro. Before saving the file if you wish, you may add a comment in the comment box so that you may easily keep track of what this conversion macro does in the future.

Take the following steps to save the macro:

1. Activate the File Conversion Window (click mouse on it).
2. Display the "Files" menu.
3. Select the "Save As..." item.
4. Enter a name for the macro.
5. Select "OK" to save the macro.

Step 8 - Load a Conversion Macro

If you want to use a conversion macro you have written and saved or received from someone else, follow these steps to load it:

1. Select on the File Conversion Window to activate it.
2. Display the "Files" menu.
3. Select the "Open" item.
4. Use the File Selector to select the macro.
5. Press "OK" to load the macro.
6. MIDI QUEST will load the selected macro for use.

NOTE: the conversion macro file does not actually save the Sound Quest Driver it is using, only a path to where that driver is located on disk. If you obtain a conversion macro from someone else, make sure that you either have the requisite driver yourself or you obtain the driver as well.

The Convert Menu

When the File Conversion Window is active, the "Convert" menu is available as the third menu strip. It contains four items (three on the PC) used in creating conversion macros.

The Menu Items are:

1. Compile
2. Load File to SysX View
3. Load Driver From Disk
4. Load Driver From List

Compile

1. Enter a macro or a portion thereof.
2. Select "Compile" from the "Convert" menu.
3. Find and correct the error if an error message results.

The "Compile" item runs a test compile on the conversion macro. This option should be run before trying to convert a file from one format to the Sound Quest format. If there is an error, a message window is displayed indicating the line that the error occurred on, along with the character position at which it occurred.

Drop To

This item contains four sub options which controls what happens to files which are dropped in the window.

- a) **SysX View Window** - is the default. Files dropped in the window are opened in a SysX View Window. This is the equivalent of choosing *Load File to SysX View* and selecting a file.
- b) **Convert Using Macro** - runs the conversion process *Convert File to SQ* on each of the dropped files
- c) **Convert as Midi File** - runs the conversion process *Extract from SMF* on each of the dropped files

d) **Build Macro** - runs the *Build Sample Macro* function on the supplied Data File to attempt to create a conversion macro

Each each case, you can read about the appropriate function (on this page) for more information.

Load File to SysX View

1. Select "Load File to SysX View" from the "Convert" menu.
2. A File Selector will be displayed.
3. Choose the file to be converted.
4. The entire file will be loaded into a SysX View Window.

This option provides an effective way to load an ENTIRE foreign data file into MIDI QUEST for viewing in both hexadecimal and character formats. Keep in mind that this is NOT A TRUE DATA FILE and should NOT be saved from the SysX View Window.

Once the data has been placed in the SysX View Window, the window can be used to find the location of relevant data and greatly assists in the creation of conversion macros.

WARNING: Do NOT abuse the capability of loading foreign data files into MIDI QUEST unconverted. They should be used only for viewing. The program can not understand them in any other way. We are not responsible for the results if they are.

Load Driver From Disk

1. Select "Load Driver From Disk" from the "Convert" menu.
2. A File Selector will be displayed.
3. Choose the driver to be loaded.
4. The driver will be loaded into File Conversion Window.

When a new macro is being written, the File Conversion needs a driver to know what kind of data file it will be creating. This option enables any driver to be loaded from disk.

Load Driver From List

1. Display the Driver List Window if necessary.
2. Select the driver for the File Conversion Window.
3. Select "Load Driver From List" from the "Convert" menu.
4. The driver will be copied into File Conversion Window.

When a new macro is being written, the File Conversion needs a driver to know what kind of data file it will be creating. This option enables any driver in the Driver List Window to be copied into the File Conversion Window.

Note: when converting a directory of SysX files with the 32 bit version of the software, the program will use only the 8.3 version of the file name. It will not use the long file name for the conversion. If any of the files use long file names, not only will you see a change of extension but also a change of name. If you wish to retain the full long name version, convert the files one at a time.

See also: [The Utilities Menu](#)

Driver Creator Window (Utilities Menu)

General Make: <input type="text" value="Korg M1"/> Model: <input type="text" value="Combi Bank"/> Defn: <input type="text"/> Company ID#: <input type="text" value="42"/> Filesize: <input type="text" value="14179"/> Headersize: <input type="text" value="6"/> Template V40: <input type="text" value="0"/> Storage Format: <input type="text" value="2"/> MIDI CommCh Ofs: <input type="text" value="2"/> <input checked="" type="checkbox"/> F7 End <input checked="" type="checkbox"/> Fixed Length <input checked="" type="checkbox"/> I/O Macro Mode <input type="checkbox"/> VariSize <input type="checkbox"/> Timeout		Roland (non-Macro Mode) Starting Address: <input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="00"/> Checksum (non-Macro Mode) Type <input type="text" value="0"/> Start <input type="text" value="0"/> #Bytes <input type="text" value="0"/> COfs <input type="text" value="0"/>		Dump Request (non-Macro Mode) # of Bytes <input type="text" value="7"/> MIDI Req Ch Ofs <input type="text" value="2"/> <table border="1"> <tr><td>F0</td><td>42</td><td>30</td><td>19</td><td>1D</td><td>01</td><td>F7</td><td>00</td></tr> <tr><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td></tr> <tr><td>00</td><td>00</td><td>00</td><td>00</td><td></td><td></td><td></td><td></td></tr> </table>		F0	42	30	19	1D	01	F7	00	00	00	00	00	00	00	00	00	00	00	00	00				
F0	42	30	19	1D	01	F7	00																						
00	00	00	00	00	00	00	00																						
00	00	00	00																										
Bank <input checked="" type="checkbox"/> Bank <input type="text" value="100"/> Voice Size: <input type="text" value="124"/> Voice Name Offset: <input type="text" value="0"/> Voice Name Size: <input type="text" value="10"/> Voice Name Format: <input type="text" value="1"/> Voice Number Format <input type="text" value="0"/> Adjust: <input type="text" value="0"/> Advance <input type="text" value="0"/> Voice Ofs <input type="text" value="0"/> Voice Bytes <input type="text" value="124"/> Bank Num Ofs <input type="text" value="0"/> <input type="checkbox"/> Names In Voices		Receive <input type="text" value="S 7 { F0 42 30 19 1D 0 F7 } TR 7 R 14179"/>																											
		Transmit <input type="text" value="FA 6 0 { F0 42 30 19 4D 0 } MX = 247 FX 14178 T 14179"/>																											
		Audition <input type="text" value="S 8 { F0 42 30 19 4E 1 10 F7 } T 8 D 10000 GA 148 GN FA 5 0 { F0 42 30 19 49 } GC 0 5 142 MX = 247 FX 147 T 148 GF 148"/>																											
		Patch <input type="checkbox"/> Patchize Patch Min <input type="text" value="0"/> Patch Max <input type="text" value="0"/> Patch Ofs <input type="text" value="0"/> Dump Req Ofs <input type="text" value="0"/>		Misc Comm Ch Min <input type="text" value="1"/> Comm Ch Max <input type="text" value="16"/> Driver ID <input type="text" value="86"/> Link ID <input type="text" value="89"/> Child ID <input type="text" value="0"/>																									
		<input checked="" type="checkbox"/> Prepared Format <input type="checkbox"/> Roland Handshake <input checked="" type="checkbox"/> Is a Parent <input type="checkbox"/> Is a Child Directory <input type="text" value="M1"/> File Ext <input type="text" value="CBk"/> Icon <input type="text" value="Multi Bank"/>																											



Purpose:

The Driver Creator Window is used to create and modify the instrument drivers used by the program. Driver creation, on whole, is not easy and requires a certain amount of technical ability and knowledge of System Exclusive data. However, this editor is the same one that Sound Quest uses to create all of the drivers for use with its products. At the back of this online help file are instructions for creating your own drivers. Please press [here](#) to jump to this location.

Directions:

The Creator Menu

When the Driver Creator Window is active, the "Creator" menu is available as the second menu strip. It contains three entries to aid in the development of drivers.

The entries are:

1. Compile
2. To Driver List - Test
3. Load Driver from Data
4. Make Driver...
5. Make Ini...
6. Load Driver from Disk

1. Compile

1. Enter a macro or a portion thereof.
2. Select "Compile".
3. Find and correct the error if an error message results.

The "Compile" item runs a test compile on all active macros. It is recommended that a test is run after each macro is completed to ensure that there are not any problems.

If there is an error, a message window is displayed indicating the line that the error occurred on, along with the character position at which it occurred.

2. To Driver List - Test

1. Complete the creation of the Driver.
2. Select "To Driver List" from the menu.
3. Go to the Driver List Window.
4. The new driver will be at the bottom of the list.
5. Test it with whatever option is most appropriate.
6. If there are problems return to the Driver Creator to correct them.

The "To Driver List - Test" item adds the new driver to the bottom of the current Driver List so that it can be tested. If it does not work properly, it can always be deleted from the list and you can return to the Driver Creator Window to make whatever corrections are necessary.

PC NOTE: Because of memory restrictions, it was necessary to place the Driver Creator and File Conversion Utilities into a separate product so this option is not available. Once a new Driver is complete, it should be saved to disk, load MIDI QUEST into your computer, load the new driver and test it.

3. Load Driver from Data

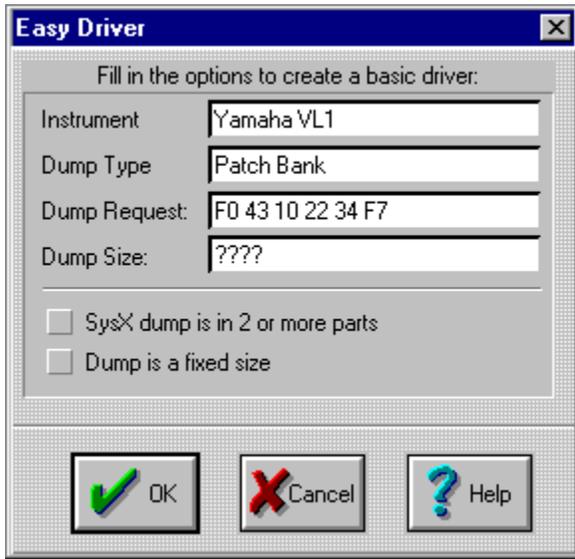
1. Select "Load Driver From Data" from the menu.
2. Select the DATA FILE which has the desired driver.
3. The data file's driver will be loaded into the Driver Creator Window for editing.

We at Sound Quest were determined to build all of the tools we would need to create MIDI QUEST into MIDI QUEST. One of the things we required was a way to access old Sound Quest drivers which are built into our data files. This option provided that capability. Its sole use in the normal course of events would be a situation where you obtained a data file from someone using a driver that you did not have and you

wanted the driver for your own use. This option will extract the driver.

4. Make Driver...

This menu selection provides you with the easiest way to create your own driver for MIDI QUEST or MIDI QUEST Jr. It contains only 6 basic parameters which you set. From this information, a basic driver is created so that the program can request, store and return SysX data to an instrument. Choosing this entry brings up the "Easy Driver" dialog. Just fill in the parameters and it will create a driver for you.



Instrument: Enter the company and model name of the instrument (eg. Yamaha DX7).

Dump Type: Enter the type of dump this driver will store (eg. Patch Bank).

Dump Request: Enter the dump request that is used to request the SysX data from the instrument. If you do not enter this information, the program will not be able to automatically request the SysX from your instrument. You will have to trigger the dump manually from the front panel. This request must be entered in hexadecimal as you find it in your manual. Each byte must be separated by a space (eg. F0 43 32 0 10 F7).

Dump Size: If the SysX coming from the instrument is ALWAYS the same size and you know what this size is, replace the question marks ("????") with this number (in decimal). This number must be the total size of the dump. If you do not know the size of the dump, leave the question marks and the program will allocate enough space to store your dump.

SysX Dump is in 2 or more parts: Check this parameter if the SysX data you are requesting from the instrument arrives in two or more SysX blocks. For example, if you have requested a patch bank and the instrument transmits this bank as one large dump, leave this option unchecked. However, if the instrument responds by sending a series of patches, each in its own SysX message, you should check this option.

Dump is fixed size: This option should be checked if you know the size of the dump that the instrument will return in response to a SysX dump request. It enables the program to verify that the correct number of bytes were received by the computer and that there were no errors during communication. When you enter a size into the "Dump Size" parameter, this check box will automatically be set. If the dump is not a fixed size, unselect it.

5. Make Ini...

Since v5.0, Midi Quest, Solo Quest, and Midi Quest Jr. have operated from an instrument centric view, that is you don't load individual drivers into the Driver List but an entire instrument. Instrument definitions are created by using .INI files that are stored in the INSTR directory. To create an INI file for you new driver, save the driver in a directory under the INSTR directory. *You must save the driver before using this function.* Now, choose *Make INI* and a File Selector will appear. Enter the name of the instrument for the INI file and press OK.

If you now switch to the Driver List Window and choose Drivers/Add Instrument you will see the instrument name that was just added. Double click on it and the driver you created is added to the Driver List. If you wish to add additional drivers to the instrument INI file you just created, you will need to click on the following line for more information on creating your own custom INI files.

[Press here for more information on defining you own custom .INI files.](#)

6. Load Driver from Disk

1. Select "Load Driver From Disk" from the menu.
2. A file selector appears, us it to select and load an existing driver.
3. The data file's driver will be loaded into the Driver Creator Window for editing.

See also: [Utilities Menu](#)

Midi Cleanup (Midi Menu)

Purpose:

Midi Cleanup, when checked, will automatically open the selected MIDI Ports when the program becomes active and will close the ports while MQ is inactive.

Many interfaces provide Windows MIDI drivers which will only allow one application to access them at a time (eg Creative Labs Sound Blaster drivers). In order to have two or more MIDI applications running simultaneously, it is often necessary to manually close particular Windows MIDI drivers in one program so they may be opened by another application.

This feature solves this problem by allowing the program to share the MIDI Ports with other applications automatically. This is accomplished by having the program close the MIDI Ports when another program is activated. For example, when your sequencer window is activated. Later, upon returning to the program, the ports are reopened for use.

With this capability, both application can maintain their optimal port configurations without interfering with other programs. **Note:** for this feature to work correctly, the other programs running must have the same type of feature and it must be enabled.

Directions:

To enable this feature, set the check mark beside *Options/Midi Cleanup*

See also: [Options Menu](#)

of Notes

This parameter determines how many notes are played when a chord or sequence is triggered in the program. If for example **4** (four) notes are selected, the first four notes defined in the Tones Window will be played as a sequence or grouped together to form a four note chord.

The range of this parameter is 1 through 16 so a sequence or chord of up to 16 notes may be created.

To change the number of notes in the chord and sequence, click the mouse on the parameter and highlight a new value.

Playback Speed

Playback speed determines how fast the sequence plays if a sequence is triggered in the program. Alternately, if a chord is triggered it determines the duration of the chord.

The range of the playback speed parameter is 1 through 16 where 1 is the fastest playback speed and 16 is the slowest.

To change this value, click the mouse on the parameter and highlight a new value.

Editing Note #

This parameter is used to select which note is currently being displayed and edited by the two parameters below this one.

The range is between 1 and 16, which covers the total number of available notes.

To select a note to edit, click the mouse on this parameter and point to the new note number. When the mouse button is pressed, the Midi Note # and Note Velocity parameters will change to represent the current settings for that note.

Number Library (Library)

Purpose:

Number Library opens a dialog which allows you to select which column will be used to number the Library.

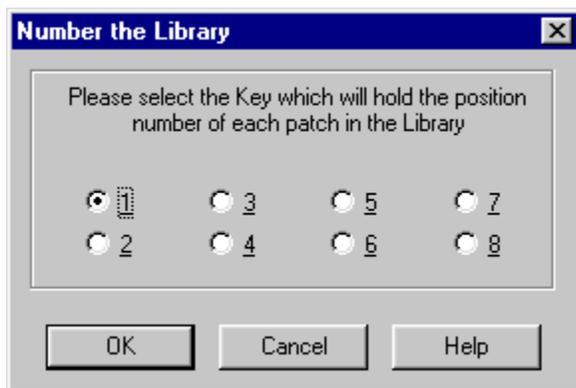
If you have a situation where you would like to be able to quickly order your Library in a particular fashion, order the Library as you would like it and then use this function to add a key number to each of the Patches in the Library. At a later date, to return to this order, all you need do is sort on the particular key.

Directions:

To order the Library:

- (1) Organize the library into the order you would like
- (2) Choose *Library/Sort/Number Library*
- (3) Use the displayed dialog to select the key that is used to number the Library

The Number Dialog



Purpose:

The Number Dialog allows you to select which key will hold the current sorting order of the Library. Select the key (1 - 8) and press the OK button.

See also: [Library Window](#)

Midi Note #

This parameter shows which note pitch has been chosen for the selected note number.

For example, if you use the 'Editing Note #' parameter to select note 3. This parameter, 'Midi Note #' will show the note pitch assigned to note #3 in both pitch/octave form (C4) and MIDI note number form (48).

The pitch of the note can be set to any any one of the 128 MIDI values (- 127). To change the value, click on the parameter and drag the mouse to the new desired setting.

Note Velocity

This parameter shows the note velocity chosen for the selected note number.

For example, if you use the 'Editing Note #' parameter to select note 3. This parameter, 'Midi Velocity' will show the note velocity assigned to note #3.

The range of the velocity is 1 through 127. To change the value, click on the parameter and drag the mouse to the new desired setting.

Select All (Bank)

Purpose:

Select All selects all of the Patches in a Bank for further processing. An example would be to copy all of the Patches from one Bank to another.

Directions:

To select all of the Patches in a Bank:

(1) Choose *Bank Edit/Select All*

See also: [Bank Editor Window](#)

Patch Edit (Bank)

Purpose:

Patch Edit opens the Patch Editor for the Patch you have selected in the Bank so that the Patch may be edited.

In virtually all instances, a Bank will have a driver that corresponds to one Patch in the Bank. For example, if there is a Voice Bank, there will also be a single Voice driver. When a Patch is selected in a Bank for editing, the program finds the Patch driver in the Driver List, creates a Patch and opens an editor for it. The Bank then transfers the selected Patch into the editor. Use the Editor to edit the Patch parameters

Remember, to edit a Patch, the individual Patch driver **must** be in the Driver List. For example, to edit a Patch in an M1 Patch Bank, the M1's Patch driver must be in the Driver List.

Note: In some instances, there will not be an individual editor for the data type. This occurs when Editors are not created for a particular instrument, for example. In this case, attempting to edit the data will only result in placing the data into a SysX View Window.

Directions:

To send a Patch in a Bank to a Patch Editor for editing:

- (1) Select the Patch to edit
- (2) Choose *Bank Edit//Edit/Patch*

See also: [Bank Editor Window](#)

See also: [Patch Editor Window](#)

Play Chord (Bank)

Purpose:

Play Chord is used to play the Patch currently in the instrument. This allows you to listen to the Patch without going to the instrument to play it. Play Chord is most commonly used after auditioning a Patch in order to listen to it.

The chord that is played is defined in the Properties Tones Tab. The chord is sent on the MIDI channel assigned to the Bank as set in the Settings Dialog.

Directions:

To play the chord defined in the Tones Window:

(1) Choose *Bank Edit/Audition/Play Tones Chord*

See also: [Bank Editor Window](#)

Play Sequence (Bank)

Purpose:

Play Sequence is used to play the Patch currently in the instrument. This allows you to listen to the Patch without going to the instrument to play it. Play Sequence is most commonly used after auditioning a Patch in order to listen to it.

The sequence that is played is defined in the Properties Tones Tab. The sequence is sent on the MIDI channel assigned to the Bank as set in the Settings Dialog.

Directions:

To play the sequence defined in the Tones Window:

(1) Choose *Bank Edit/Audition/Play Tones Sequence*

See also: [Bank Editor Window](#)

Auditionable? (Bank)

Purpose:

Auditionable? determines whether Patches can be auditioned from a Bank or not.

In most cases, it is possible to audition the Patches in a Bank. In a few cases, however, due to instrument design, it may not be possible. Choose the Auditionable? function. If the displayed dialog box indicates the Bank is not auditionable, you will have to transfer the entire Bank to the instrument and listen to it there. If the displayed dialogue box indicates the Bank is auditionable but you are still having trouble, verify the Comm Channel, MIDI Channel, and MIDI Port settings in the [Settings](#) dialog.

Directions:

(1) Choose *Bank Edit/Auditionable?*

See Also: [Bank Editor](#)

Load All Banks and Patches (Library)

Purpose:

This function provides an extremely fast way to build or update a library.

When this option is selected, Midi Quest will automatically look at all data files stored in the instruments data directory (eg. for the M1, the MQDATA\M1 directory is searched). All individual Patches and Patch Banks containing the same type of data as the list are loaded and incorporated into the library.

Midi Quest will automatically ensure that no duplicates are added to the library.

If you perform this function on an existing Library, the only Patches added to the Library are those which are not already in the Library. These is an extremely fast methosonly those Patches found in the directory that are not currently in the Library

This is an extremely easy method of updating your Library so that it contains all of the patches for an instrument.

Directions:

To add all patches from either individual Patch files or Patch Bank files to the current library:

(1) *Choose Library/Disk/Load all Patches and Banks.*

See Also: [Library Window](#)

SysX View (Patch Editor)

Purpose:

SysX View moves the Patch's SysX data into a SysX View Window where the data can be viewed in its raw hexadecimal form. This feature is useful for viewing data while developing and testing instrument drivers in MIDI QUEST.

Directions:

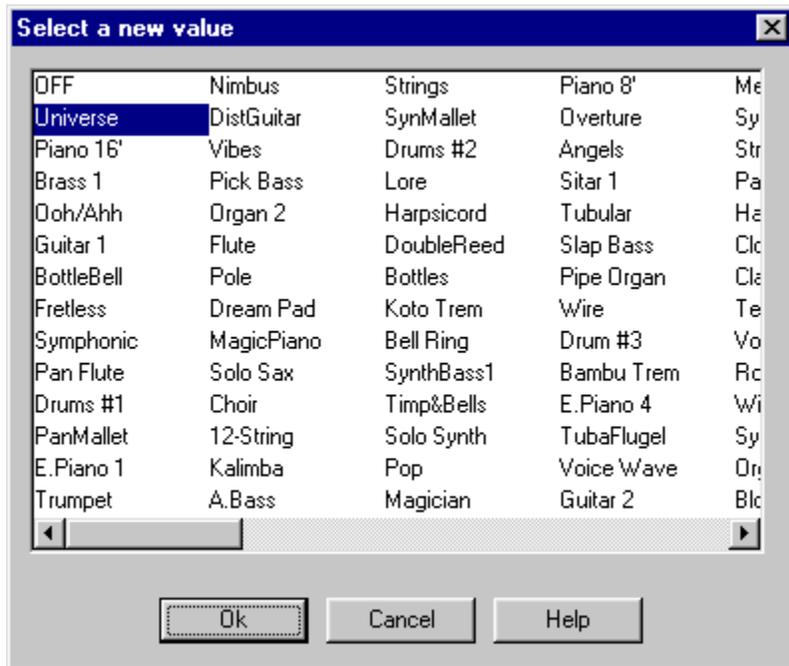
Choose *Sngl Edit/Edit/Open/New SysX View*.

Note: not available in MQ Jr.

See also: [SysX View Window](#)

See Also: [Patch Editor Window](#)

Parameter Selector Dialog (Patch Editor)



Purpose:

The Parameter Selector Dialog is a special pop-up selector. Some parameters are displayed as text. For instance, the waveform for an LFO will be one of: triangle, saw up, saw down, random, or sine. To change the value of the LFO, you can click and drag, or double click on the current value and use the above dialog to select a new value. Each option you click on is automatically sent to the instrument so that you can try a number of different options before selecting one. Double click on the new value or press the OK button to accept the new value. Press the Cancel button to return to the original value. This window is ideal for viewing and editing parameters with a wide range of text values.

Directions:

- (1) In your Patch editor, double click on a parameter which is displayed as text but is not a name
- (2) The Parameter Selector dialog will be opened for viewing
- (3) To choose a new value for the parameter, double click on the value or, alternatively single click and press OK

See also: [Patch Editor Window](#)

Move To (DBase)



Purpose:

Move To moves one or more data files into any of the following destination windows: Bank Editor, Patch Editor, Group, DBase, or Library window.

Directions:

To move one or more data files:

- (1) Choose the data files to move
- (2) Choose *DBase/Move To...*
- (3) The selected data files are moved to the destination window

See Also: [Move To Dialog](#)

See Also: [DBase Editor](#)

DBase Contents (DBase)

The DBase can hold up to 37,000 data file entries (limited only by available RAM).

Data Files can be dragged and dropped between the DBase and other windows in the program. For more information, please see [Dragging and Dropping](#).

Individual entries in the DBase can be reordered by following these steps:

- 1) Choose the data file to move;
- 2) hold down the ALT key on the keyboard;
- 3) use the up and down arrow keys to move the data file to the desired location.

Each data file is shown as a row in the DBase. The first row of the DBase contains the column titles: Name, Type, Comm, Port Out, Date, and File Size. The first column has no title but will show a disk icon if the entry is linked to a disk file instead of actually being stored in the DBase.

Link

The first column is untitled, but will show a disk icon if the row contains a disk linked data file. Disk linked files are references in the DBase to other Sound Quest data files stored on disk. These files are only referenced by the DBase and are not actually stored in the DBase. This allows any DBase or Library to have access to the same Patches and Banks. Any changes to the disk based data file will also be reflected in the DBases, Groups and Libraries that have links to the source data file. All of the text in the row will appear in blue instead of black. For more information on linking to data files, please see [Link a Data File](#), [Save a Data File and Link](#), and [Resolve Links](#).

Name

The Name column displays the name assigned to the data file. If the data file has already been named when the data file is added to the DBase that name will be retained. If the data file is loaded directly from the instrument into the DBase or is still named "New", the program will create a name for the data file based on the name of the driver. The name can be changed in the Settings Dialog. Select the data file and choose the Settings button.

Data Type

The Type column shows the type of SysX data stored in the data file. For example, a Korg M1 Patch Bank data file. The Data Type parameter is not editable and is meant to give a visual indication of the different types of data contained within the DBase. This makes it easy to spot different types of data from specific instruments.

Comm Channel

The Comm column shows the current Communication Channel setting for the data file. The Comm parameter can be changed using the [Settings Dialog](#).

Port Out

The Port Out column shows which MIDI OUT Port the data file will be transmitted on. The Port Out parameter can be changed using the [Settings Dialog](#).

Date

The Date column shows the date the data file was created or last edited. The Date parameter can not be edited but will change each time the data file is edited.

File Size

The File Size column shows the size of the data file stored within the DBase. The File Size parameter can not be edited and is included for information purposes only.

See also: [DBase Window](#)

Morph (Bank)



Purpose:

To Morph, choose any two Patches in a Bank and "morph". the result is a Bank of new Patches which incrementally transpose from the first Patch through to the second Patch. While this function can be applied to any kind of Patch in a Bank, performing a morph on Multi Patch does not serve any practical purpose. The function is best suited for use with Patches that create an actual sound (may also be called Tones, Sounds, etc).

This function is ideal if you are trying to find a sound that is somewhere between two given Patches. After the Bank is created. Click on the various Patches to audition them.

Technical Note: in the case of some sound Patches, there are certain parameters where it does not make sense to "morph" the values. In these cases, the program sets the first half of the Patches to the value selected in the first Patch and the second half of the parameters set to the second half. An example of this situation would be the operator selector in Yamaha's 4-op and 6-op instruments. It does not make sense for this parameter to be morphed. In most cases, Patch names do not morph. The ids of the parameters which are to be blocked from sliding are contained in an .ini file with the same name as the instrument's Patch editing template. (eg. Patch.ini for Patch.sqt).

Note: Print XReference is only available in Midi Quest and Solo Quest.

Directions:

To morph from one Patch to another:

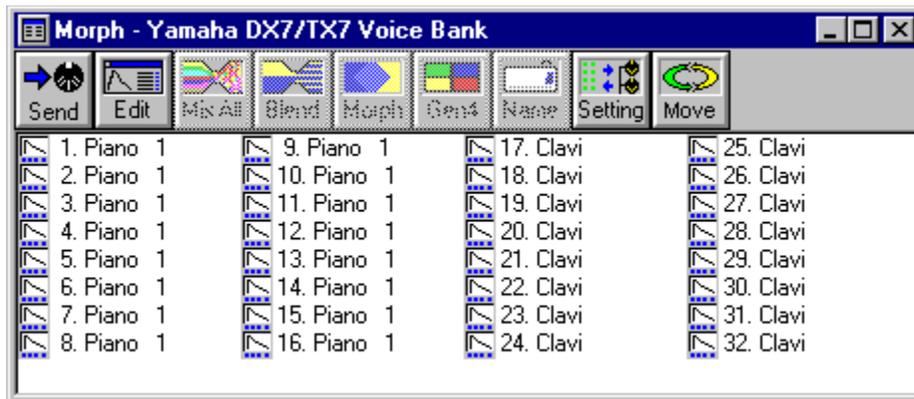
- (1) Click the mouse on the first Patch
- (2) Hold down the Cntrl key and select the second Patch
- (3) Choose the 'Morph' button or *Bank Edit/Morph*

Example:

The following example shows the result of choosing two Patches



and then selecting the Morph function. Note: In most cases, Patch names do not morph.



See Also: [Morph in the Patch Editor](#)

See Also: [Morph in the Library Window](#)

See Also: [Bank Editor Window](#)

Technical Docs

The Driver Creator Window is used to create new drivers for the Driver List Window. To open the Driver Creator Window, [click here](#).

Sound Quest doesn't claim that creating drivers is easy. In fact it can be quite difficult. The information provided by the manufactures is often incomplete and occasionally incorrect. Additionally, the information is not provided in a standardized format. The layout of the SysX information is different for every company and often the documents vary between instruments from the same company. Reading SysX documentation is almost an artform in itself. With that said, we are ready to proceed.

The drivers provide complete control for receiving and transmitting System Exclusive information from virtually every instrument. The driver also includes the information necessary to audition Patches and perform Bank editing where appropriate.

WARNING: To use this window effectively it is necessary to be able to read and understand the System Exclusive implementation charts for your instrument. Also it is necessary to be able to work in the hexadecimal numbering system when entering SysX information that is sent to the instrument.

If you are just learning about creating drivers, we strongly suggest that you obtain System Exclusive documentation for an instrument which Sound Quest is already supporting with Drivers. Load the existing Sound Quest Driver into the Driver Creator Window and compare the documentation with what is entered in this window. If you use one of our drivers as an example, and use the manufacturers documentation as you work through this chapter, you should find it much easier becoming comfortable in obtaining the kind of information necessary to create a Driver.

Entering Information in the Driver Creator

When entering information in this window take care to:

1. Most numeric information is entered in decimal EXCEPT
2. Enter MIDI strings in hexadecimal format. (including Dump Request)
3. Numbers type is indicated by the ending (in this manual):
 - H = Hexadecimal (eg. 30H)
 - D = Decimal (eg. 4104D)

Note : H and D are NOT entered in the window!

Using the Driver Creator Window

In order to create a new driver it is necessary to fill in all of the information in active areas (the areas in which a cursor will appear when the parameter is clicked on). The window is divided into nine sections. Each section will be discussed separately with examples of when the section will be used.

Chapters

The following are the chapters in the Technical docs. They relate directly to the various portions of the Driver Creator WINDOW:

[General](#)
[Bank Edit](#)
[Roland](#)

Check Sum
Dump Request
Macros
Macro Language
Midi Errors
Disk Errors

General Information (Technical Docs)

It is ALWAYS necessary to enter General information.

Make: the manufacturer and instrument (eg. Roland D-10)

Model: the type of data (eg. Tone)

Defn: where the editing template is stored for the data
(eg. D-10/Tone.sqt)

- 1) this should be left blank unless a template is available to edit the data
- 2) this contains the path and a file name to the data file's template. It is appended to the path set by the "Options" menu selection "Set Template Path" to load the editing template.
- 3) look in the subdirectory structures on your disk where the templates are stored to see how template files are organized.

Company ID#: the company's SysX ID number (in hexadecimal)
(eg. Roland = 41H, Yamaha = 43H, Korg = 40H)
this is byte 2 of a SysX dump.

FileSize: the amount of memory to be allocated for the data
(eg. 246D)
if the data is of variable size it MUST be
the MAXIMUM size

Header Size: the number of bytes before the first byte of data
(eg. 6D)
used during editing
used to move Patches in and out of Banks
(ie copying starts at this offset)
NOTE: can be left as 0D if neither above option
is used

Template V40: Value to be placed in Variable 40 for Patch edit
set to 0D unless needed for template control

Storage Format: the format in which the data transmitted
enter (0 - 8) (0 default and most common)

0) 7-bit format

1) Nibble format where 8 bits are sent as
two 4 bit nibbles
(ie HHHHLLLL -> 0000LLLL 0000HHHH)
Low nibble sent first

2) Korg M1 packed format
used by M1, M3R, T1-2-3
takes 7 8-bit bytes
strips the top bit of each to make
a new byte
sends 8 bytes (new byte first)

- 3) TX802 format
hopefully never to be used again
- 4) DX7II/TX802 fractional scaling
nibblized and OR'd with 30H.
- 5) 14-bit word divided into two bytes
- 6) Reverse nibble format
two 4 bit nibbles - high nibble first
(ie HHHHLLLL -> 0000HHHH 0000LLLL)
- 7) Alesis HR-16 Format
7 -> 8 byte storage format
- 8) Ensoniq VFX Performance format
Inverted Reverse nibble byte format
- 9) 14-bit high-low format
- 10) 14-bit low-high format
- 11) Alesis QSynth byte format
- 12) Alesis QSynth 2 byte format
- 13) Alesis QSynth bit format
- 14) Korg Poly 6 format
- 15) Ensoniq MR 4->5 byte format

- F7 End:** Toggle 'Yes' or 'No' whether data ends with F7H
almost always 'Yes'
- Fixed Length:** Toggle 'Yes' or 'No' whether data length is constant. This is almost always 'Yes' except for Sequence dumps which vary in size.
- VariSize** Toggle 'Yes' or 'No' whether all SysX received is of a fixed length. If data is requested and a valid response is a dump smaller than that requested, this option should be checked. (See Yamaha SY77 or Kawai K5000 series)
- TimeOut** Toggle 'Yes' or 'No' whether if there is not response to a requested block of SysX whether the software should continue with the next request (used by Kawai K5000 series)
- I/O Mode:** Whether a Macro is used for Receive/Transmit
Sgl - Single Mode
uses Dump Request (if any) and expects the complete data transmission in response

Ext - Receive & Transmit with Macro

**MIDI
Comm Ch
Ofs:** If the MIDI Channel Ofs is greater than 0 then the Communication Channel set in the Driver Window is placed in the byte selected. The byte offset starts at 0. (eg. Yamaha uses byte 2 for the communication channel)

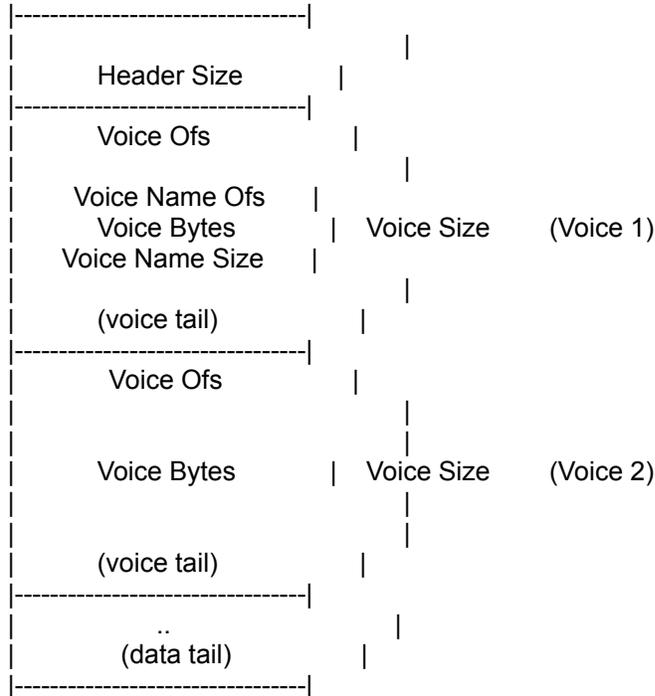
**MIDI
Comm Ch
Ofs:** Most System Exclusive information is channel sensitive. This could either be to the MIDI channel or some other type of channel (for example Roland's UNIT #). The channel is called the Communication Channel (Comm Ch) in MIDI QUEST. This value provides a byte offset for where the channel information is to be stored. If the byte offset is > 0 then the communication channel information entered in the Driver List Window or in an Settings dialog is automatically formatted into ALL system exclusive transmissions made to the instrument. (eg. Yamaha uses byte 2 for the communication channel)

Bank Edit (Technical Docs)

The Bank Edit parameters control the Bank Editing Window and Library Display.

Bank Edit Memory Map

The following is a memory map of a Bank which will help in entering the correct parameters to perform Bank editing.



Bank/Library Editing Parameters

The Bank parameters must be set correctly for Patch drivers as well if a Patch Library is to be created using the driver. For a Library to be created, the number of Voices must be set to '1' and Bank set to 'No'.

- Bank:**
 - Toggle 'Yes' or 'No'
 - if this is a Bank, make it 'Yes'
- #Voices:**
 - The number of voices or Patches in the Bank (ie 64D)
 - set to 1 for a Library
- VoiceSize:**
 - TOTAL number of bytes used by each voice
 - For copying, swapping and pasting Patches
 - This should include all bytes relating to a particular Patch regardless of whether they actually contribute to sound creation
- Voice Name Offset:**
 - The byte position from the start of

the sound parameters where the name is to be found.

- the count MUST start from 0
- (eg. 0D for Tone, 145D for DX7 Voice)

Voice Name Size: - The number of characters in the Voice
- (eg. 10D)
NOTE: IF NAMES IN VOICES IS SET 'YES' THEN SET THIS PARAMETER TO '10' (NO LARGER!!!!)

Voice Name Format: - A special entry to determine the Bank numbering scheme and accommodate non-ASCII standard ways of storing the name
- current range is 0 -> 9

- 0) ASCII characters with 1 -> #Voices
Bank numbering
- 1) ASCII characters with 0 -> #Voices - 1
Bank numbering
- 2) ASCII characters with Roland 1-1 -> 8-8
Bank numbering
- 3) D-50 character conversion and display
- 4) Emu Proteus format
- 5) Matrix 6/1000 character conversion and display
- 6) MKS-50 voice format
- 7) nibblized in a regular voice format
- 8) MX-8 voice format
- 9) SQ-1 voice format

Names in Voices: - normally set to 'No'
- if the Bank data does not have space for Names within the data itself, it is necessary to allocate memory for names by setting this parameter to 'Yes'. Names can be given to each Patch by using the Bank menu "Fill Name" option.

NOTE: THE ABOVE PARAMETER IS A MEMORY ALLOCATION PARAMETER! IF THE DATA HAS ITS OWN SPACE FOR NAMES, SET THIS TO 'NO'. IF THE Bank DOES not HAVE SPACE FOR NAMES IN ITS OWN DATA, SET THIS TO 'YES'.

Adjust: - used for displaying Banks which do not have symmetrical data. Every "Adjust" bytes the Bank data pointer will advance by "Advance" bytes.

Advance: - used for displaying Banks which do not have symmetrical data. Every "Adjust" bytes the Bank data pointer will advance by "Advance" bytes.

Voice Ofs - the number of bytes between the start of Patch and where the actual editable parameters start

Voice Bytes

- the number of accessible bytes

The last two parameters are very important for ALL of the Bank editing functions which deal directly with the data, such as all the randomizing functions.

The "Voice Ofs" parameter is necessary to make up the difference between where the Header Size parameter points within the data and where the first byte is that actually makes up the sound. These two locations may not actually be the same since there are sometimes informational parameters placed before the data.

The "Voice Bytes" parameter is required for all Patch randomization features. Because of the varying data storage formats, the number of accessible bytes in a Patch may not be the same as the number of bytes used by the Patch. For example, in a nibblized format there may be 75 accessible bytes which actually uses 150 bytes because each byte is divided into two nibblized bytes. The "Voice Bytes" must contain the number of ACCESSIBLE bytes.

Both of these parameters must be correctly entered to randomize Patches, move single Patches into parameter edit windows, move Patches from parameter edit windows into the Bank, and to audition Patches.

Roland (Technical Docs)

MIDI QUEST contains special Roland System Exclusive communication code to make working with these instruments easier. If the I/O Mode (in General) is set to 'Sgl' and the Company ID# = 41H, then the Roland parameters will become active. They MUST contain the starting address of the data to be loaded in Roland's format. This information is copied directly into bytes 5 -> 7 of the request for data.

When Roland communication takes place in this mode all header and tail bytes are stripped from the data before it is stored in the data file. The File Size parameter of the Driver should reflect this fact. Please see the Roland D-10 Tone Bank driver for a good example.

Check Sum (Technical Docs)

If there is a check sum routine for the data, it will be run each time before the data is sent out via MIDI (note: when using macros this must be a component of the macro). There are four parameters which control the Check Sum.

Type: there are thirteen predefined check sum types

- 0) Yamaha format 2's complement check sum
(sum bytes and take 2's complement)
- 1) Nibble format where checksum is divided and
COFs stored in two bytes COFs & COFs + 1
(combine nibbles, sum bytes, nibblize to
store)
- 2) Nibble format
(combine nibbles, sum bytes, strip high bit,
store)
- 3) K1 format
(sum bytes, add A5H, strip high bit, store)
- 4) Regular Checksum
(sum bytes)
- 5) 1's Complement check sum
(sum bytes and take 1's complement)
- 6) VZ1 format - combine nibbles and sum
- 7) K5 checksum format
- 8) SY22/35 Multi Bank checksum
- 9) SY22/35 Voice Bank checksum
- 10) Sum (sum bytes, strip high bit, store)
- 11) SY22 Voice checksum
- 12) K3 checksum (combine nibbles, sum, store nibbles)
- 13) 14 bit sum and store

NOTE: If the equipment uses a checksum other than one of these, communication will have to use a macro so that the checksum can be customized.

#Bytes - the number of bytes to perform the checksum on
- set to 0D if there is no checksum
- if using a Nibble format checksum, the number of
bytes operated on will be double that of the
number entered as a result of the nibble format

Start - the number of bytes from the start of the data
where the checksum calculation will start

COFs - the number of bytes from the start of the data
where the calculated checksum will be stored

Dump Request (Technical Docs)

In 'Sgl' mode (in General parameters) a Dump Request will be sent before data is requested.

- | | |
|------------------------|---|
| # of Bytes | <ul style="list-style-type: none">- the number of bytes to send a Dump Request- set to 0 if no request- the request can be up to 20 bytes long- (ie 5 for Yamaha DX Voice) |
| MIDI Req Ch Ofs | <ul style="list-style-type: none">- the byte offset to store the Communication Channel making a Dump Request- set to 0 if the data is not channelized- remember the bytes are ordered 0->#bytes-1- (eg. 2 for Yamaha DX Voice) |
| Dump Request | <ul style="list-style-type: none">- below the Ch Ofs there is space to enter the data's dump request- it MUST be entered in HEXADECIMAL- F0 43 20 0 F7 for a Yamaha DX Voice- note: each number must be entered separately with a <RETURN> afterwards |

Macros (Technical Docs)

Receive/Transmit Macros

When I/O mode is 'Ext' a macro must be entered in Receive and Transmit to control the collection and transmission of the desired System Exclusive data. See the full explanation of how to write macros below.

For macro language information, click on the macro language.

Audition

When the Data is a Bank, Audition is enabled and it is possible to enter a macro which will transmit an individual Patch selected from the Bank to the instrument so it can be heard.

The selected data is passed to the Audition macro as a chunk of memory Voice Size large. In the macro it will be necessary to allocate a new block of memory, copy the data into the new block, and then add the proper header and tail information (including checksum if required) for the transmission of a Patch. The newly created Patch can then be sent to the instrument.

For macro language information, click on the macro language.

Patch Parameters (Technical Docs)

The five parameters relate directly to control over the Pch# parameter in the Driver List Window and Settings Dialog. This parameter is typically used when it is possible to select which Patch is to be loaded or stored as a function of the data and/or Dump Request. If you want the Pch# selected in the Driver List Window or Settings Dialog formatted into the data, turn Patchize on (set 'Yes')

Patchize	- Toggle 'Yes' or 'No'
Patch Min	- the lowest Patch number which can be requested
Patch Max	- the highest Patch number which can be requested
Patch Ofs	- the offset where the Patch# is stored in the Data
Dump Ofs	- the offset in Dump Request where Patch# is stored

Miscellaneous Parameters (Technical Docs)

Comm Channels

These parameters give the maximum ranges that the Communication Channel can assume. The parameters are used in the Driver List Window and the Settings Dialog. In many MIDI instruments the Communication Channel will be the same as the MIDI channel with a range of 1 - 16. However, there are exceptions such as Roland LA instruments (except D-50) which use a UNIT # that ranges from 17 - 32 and Yamaha DX7II/TX802 which use a Device # that is independent of the MIDI Channel.

Channel Min - the lowest possible communication channel
Channel Max - the highest possible communication channel

Roland Handshake

When using Roland Communication in 'Sgl' communication mode, it is necessary to distinguish whether data communication required handshaking. This parameter controls whether it is used or not.

Handshake - toggle 'Yes' or 'No'

Prepared Format

If 'Format' is set to 'Yes' then MIDI QUEST allows an empty edit file to be added using the "Add Empty Data File" in the Driver List Menu. In order for MIDI QUEST to work properly however, the driver must use macros in the extended mode and perform all SysX header and tail formatting in the Transmit Macro. For example, a DX7 Voice driver must add the header 'FA 5 0 {F0 43 0 1 19}' macro and tail macro 'K 0 0 6 155 161 MX = 247 FX 162'. Note: macros will be discussed later.

IDs

There are 2 ID parameter, 'Driver ID' and 'Link ID'. These numbers are used to find companion drivers in the Driver List and give MIDI QUEST these important features: the ability to edit a Patch from within a Bank, to disassemble a Bank into a Patch library, and the ability to create a Bank from a library of Patches. These functions require a Patch driver to be able to find a Bank driver and vice versa. The Driver ID should be unique for each Driver. Sound Quest reserves all driver numbers below 16,000. The Link ID should contain the Driver number of the relative driver. For example a Patch driver has the number 1200 and the Bank driver has a number 1201 then the settings would be as follows:

	Driver ID	Link ID
Patch Driver	1200	1201
Bank Driver	1201	1200

Macro Language (Technical Docs)

This section will describe each of the available commands for using macros along with an example and explanation of where the command is likely to be used.

The MIDI QUEST program is designed so that there are no temporary data buffers used to consume memory. Any data received via MIDI is loaded DIRECTLY into the computer memory that will be used for editing. There is no intermediate memory used. Because of this design, you should be aware of getting from a command that requests SysX from an instrument (T n or TR n) to the command which actually receives the data (R n) quickly.

For both transmission and reception of MIDI data, you must work in terms of COMPLETE, properly formatted MIDI messages. You can not send a byte at a time.

The MIDI QUEST macro language has 4 variables: W, X, Y, and Z. These can be assigned values or used as variables in place of any parameter which would take a numeric value.

When a macro is called it is automatically given the memory required for storing data (as defined by FileSize in the Driver).

In the case of the Receive Macro:

1. A buffer containing 'FileSize' bytes is pre-allocated.
2. R n (receive n bytes) commands automatically load data into this buffer
3. After reception the data pointer is advanced so that additional data can be loaded

In the case of the Transmit Macro:

1. A pointer to the data is provided
2. T n (transmit n bytes) sends the desired data
3. commands automatically advance the pointer after n bytes have been transmitted

Notes

All character entries can be made in either upper or lower case. When a driver is loaded from disk, any macros are automatically uncompiled using upper case letters. (eg. TR 15 R 62).

Defaults

These are the values of various parameters upon macro entry

Data Pointer - points to the buffer used to load or send data

Data Size - the number of bytes in the buffer

Wait Mode - Wait for F0H to start receiving SysX information

Variable W = 0

Variable X = New value of an edited parameter(Templates only)

Variable Y = Communication Channel (Comm Ch)

Variable Z = selected Patch (Pch #)

Macro Commands

Transmit Bytes

T n

where

n = number of bytes to send

n = 0 will send all bytes Pointed at by the Data Pointer (for files of unknown length)

eg. T 163 (to send a Yamaha DX Voice)

T is used to immediately transmit n bytes via MIDI OUT starting at the Data Pointer.

WARNINGS:

Should only be used in the Receive macro after the (S)et command has been used to set up data to send.

NOTES:

All 3 MIDI transmit functions will automatically format in the data's selected MIDI channel if it is a standard MIDI message or will format in the Communication channel if the data is a SysX message and the Communication channel offset is > 0.

Transmit Bytes on next Receive

TR n

where

n = number of bytes to send

n = 0 will send all bytes in the current buffer (for files of unknown length)

eg. TR 5

TR is used to setup n bytes, starting at the Data Pointer, so that they are transmitted immediately before the macro waits to receive SysX bytes.

WARNINGS:

Should only be used in the Receive macro after the (S)et command has been used to set up data to send.

Transmit Bytes from Buffer

B n

where

where

n = the number of times to loop the macro
[] = macro statements to be looped

L allows a portion of the macros to be looped any number of times. This is useful for repetitively receiving or transmitting SysX information or to set multiple checksum values.

WARNINGS:

There can only be one loop at a time

Wait for SysX Start - Toggle

N

The N option toggles whether the software's MIDI SysX receive section is supposed to wait for a SysX Start Command (F0H) or not. By default it does, and the situations would be very rare when this command will be used. (An exception is the reception of Casio dumps where data must be received starting with a byte that is NOT F0H).

Delay

D n

where

n = number of microseconds

The Delay command stops the execution of the macro for n microseconds. It can be used in any situation where the execution needs to be slowed.

eg. An instrument can not handle the transmission of two large SysX messages one immediately after the other. Try transmitting one message, delay 5,000 microseconds (D 10000) and transmit the second message. The instrument is much more likely to receive the second transmission more accurately.

End of Macro

E or]

There are situations where it may be necessary to exit out of the macro before the end of the macro. This can be accomplished by entering an E or].

(This option is more commonly used in Template macros.)

Set Data Pointer to MIDI String

S n {an n byte complete MIDI message}

where

n = number of bytes in the MIDI message

This command sets the Data Pointer to the MIDI message, and all subsequent operations which write into the data will write into this string until it has been transmitted using the T or TR commands.

This command can be used to create any MIDI message you want in order to accomplish whatever your purpose is.

eg. S 5 {F0 43 20 0 F7} - Yamaha DX7 Voice Dump Request
eg. S 3 {90 60 40} - turn on note 96D with velocity of 64D
eg. S 2 {C0 02} - set up Patch change message for Patch2

WARNINGS:

1. The MIDI string MUST be entered in HEXIDECIMAL format.
2. You must ensure that a complete MIDI msg is being provided! In other words, a SysX message must start with F0H and end with F7H and a regular MIDI Message (eg. Note ON) must start with a status byte.
3. Running Status is NOT supported.
4. The Data Pointer is literally set to a position in the compiled macro. You must ensure that any subsequent operations, which write into the MIDI string, stay within the confines of the message. VERY NASTY PROBLEMS CAN RESULT OTHERWISE!

Assign Buffer

A

Force the Data Pointer to point back to the Buffer data instead of MIDI data built into the macro. This is useful in a situation where the (S)et command has been used but all of the data has not been set and it is necessary to access the Data Buffer again for either reception or transmission of data.

Checksum

K m n start len ofs

where

m = Data Format (should be the same as Storage Format)
n = Check Sum type (should be the same as in Check Sum)
start = offset from current Data Pointer position to start CSum
len = the number of bytes in the checksum calculation
ofs = offset from current Data Pointer position to store CSum

This is the macro equivalent of the 'Check Sum' portion of the Driver Creator window. Any data which contain a check sum should have this calculation run before transmission.

eg. K 0 0 6 155 161 (format command for DX7 voice)

Format All Bytes

FA m n {m MIDI bytes to copy}

where

m = number of bytes to format into current MIDI string
m = offset from current Data Pointer position to place the bytes
{ m bytes } = the bytes which are to be copied into the string

This command will format a string of bytes into the current MIDI String. This option is particularly useful for the Audition macro where a completely new message is being created.

eg. FA 8 0 {F0 41 0 16 12 4 0 0}

This example would format the correct header code necessary to transmit a new tone to a Roland D-10.

Write a Byte (Unformatted)

F(W..Z) n

where

(W..Z) = any 1 of the four variables: W, X, Y, Z
n = offset from the current Data Pointer to store the data

This command takes the contents of the variable (W..Z) and stores it in the nth byte from the current Data Pointer

eg. FW 4

This takes the contents of W and stores it in the fourth byte from the Data Pointer.

NOTE: This command does NOT use the Storage Format (See FR).

Read a Byte (Unformatted)

FG (W..Z) n

where

(W..Z) = any 1 of the four variables: W, X, Y, Z
n = offset from the current Data Pointer to store the data

This command takes the contents of the variable (W..Z) and stores it in the nth byte from the current Data Pointer

eg. FG W 4

This takes the contents of the fourth byte from the Data Pointer and stores it in the variable W.

NOTE: This command does NOT use the Storage Format (See FS).

Math Operations

M(W..Z) op (W..Z,n)

where

(W..Z) = any 1 of the four variables: W, X, Y, Z -> destination

op is one of the following math operations:

+	(add)	-	(subtract)
*	(multiply)	/	(divide)
	(bitwise OR)	&	(bitwise AND)
>	(shift right)	<	(shift left)
=	(equal)	%	(mod)

(W..Z) = any 1 of the four variables: W, X, Y, Z -> source

n = any numeric constant

The macro language has an extensive selection of macro operations which can be performed on variables. For those of you familiar with the 'C' language, the operations are identical. We will use a number of examples to illustrate how the math operations work.

A table will be kept showing the contents of each variable after the completion of each operation.

Operation	Explanation	Var	W	X	Y	Z
START			1	6	3	0
MW = 3	W takes on the value 3	3	6	3	0	0
MX = Z	X takes on the value of Z		3	0	3	0
MX + 1	add 1 to X		3	1	3	0
MW + Y	add contents of Y to W	6	1	3	0	
MW / Y	divide W by Y		2	1	3	0
MY < 4	bit shift Y left 4	2	1	48	0	
MY 21	00010101 00110000 -> 00110101		2	1	53	0
MX < 7	bit shift X left 7		2	128	53	0
MX > 2	bit shift X right 2		2	32	53	0
MY & 21	00110101 & 01010101 -> 00010100		2	32	20	0
MX % Y	X mod by contents of Y	2	8	20	0	

WARNING:

You must NOT attempt to divide a variable by 0.

Delay MIDI transmission

D M n

where

n = every nth byte to delay on

This command slows the MIDI transmission rate to accommodate synthesizers which can not receive transmissions at full MIDI rate. This command will slow SysX transmission so that for every nth byte there is a 10 millisecond delay before transmission recommences.

eg D M 20

Delay 10 milliseconds every 20th byte transmitted.

WARNING:

If this type of transmission is required for a particular instrument to function properly, the sequencer should NOT be run while communication of this type is taking place!

Receive Data Unstored

RB n

where

n = the number of bytes to receive (between 1 and 100)

This command is used when data must be received from an instrument and it is not to be kept. The data is read and stored in a temporary buffer and immediately dumped.

eg. RB 52

Read 52 bytes into the temporary buffer and dump.

Global Commands

GA n	- allocate n bytes of memory
GN	- set Data Pointer to the new memory
GO	- set Data Pointer to the old memory
GR	- reset Data Pointer to start of new memory
GC m p q	- copy block of bytes from old mem to new
GF n	- free the allocated memory

where

n = number of bytes

m = number of bytes offset from the start of the old memory

p = number of bytes offset from the Data Pointer

q = number of bytes to copy from the old memory into the new memory

All of the operations in this section are usually used in the Audition macro and in the File Conversion macros which can be used to convert other manufacturers files to Sound Quest format files.

GA is used to allocate the memory necessary to create a complete Patch from a Bank, usually n will equal the complete byte size of the Patch.

GN is used after the memory has been allocated and sets this memory for use by the other Data Pointer related operations.

GO is used to set the Data Pointer back to the original data so that it can be operated on if necessary.

GR is used to reset the Data Pointer to the beginning of the allocated memory so that it can be reused if

desired. This function will probably be used if there is a loop in the Audition macro.

GC is usually used to copy the sound data from the Bank into the newly allocated data which will be used for the audition.

GF is used to free the memory - this should be called before exiting the macro.

Here is an example which incorporates most of the functions in this section and is drawn directly from the D-10 Tone Audition macro. It is the complete code to prepare and transmit the data.

GA 256	- allocate memory
GN	- make memory active
FA 8 0 {F0 41 0 16 12 3 0 0 }	- format in header
GC 0 8 246	- copy in body
K 0 0 5 249 254	- generate checksum
MX = 247	- set F7H end
FX 255	- copy F7H into end
T 256	- transmit complete tone
GF 256	- free the data

WARNING:

DANGER! When using GC ensure that the copy does not exceed the available buffer space.

Read Byte from Data (Formatted)

FR (W..Z) ofs

where

(W..Z) = one of four variables: W, X, Y, Z
ofs = offset from the data pointer to get the data

This command reads a byte of data out of the buffer currently pointed to by the Data Pointer. The command uses the Storage Format parameter when reading the data so that if the data is in a non-standard format such as nibblized or in M1 format, this is accounted for and the complete byte is read and provided. This command is more commonly used in Template macros but could be used for other purposes.

eg. FR W 4

This reads the fourth byte offset from the Data Pointer into the variable W.

The complement of this command is FS W 4.

WARNING:

Ensure that the byte read is within the current buffer.

Store Byte in Data (Formatted)

FS (W..Z) ofs

where

(W..Z) = one of four variables: W, X, Y, Z

ofs = offset from the data pointer to store the data

This command stores data in a variable into a location offset from the buffer currently pointed to by the Data Pointer. The command uses the Storage Format parameter when storing the data so that if the data is in a non-standard format such as nibblized or in M1 format, this is accounted for and the complete byte is stored appropriately. This command is more commonly used in Template macros but could be used for other purposes.

eg. FS X 10

This command stores the value in W in the tenth byte offset from the Data Pointer.

The complement of this command is FR X 10.

WARNING:

DANGER! Ensure the offset does not exceed the size of the data

Disk Read

DR n ofs

where

n = the number of bytes to read from the current disk file

ofs = where to load the data relative to the Data Pointer

This command reads a block of n data bytes from an open disk file and stores it in memory with a Data Pointer offset of 'ofs'. This command is necessary for converting files from other manufacturers.

eg. DR 155 6

Read 155 bytes and store it starting 6 bytes from the Data Pointer.

WARNING:

1) This command should ONLY be used in a File Conversion macro created in the File Conversion Window!

2) DANGER! Ensure the data read is not larger than the available buffer size!

Disk Seek

DS n

where

n = the number of bytes to advance

This command moves the position of the disk file pointer forward n bytes so that data can be read from the middle or end of a file. Not all companies use the same file format so it is necessary to be able to

move anywhere in the file to collect the data necessary to create a Sound Quest file of the same type.

eg. DS 166

Advance the disk file pointer 166 bytes.

WARNING:

1) This command should ONLY be used in a File Conversion macro created in the File Conversion Window!

Conditional Execution

IF (W..Z)	- execute code if variable is NOT equal 0
IFN (W..Z)	- execute code if variable is equal 0
IFL (W..Z)	- execute code if variable is less than 0
IFG (W..Z)	- execute code if variable is greater than 0
IFE	- end any conditional execution

where

(W..Z) is one of four variables: W, X, Y, Z

These commands allow execution of macro code only if certain requirements are met. These commands are far more commonly used when writing Template macros. They allow the limitation of certain actions based on the state of variables. If a test fails, all macro code is ignored until the first IFE command is executed and operation returns to normal.

NOTE: If IFE is used at the very end of a macro, an error is generated. Enter E or] to end the macro correctly.

PC MPU mode commands

ND	- enter MPU dumb mode
NI	- enter MPU intelligent mode

There are a number of limited circumstances where it is absolutely necessary to enter the MPU dumb mode in order to receive or transmit the required data. Two of these are transmitting and receiving Casio SysX information and Receiving large Ensoniq sequence dumps.

If your macro causes the MPU to enter dumb mode, ensure that it is reset to intelligent mode before exiting the macro.

NOTE: The macro compilers on other computer systems are capable of ignoring these commands.

WARNING:

PC MPU users MUST NOT be running the sequencer when executing these macros.

Examples

Here are a number of macro examples drawn directly from existing Sound Quest Drivers, with explanations:

M1 Combination Bank Receive Macro

S 7 {F0 42 30 19 1D 1 F7} - set the dump request
TR 7 - transmit bytes on next receive
R 14179 - receive M1 Combination Bank

M1 Combination Bank Audition Macro

S 8 {F0 42 30 19 4E 1 10 F7} - set M1 in Combination edit
T 8 - transmit the message
D10000 - delay 10 ms. to allow response
GA 148 - allocate memory for audition
GN - set new memory for work
FA 5 0 {F0 42 30 19 49} - setup Single Combination header
GC 0 5 124 - copy the combination data
MX = 247 - set X = 247 (F7H) end of SysX
FX 147 - put F7H into last byte
T 148 - transmit the 148 bytes
GF 148 - free the allocated memory

FB01 Patch Bank Transmit Macro

NOTE: Each Patch has its own Check sum which must be recalculated.

MW = 76 - set offset to start of the first
checksum to be calculated.
MX = 204 - set offset to store first the
checksum.
L 48 [- loop for each of 48 Patches
K 0 0 W 128 X - calculate Patch's checksum
MW + 131 - advance W to next Patch
MX + 131 - advance X to next Patch
] - end of loop
T 6363 - transmit the Bank

Macro Errors (Technical Docs)

These errors are all generated while attempting to compile a macro for a driver, a template or a file conversion. The error message is provided along with the line that the error was found in and the character position.

As with the MIDI communication errors, the errors are presented in the order in which they are most likely to occur.

Destination Buffer Exceeded!

Possible Causes:

1. Some macro command was entered improperly and the compiler has missed it. (This is the much more likely of the two.)
2. The macro is so incredibly long that it has used up the available buffer space (currently 1000 bytes for each macro).

Solutions:

1. There is an error somewhere in the macro, unfortunately you will have to look a little to find it. To test this methodically, use a letter which is definitely NOT a command such as 'H'. Insert the H in the code. Compile the macro. A "Bad Macro Command" error will occur. Continue moving the 'H' through the macro until the "Destination Buffer Exceeded" error is generated instead of the error for a bad macro command.
2. Find ways to optimize the macro by using the loop and conditional execution commands.

Unknown Character

Possible Causes:

1. An invalid macro command was entered at the given location.
2. An extension to the IF command was invalid (ie NOT IFN, IFE, IFG, or IFL)
3. The compiler could not process the character in the position it was found.

Solutions:

1. Check the position and enter the correct macro command.
2. Enter the correct conditional execution command.
3. Check the code in the area where the error was generated very closely.

Invalid Value

Possible Causes:

1. A hex value was entered which exceeded 255
2. A 'Set' command was used which described a MIDI string with more bytes than were actually entered.

Solutions:

1. Check to see that the entry at the error point is less than FFH or 255D.
2. If a MIDI string of 8 is set (ie. S 8 { }) ensure that there are actually eight MIDI bytes entered in the string.

Invalid Math Operation

Possible Causes:

1. An invalid math operation was entered.

Solutions:

1. Check the error position and enter a valid math function.

Invalid Assignment Opcode

Possible Causes:

1. The second character of an opcode is invalid at the error position.

Solutions:

1. Check the error position and enter the valid op code.
(eg GA and GF are valid but GH is not)

End of Macro Buffer

Possible Causes:

1. An IFE command can not be the last command in a macro.
2. The compiler has reached the end of the Macro Buffer unexpectedly.

Solutions:

1. Enter an end of macro command after IFE (either ']' or 'E').
2. This is a very insidious problem in the macro. Use the method described in the "Destination Buffer Exceeded" error to find it.

Midi Errors (Technical Docs)

These errors are all generated while attempting to either transmit or receive System Exclusive information. Before attempting to deal with the particular error, you should verify this information:

1. Communications channels for data and instrument match.
2. SysX communication is ENABLED for the instrument.
(It is possible to shut off SysX on many instruments)
3. Check that the Patch Bay is properly mapped to the instrument.
4. All requirements & warnings in Fast Tips have been met.
5. Connect the instrument DIRECTLY to the computer.
6. ie. No merge boxes, mappers, etc. are connected.
7. Retry the communication.

If after checking all of these possibilities an error is still being generated, it will be necessary to check out the particular error. The "Possible Causes" listed below will not include those already covered!

In the listings below, the problem causes are listed in order of greatest to least likely. You should try to sort out the problem based on the order of this listing.

NOTE: If you are working on a new driver, it is much more likely that there is an error in the driver. Concentrate on correcting the driver parameters. While the listings below include errors resulting from an incorrect driver, it considers them to be improbable and so usually they are listed at the bottom. CHECK THESE OUT FIRST!

ERROR! Incorrect # of bytes received!

Possible Causes:

1. Error during MIDI communication.
2. Error in the instrument's ROM software.

Solutions:

1. Retry MIDI communication.
- 2A. Check Fast Tips to see if there are any known problems.
- 2B. Contact Roland to determine whether the latest software is being used. If you are not, have your ROMS replaced.

NOTE: This error can only be generated by ROLAND communication.

ERROR! MIDI Unit did not respond!

Possible Causes:

1. The driver dump request is incorrect.
2. The MIDI Channel Ofs (Communication Channel Ofs) is incorrect.
3. Instrument did not respond within timeout period.

NOTE: Cause 3 (though very unlikely) should only occur in systems which require a polled MIDI reception scheme. The only polled reception currently used is with an ATARI ST running under CLab's Director.

Solutions:

1. Verify and correct if necessary.
2. Verify and correct if necessary.
3. Load the macro into the Driver Creator Window and add a small delay in the Receive Macro between transmitting the dump request (T n) and receiving the data (R n).

ERROR! End of SysX not received (TimeOut)!

Possible Causes:

1. The wrong dump request is being used for the expected data.
2. Instrument did not complete transmission within the timeout period.

NOTE: Cause 2 (though very unlikely) should only occur in systems which require a polled MIDI reception scheme. The only polled reception currently used is with an ATARI ST running under CLab's Director.

Solutions:

1. Verify and correct if necessary.
2. Load the macro into the Driver Creator Window and make sure that the TR n macro command is used for transmitting any dump request messages.

ERROR! SysX missing bytes!

Possible Causes:

1. A byte was lost by the MIDI receive circuitry.
2. Driver has an incorrect "FileSize" entry ('Sgl' mode).
3. Macro Receive call has incorrect # of bytes ('Ext' mode).
4. Driver is expecting a fixed length MIDI message when the message is in fact a variable length.

Solutions:

1. Retry the MIDI operation.
2. Verify and correct the "File Size" entry if necessary.
3. Verify and correct R n or RA x n macro calls if necessary.
4. Take generic dump into SysX View Window and verify file size or make "Fixed Length" 'No' if this is the case.

NOTE: When "Fixed Length" is 'Yes', every SysX reception undergoes a number of tests to determine whether valid data was received.

ERROR! Unexpected end of Macro!

Possible Causes:

1. The Data Pointer has been set into the macro (use S cmd) and you are then writing beyond the size of the MIDI string.

Solutions:

1. Load and check your macro very carefully if your computer has not already crashed.

ERROR! Unexpected Macro Command

Possible Causes:

1. The macro was written on a newer version of MIDI QUEST which has an enhanced macro language.
2. The Data Pointer has been set into the macro (use S cmd) and you are then writing beyond the size of the MIDI string.

Solutions:

1. Contact Sound Quest, report the problem and we will attempt to determine whether this is the problem.
2. Load and check your macro very carefully if your computer has not already crashed.

ERROR! No EOF transmission found!

Possible Causes:

1. A byte was lost by the MIDI receive circuitry.
2. Driver has an incorrect "FileSize" entry ('Sgl' mode).
3. Macro Receive call has incorrect # of bytes ('Ext' mode).
4. Driver is expecting a fixed length MIDI message when the message is in fact a variable length.

Solutions:

1. Retry the MIDI operation.
2. Verify and correct the "File Size" entry if necessary.
3. Verify and current R n or RA x n macro calls if necessary.
4. Take generic dump into SysX View Window and verify file size or make "Fixed Length" 'No' if this is the case.

NOTE: When "Fixed Length" is 'Yes' then every SysX reception undergoes a number of tests to determine whether valid data was received.

ERROR! Incorrect Company ID!

Possible Causes:

1. Error during MIDI communication.
2. An instrument responded which should not have.
3. The incorrect "Company ID" was entered in Driver.

Solutions:

1. Retry the MIDI communication.
2. Ensure the correct instrument is connected to the computer.
3. Verify and correct the "Company ID" - IN HEX!

ERROR! Unexpected ACK recvd!
ERROR! Roland Communication Error!
ERROR! Unrecognized Roland Error!

Possible Causes:

1. Error during MIDI communication.
2. Error in the instrument's ROM software.

Solutions:

1. Retry MIDI communication.
2. Check Fast Tips to see if there are any known problems.
2. Contact Roland to determine whether the latest software is being used. If you are not using the latest software, have your ROMS replaced.

NOTE: This error can only be generated by ROLAND communication.

ERROR! SysX Data Rejected!

Possible Causes:

1. Error in MIDI communication.
2. Instrument is in the wrong mode for the desired communication.
3. Error in the instrument's ROM software.

Solutions:

1. Retry MIDI communication.
2. Check Fast Tips to see if there are any known problems.
2. If the driver uses handshaking - set communication properly.
2. If no handshaking - ensure instrument is not in that mode.
3. Contact Roland to determine whether the latest software is being used. If you are not using the latest software, have your ROMS replaced.

NOTE: This error can only be generated by ROLAND communication.

ERROR! Dump Too large for allocated memory!

Possible Causes:

1. A driver for a variable length SysX dump only made a partial memory allocation to save memory. The instrument sent more bytes than were allocated.

Solutions:

1. There will usually be a driver with a larger "FileSize" (which controls memory allocation). If not, increase the driver's "FileSize" using the Driver Creator Window.

ERROR! Memory Allocation Failed!

Possible Causes:

1. When a driver does not have a fixed length (ie "Fixed Length" is 'No'), then after SysX data is received a new buffer is allocated which is the same size as the number of bytes received. If the allocation of the new buffer fails, this error is generated.

Solutions:

1. Make more memory available to the computer for this buffer by closing as many windows as is necessary to correctly receive the data.

Morph (Patch Editor)

Purpose:

Morph creates a Bank of new Patches by creating Patches which incrementally transpose from one Patch to a second. This function will probably be executed more frequently from the [Bank Editor Window](#) or [Library Window](#) where there is a wide selection of source Patches to choose and you wish to morph from one voice entirely to another.

The morph capabilities in the Patch Editor Window are slightly different. Here, only the currently selected parameters are morphed. All other parameters are set to the parameter value in the first Patch for the first half of the new Patches and to the second value for the second half of the new Patches.

This function is ideal for creating a sequence of new Patches which have only very minor changes between them. For instance, you may have set up two Patches which are identical except for one envelope and you would like to create a sequence of Patches which gradually transpose from the first sound to the second. This is the function which will do it.

The function works by loading [buffer 1](#) and [buffer 2](#) with two working/source Patches, selecting the parameters that you wish to morph, and choosing the Morph function. The program will create a new Bank and load the new Patches into the Bank.

For more information on the [Morph](#) function, please see the Bank Editor.

Directions:

To create a new Patch Bank by sliding from one Patch to another and morph only selected parameters::

- (1) Make any changes necessary to prepare the current Patch to be the initial Patch for a Morph
- (2) Choose *SnglEdit/Organize/Buffer Save/1*
- (3) Make any changes necessary to the current Patch to be the end Patch for a Morph
- this might include transferring a different Patch into the current editor window
- (4) Choose *SnglEdit/Organize/Buffer Save/2*
- (5) Select the parameters you wish to morph
- (5) Choose *SnglEdit/Randomize/Morph*
- (6) A new Bank of Patches is created to audition or work with

See Also: [Morph](#) (Bank Editor)

See also: [Patch Editor](#)

Disk Errors (Technical Docs)

These errors are all generated while attempting to save or read data from disk.

As with the MIDI communication errors, the possible causes are presented in the order in which they are most likely to occur.

File NOT Found!

Possible Causes:

1. The path contains an invalid path.
2. The file name was entered incorrectly.
3. The file is not stored in the selected directory.

Solutions:

- 1A. Check that the path is valid.
- 1B. Click the mouse in the path string to activate it. Hit the <RETURN> key to display all of the files in the directory. If no files are displayed, the path is probably invalid and should be reentered.
2. Use the Up and Down Arrows or the slider to display the desired file. Double click on it to load.
3. Using #2 will determine whether the file is stored in the directory.

File READ Error!

Possible Causes:

1. The file has somehow been corrupted.
2. Hardware access error.

Solutions :

1. If you have a program that checks the integrity of a disk, run it on the disk which contains the problem file.
2. Save your current work. Shut the computer off and restart it. Try loading the file again.

NOTE: ALWAYS KEEP BACKUPS OF ALL FILES ON A SEPARATE DISK!

Not an IFF File!

Possible Causes:

1. The selected file does not use the storage format MIDI QUEST uses and was not created by MIDI QUEST.
2. The file has been corrupted.

Solutions:

- 1A. The wrong file was probably selected by mistake so select a different file.
- 1B. The file was created by a different software package and requires file conversion.

2. Load the data from a backup disk.

Not an IFF Data File!

Possible Causes:

1. The selected file contains some kind of information other than data.

Solutions:

1. Select a data file.

Not an IFF Driver File!

Possible Causes:

1. The selected file contains some kind of information other than a Sound Quest MIDI driver.

Solutions:

1. Select a Sound Quest MIDI driver file.

File NOT a Data Base!

Possible Causes:

1. The selected file contains some kind of information other than a data base.

Solutions:

1. Select a data base file.

Directory NOT Found!

Possible Causes:

1. The name path of a directory on the disk was entered manually and incorrectly.

Solutions:

1A. Correct the mistake (typo) in the path retry.

1B. Enter only the disk name and use the mouse to select directories to make the path.

Insufficient Memory!

Possible Causes:

1. The computer did not have sufficient memory to load the entire file.

Solutions:

- 1A. Close some already open windows to free sufficient memory to load the file.
- 1B. If this is a consistent problem and you are loading Banks, consider turning 'Keep Backup' Off in the Options menu.

File WRITE Error!

Possible Causes:

1. There is insufficient disk space to save the file.
2. There is a hard error on the disk.
3. There is a hardware problem.

Solutions:

1. Try saving to a different disk.
2. Try saving to a different disk.
3. Try saving to a different disk in a different drive.

File CREATION Error!

Possible Causes:

1. An unacceptable file name was used (ie unacceptable characters).
2. The file is being saved to a directory which does not exist.
3. The file being written to has been given READ ONLY status.

Solutions:

1. Give the file a valid operating system name.
2. Verify that the destination directory is valid.
3. If a file of the same name already exists, verify its READ/WRITE status.

Expand All (Driver List)

Purpose:

Expand All is used to automatically expand each of the instrument bars so that all of the drivers are displayed. Use this function if you would like to quickly view a list of all available drivers.

Directions:

(1) Choose *Drivers/Expand All*.

See also: [Driver List Window](#)

Gen4 (Bank)



Purpose:

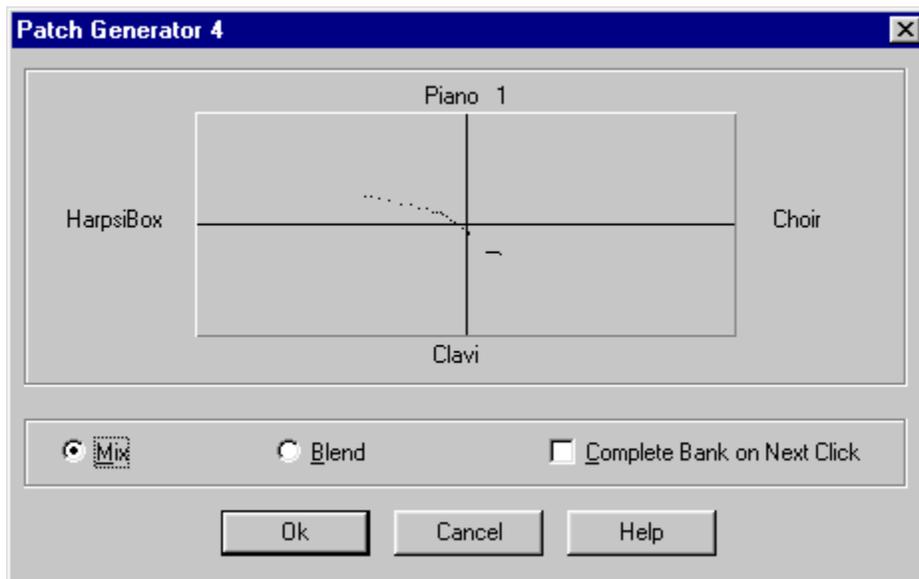
Gen4 opens the Patch Generator 4 Dialog. It is used to create a Bank of new Patches by mixing the components of four selected Patches. The percentage component of each Patch is determined by the location of each of your mouse clicks.

Directions:

To open the Gen4 Dialog:

- 1) Select four Patches from the Bank
- 2) Choose the 'Gen4' button or *Bank Edit/Gen 4...*
- 3) Use the Generator 4 dialog to create a Bank of new Patches

The Generator 4 Dialog



Purpose:

The Generator 4 dialog allows four Patches to be mixed together to create a Bank of new Patches. Two Patch names are displayed at the ends of both the X and Y axes. Each click of the mouse in the grid area will create a new Patch. The components of the new Patch are determined by the location of each of your mouse clicks. As the mouse is clicked closer to a particular Patch, more of that Patch is included in the newly created Patch. With each click, a black dot is placed in the grid to indicate that a Patch was created for that position.

An entire Bank of new Patches can be created in one of two ways. For instance, First, by the click of your mouse. If the Bank has 32 Patches, the mouse can be clicked 32 separate times to create 32 new Patches. Secondly, the mouse button can be clicked, held down and dragged. New Patches are then created as the mouse is dragged within the grid. In each case, it is possible to automatically finish creating the Bank at any time by checking **Complete Bank on next click** and clicking once in the grid.

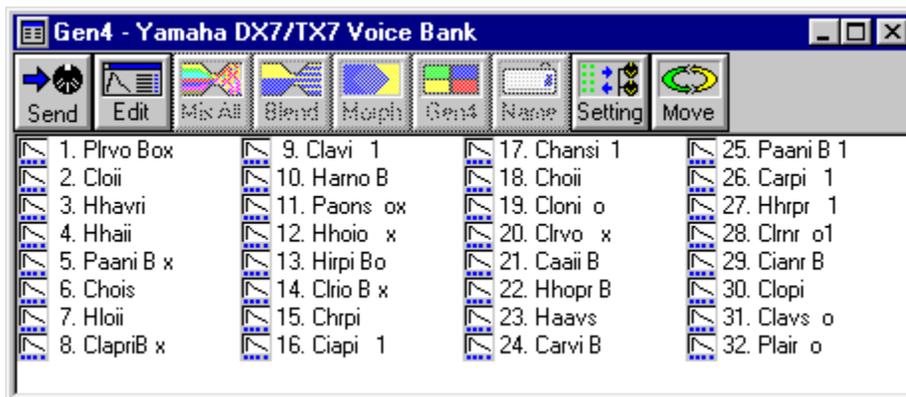
New Patches can be created in one of two ways. The method of creation can be changed at any time. If the **Mix** button is selected, the program randomly gets a parameter value directly from one of the four Patches based on the location of the mouse click in the grid. In this way the new Patch will only contain parameter values drawn from the four source Patches. If the **Blend** button is selected, the program takes the parameter values from each of the four Patches and combines them to create a new value based on the location of the mouse click. This new value may not be the same as any of the source values.

For example, assume the attack time of the amplitude envelope is 85, 65, 80, and 84 for each of the four Patches. If **Mix** is selected, the new Patch will have either 85, 65, 80, or 84 as it's amplitude. If **Blend** is selected and equal percentages of the four Patches are requested, the program will create the new value as follows: $(0.25 \times 85) + (0.25 \times 65) + (0.25 \times 80) + (0.25 \times 84) = 78$. The amplitude of the new Patch would be 78.

The following are some examples of where to click in the grid to create various combinations of the four Patches:

- to create a Patch which is 25% Piano 1, 25% Clavi, 25% Harpsi Box, and 25% Choir, click in the very center of the grid.
- to create a Patch which is 50% Piano 1, 25% HarpsiBox, and 25% Choir, click the mouse at the very top of the Y-axis.
- To create Patches which are mostly Piano1 and Choir, click the mouse in the upper. right quadrant of the grid.

The following Bank was created by Gen4ing the following Patches: Piano1, Choir, Clavi, and HarpsiBox.



See Also: [Library Window Gen 4](#)

See Also: [Bank Editor Window](#)

Update Patch (Library)



Purpose:

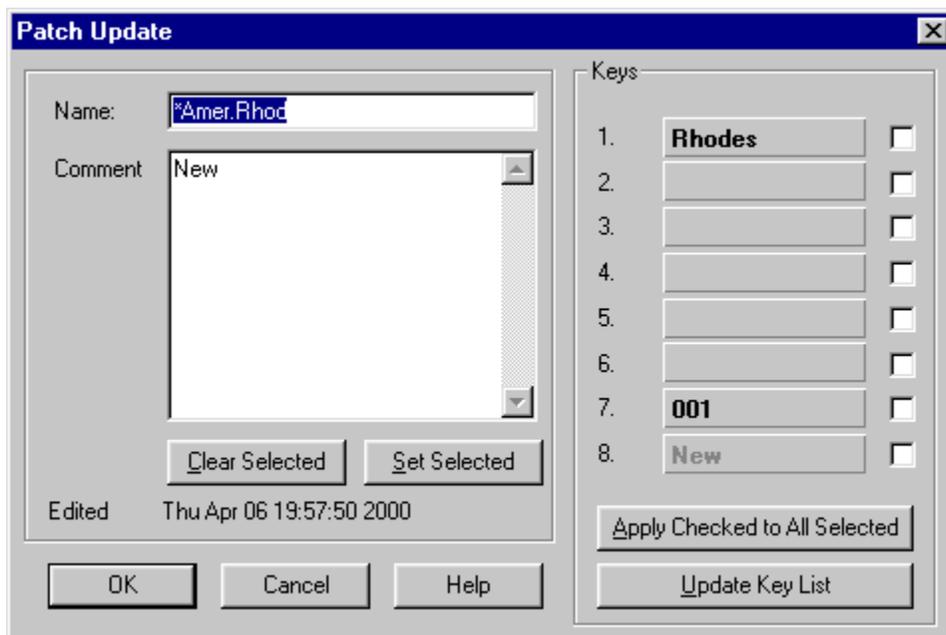
Update Patch opens the Patch Settings dialog. In this dialog you may edit the name, add a comment or set the keys for a Patch.

Directions:

To change the settings of a particular Patch in the Library:

- (1) Select one or more Patches to update or
- (2) Double click on a patch to automatically open this dialog
- (3) Choose the 'Patch' button or *Library/Open/Update Patch...*
- (4) Make any desired changes to the Patch settings in the dialog
- (5) Press the OK button
- (6) If more than one patch was selected, the dialog is reopened with the next selected patch
- (7) This continues until the settings for each patch have been displayed

The Patch Update Dialog



By using the Patch Settings Dialog you may edit the basic settings of a Patch:

Name: the name of the Patch

Comment: the comment assigned to the Patch

Clear Selected: Press this button and the comment field in ALL selected patches is immediately cleared. Please note, this option can't be cancelled.

Set Selected: Press this button to assign the current comment field to the comment field of ALL selected patches.

Keys: 8 keywords may be assigned to each Patch to describe the Patch's properties or qualities. These keywords are used by the [Choose Dialog](#) when looking for particular types of Patches. If more than one patch is selected when this dialog is opened, the program shows keywords which are identical in all patches in grey while unique keywords are shown in black. This should assist you in which modifying keys.

To add a keyword, click the left mouse button in the raised area. A pop-up menu will appear from which any of the 8 key words can be selected. If you take the time to add keywords to your patches, you will be repaid in full over time as you will be able to find different kinds of Patches quickly and easily. To add new entries to the current list of available keywords, click on [Update Key List](#).

In normal operating mode, the key selected key is added to the list of keys. At the bottom of the popup key list is an option called "Key and Category". It can be checked by selecting it. When this option is checked, selecting a keyword actually adds two entries. The selected keyword is added but in addition, the category from which the key word came is also added.

Apply Checked to All Selected: To apply keys assigned to the current patch to all of the selected patches, check the check box beside each of the keys you would like to assign and press the "Apply Checked to All Selected" button. This applies the current keys to each patch and over-writes the current key assignment for each of the patches. The use of checkboxes allows you to be selective about which keys are assigned to the group

Note: Once you have pressed **Apply Checked to All Selected** you can press the Cancel button to exit the dialog. The changes you have made will still be retained and you will not be forced to step through each selected patch in the Library.

See Also: [Settings Dialog](#)

See Also: [Library Window](#)

Resolve Links (DBase)

Purpose:

Resolve Links converts all linked disk files into data files that are stored in the DBase.

For instance, if you wish to take a DBase which references linked disk files to a studio, you will have to take the DBase and all of the linked disk files unless you use Resolve Links before you leave for the studio. Resolve Links places any linked disk files directly into the DBase so you can just take your DBase to the studio. Everything you need will be there.

Directions:

(1) Choose *DBase/Disk/Resolve Links*

See Also: [Link a Data File](#)

See Also: [Save a Data File and Link](#)

See Also: [DBase Window](#)

Link A Data File (DBase)

Purpose:

Link a data file is very similar to [Load A Data File](#). Both are used to bring a particular data file into the DBase. Link a Data File, however, does not actually store the data file in the DBase, it just stores a reference to the data file. This means that any changes made to the source data files on disk are automatically reflected in the linked data file in the DBase. **It also means that if the data file on disk is erased, it will not be available to the DBase and will be removed!**

Linking data files into DBases, Groups, and Libraries is an extremely effective method of data management for several reasons. First, Banks and other large SysX dumps need only be saved to disk once. Each DBase, Group, or Library can then link to the data file so there is only one copy of the data file stored instead of one copy per DBase, Group and Library that uses it. Second, since there is only one data file, all changes to that data file are reflected in each of the linked DBases and Libraries. This will save you substantial organizing time. Imagine having five different DBases and Libraries that are all supposed to use the same data file. If you were to change one of the data files, it would then be necessary to find all of the other DBases and Libraries that use the same data file and update them. With Link a Data File, it all happens automatically.

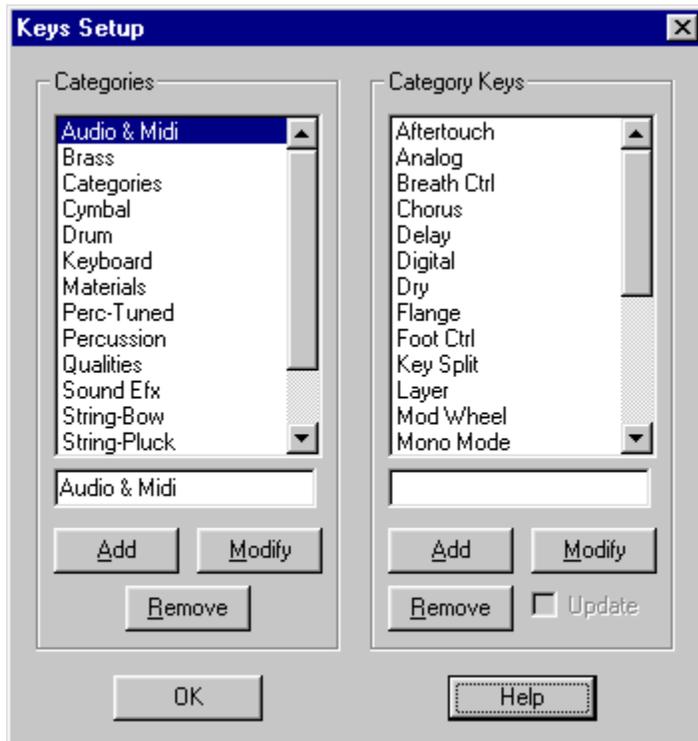
Directions:

To link one or more data files:

- (1) Choose *DBase/Link a Data File*
- (2) A File Selector is displayed
- (3) Choose the data files to be linked to the DBase
- (4) Hold down the CTRL key to select multiple files then press the OK button

See Also: [DBase Window](#)

Keywords (Common Elements)



About the Keywords Dialog

The Keywords Dialog can be opened either from the Settings Dialog which can be opened from the Driver List, Bank Editor, Patch Editor, Group, DBase, or Library. It can also be opened from the Patch Dialog in the Library Window.

This dialog is used to add, remove, and modify the list of keywords available to the program for describing Patches in a data file. Please notice that there is no Cancel button in this dialog. All changes to the list of keywords and categories are made immediately.

Categories

The left column contains a list of keyword categories currently available to the software. This provides a means of organizing your keywords into logical groups so they can be found more easily. Keyword Categories can not be placed in the Settings and Patch Dialogs themselves, they are just a means of organizing the available keywords.

To add a Keyword Category, type in the name of the new category in the text area above the **Add** button then press Add. To remove a Keyword Category, choose the category in the list and press the **Remove** button. Remember that all of the keywords in the category are also removed.

To change the name of a category, select that category then change the name of the category to your liking in the text entry area. Press the **Modify** button.

Category Keywords

The keywords are the actual entries which go in the Settings and Patch Dialogs and provide for efficient searching of different types of Patches.

To add a Keyword, type in the name of the new keyword in the Keywords text area above the **Add** button then press Add. To remove a Keyword, choose the in the list and press the **Remove** button.

To modify a keyword, select that keyword then change it to the preferred text. Press the **Modify** button.

When this dialog is opened from *Options/Update Key List...*, the **Update** option is enabled. If this option is checked then when a Key is modified (eg. you select a key, enter a new key word and press the Modify button), Midi Quest will automatically search through all of the libraries in the subdirectories of MQDATA and any patch that has the old keyword will automatically have the new key word substituted in its place. Remember that if you have many or large libraries, this operation could take some time to perform.

Important: If you enhance your list of keys and/or keywords, you will want to backup the MQKEYS.INI file before reinstalling or updating your program. The MQKEYS.INI file is found in your program directory. This file holds all of the keys entries and is automatically overwritten each time the program is installed. To maintain your custom list, make a copy of MQKEYS.INI, install the new version of the software, overwrite the new MQKEYS.INI file with your backed-up copy.

See Also: [Update Settings Dialog](#)

Save Data File and Link (DBase)

Purpose:

Save Data File and Link saves the selected DBase data file to disk. It then modifies the DBase so that the data file becomes a linked disk file. The data file itself is no longer stored in the DBase.

This is similar to [Save Data File](#). Both functions save the selected data file to disk, however, with Save Data File the data file remains stored in the DBase.

For a complete description of the advantages of file links, please see [Link a File](#)

Directions:

To save a data file and link:

- (1) Select the data file to save to disk and make a linked file
- (2) A file selector is displayed
- (3) Choose an appropriate directory for the file (if necessary) and give the file a name
- (4) Click on the save button.

See Also: [DBase Window](#)

Patch Names Only (Library)

Purpose:

16 bit only, for 32 bit [click here](#)

Patch Names Only toggles the Library between two display modes. In the default mode there is one Patch per row displaying the Patch's name, keys, and comments. In the Patch Names Only mode, only the Patch names are displayed with multiple names per row.

Directions:

(1) Choose *Library/View/Patch Names Only*

Examples:

A Library in the default display mode:



A library in the "Patch Names Only" display mode:



See Also: [Library Window](#)

Link A Patch/Bank (Library)

Purpose:

Link A Patch/Bank is very similar to [Load A Patch/Bank](#). Both are used to bring an individual Patch or Bank of Patches into the Library. Link a Patch/Bank, however, does not actually store the data file in the Library, it just stores a reference to the file. This means that any changes made to the source files on disk are automatically reflected in the linked data in the Library. **It also means that if the data file on disk is erased, it will not be available to the Library and will be removed!**

Please jump to [DBase Link a File](#) for more information on the advantages of file linking.

Note: only compatible Patches and Banks may be linked to the Library.

Directions:

To link multiple Patches and Patch Banks:

- (1) Choose *Library/Disk/Link a Data File*
- (2) A file selector is displayed
- (3) Choose the data file(s) to be linked to the Library
- (4) Press the OK button to load and link the files

See Also: [Library Window](#)

Save Patch And Link (Library)

Purpose:

Save Patch and Link saves the selected Library Patch to disk. It then modifies the Library so that the Patch becomes a linked disk file. The data file itself is no longer stored in the Library.

This is similar to [Save Patch](#). Both functions save the selected Patch to disk, however, with Save Patch the Patch remains stored in the Library.

For a complete description of the advantages of file links, please see [DBase Link a File](#)

Directions:

To save a Patch and link:

- (1) Select the Patch to save to disk and make a linked file
- (2) A file selector is displayed
- (3) Choose an appropriate directory for the file (if necessary) and give the file a name
- (4) Click on the save button.

See Also: [Library Window](#)

Resolve Links (Library)

Purpose:

Resolve Links converts all linked disk files into Patches that are stored in the Library.

For instance, if you wish to take a Library which references linked disk files to a studio, you will have to take the Library and all of the linked disk files unless you use Resolve Links before you leave for the studio. Resolve Links places any linked disk files directly into the Library so you can just take your Library to the studio. Everything you need will be there.

Directions:

(1) Choose *Library/Disk/Resolve Links*

See Also: [Link a Patch/Bank](#)

See Also: [Save a Patch and Link](#)

See Also: [Library Window](#)

Gen4 (Library)



Purpose:

Gen4 creates a new Bank of Patches based on four selected Patches.

For complete information on this function please click on [Bank Editor Gen4](#).

Directions:

- (1) Select four Patches in the Library.
- (2) Choose the 'Gen4' button or *Library/Randomize/Gen 4...*

A new Bank of Patches is created.

See Also: [Bank Editor Gen4](#)

See Also: [Library Window](#)

Morph (Library)



Purpose:

Morph creates a bank of new Patches which incrementally transpose from the first to the second selected Patch. Choose any two Patches in a Library and morph. The new bank will show the resulting morphed sounds and names.

Please click on [Morph](#) for additional information.

Note: Morph is only available in Midi Quest and Solo Quest.

Directions:

- (1) If the Library is not in split mode, click on the Edit button so that the Patch editing is enabled
- (2) Select two Patches in the Library
- (3) Choose the 'Morph' button or *Library/Randomize/Morph*

See Also: [Bank Editor Morph](#)

See Also: [Library Window](#)

Move To (Library)



Purpose:

Move To moves one or more Patches into any of the following destination windows: Bank Editor, Patch Editor, Group, DBase, or Library window.

Directions:

To move one or more Patches:

- (1) Choose the Patches to move
- (2) Choose *Library/Move To...*
- (3) The selected Patches are moved to the destination window

See Also: [Move To Dialog](#)

See Also: [Library Window](#)

MRU - Most Recently Used Files (File Menu)

Purpose:

The "Most Recently Used Files" lists the last 6 most recently used files used in Midi Quest. It provides a quick way to access "works in progress".

Directions:

To reload a most recently used file:

(1) Select the file name in the Files menu

See Also: [Files Menu](#)

Open SMF (Driver List)

Purpose:

Open SMF (Standard MIDI File) attempts to import SysX data currently stored in SMF format disk file directly into a Patch or Bank Editor. To accomplish this, you must select the driver in the program which represents the type of data stored in the SMF file. For example the Kawai K1 Patch Bank driver. Once selected, choose *Drivers/Open/SMF* to select the file to import and convert.

Note 1: This function will only work with SysX data that is stored as a single dump from the instrument. If you are trying to load a bank from an SMF file where each patch in the bank is a separate SysX dump inside the file, this function will not work.

Note 2: This direct SMF importation will NOT work with most Roland MidiX data dumps because of the inconsistent way that Roland instruments dump SysX data.

Directions:

To read an SMF file and open a Patch or Bank editor for it:

- (1) Select the driver in the program which corresponds to the SMF SysX data on disk
- (2) Choose *Drivers/Open/SMF* and use the file selector to choose the SMF file to convert
- (3) An editor window will open containing the SMF data in a patch or bank editor

See Also: [File Conversion Window](#)

See Also: [Driver List Window](#)

Copy Selected Parameters to (Patch Editor)

Purpose:

Copy Selected Parameters To copies the values of one or more selected parameters from the current Patch to one or more Patches in a Bank Editor or Library with the help of the Move To dialog. In essence, this is a Global Editing function where particular changes can be made to a Patch and these changes can then be assigned to a series of other Patches.

This function is ideal for applying new effects settings or changes in velocity sensitivity to a number of different Patches without having to edit each Patch individually.

Directions:

- (1) Select the parameters in the Patch Editor you wish to copy to other Patches
- (2) Select the Patches in a Bank Editor or Library that are to receive the parameters' values
- (3) In the Patch Editor, choose *SnglEdit/Copy Selected Parameters to...*
- (4) Use the Move To dialog to select the destination Patch Bank or Library

An error is display if the selected Bank Editor or Library does not itself have any selected Patches.

See also: Selecting Parameters

See Also: Patch Editor

Print Setup (Files Menu)

Purpose:

Print Setup selects the printer to use for all subsequent printing.

Directions:

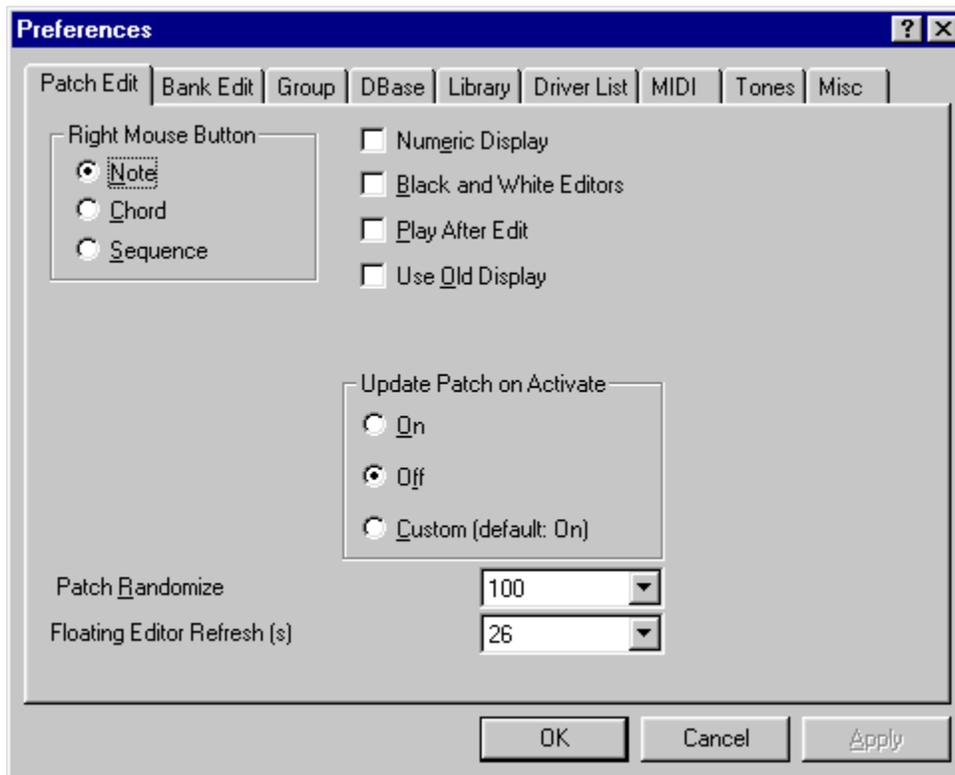
To choose the printer to output to:

- (1) Choose *Files/Printer Setup...*
- (2) Select either the default printer, or
- (3) Choose an available printer from the list

See Also: [Files Menu](#)

Patch Edit Preferences (Options Menu)

Note: because all of the following functions apply to editors, none of these settings are used by Midi Quest Jr.



Right Mouse Button (RMB)

RMB controls the type of MIDI data generated by Right Mouse Button clicks in a Patch Editor.

'Note' generates a single note. Its pitch is determined by the mouse's horizontal screen position. Its velocity is determined by the mouse's vertical screen position.

If 'Chord' or 'Sequence' are selected, the chord or sequence set in the Tones Window is played.

Default: Note

Font Selection

Patch Editor Font selects which font is used with any subsequently opened editor.

There are four different font options to choose from:

System: (default): uses the system font to display the editor

Small: uses an 8-point Terminal fixed width font

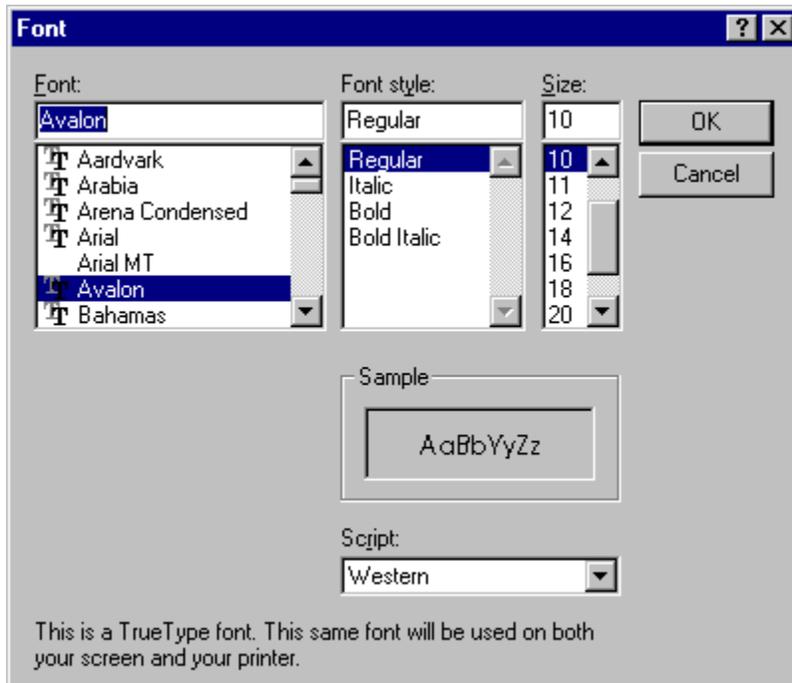
Large: uses a 12-point Terminal fixed width font

Custom: uses a font that you define using the font selector. Access this using the Select Font button in

the Patch Edit tab.

Note 1: Any currently open editor windows will continue to use the same font.

Select Font



Custom Font selector presents a dialog which is used to select the font to use with all subsequently opened editors. Select the font type, style, and size from your available fonts then press the OK button. The next editor you open will be displayed using this new font.

Directions:

- (1) Press the *Select Font* button
- (2) Use the selector to choose the font you would like to use in the editors
- (3) Press the OK button to accept the selection
- (4) Select the *Custom Font* option
- (5) Any new Patch editors will be displayed with the selected custom font

Numeric Display

Numeric Display controls whether Patch Editors with graphic sliders and levels also display numeric equivalents of these parameters. If set 'Off', only the graphic slider/level is displayed. If set 'On', subsequently loaded editors will display a numeric equivalent for each slider/level.

Note: these numeric equivalents are 'dumb' and in these cases do not show the number precisely the same as on the instrument. What is displayed instead is the actual value of the parameter as it is stored in the SysX data file.

Default: Off

Black and White Editors

The Black and White Editors option controls whether editors created in Midi Quest and Solo Quest are displayed in color or gray scale mode. If checked, subsequently open editors are displayed using gray scale only. If left unchecked, subsequently open editors are created with the color defined in their templates. This option has been provided for those who would prefer not to have color in their editors.

Default: Off

Play After Edit

Play After Edit, when checked, causes a test note, chord or sequence (as determined by the RMB setting) to be played after each change made in a Patch Editor. This function is useful if you wish to hear the change in should after each minor change is made and you are not playing a standard MIDI file.

Default: Off

Update Patch On Activate

The Update Patch on Activate function controls whether the patch is automatically sent to the instrument when its window is activated or not.

Off: the patch is not automatically sent to the instrument when the window is activated. To update the instrument with the entire patch, the patch must be sent manually. This is useful if the patch takes a long time to download and you are only editing one patch at a time.

On: easily work on different patches from the same instrument. As you move from editor to editor, the program automatically sends the patch to the instrument so you can hear the effects of your editing. This keeps the instrument "in sync" with the editor

Custom: The default is "On" but for each individual data type, this option can be turned off. This provides the option of being selective about which patches are automatically sent and which are not.

Default: On

Use Old Display

When checked all subsequently opened patch editors use the v6.0 display style instead of the v7.0+ style. The older display style has one benefit over the new style, it requires about 30% less vertical space to display the same information. The old display style has been left in for those who prefer it.

Patch Randomize

This parameter sets the level of randomization use by the Patch Editor's Constrained Randomize function.

Default: 100%

Floating Editor Refresh

This parameter sets how frequently floating editors are updated so that they stay in sync with each other. The frequency is from 1 to 119 seconds.

Default: 30s

See Also: [Patch Editor Window](#)

See also: [Options Menu](#)

Save Desktop As (Files Menu)

Purpose:

Save Desktop As saves a list of each open program windows. These windows can be reopened together in a later session by loading the desktop file.

This feature allows you to quickly load particular combinations of files which are frequently used together. For example, if you have a particular set of files you would like opened each time the program is run follow these steps. First open each of the windows you will need, choose Files/Save Desktop As..., and give the file a name. Once the desktop file is saved, select the program icon in Program Manager, choose *File/Properties*, and append the desktop's path and file name to the command line. The next time the program is run, it will automatically load all of the files in the desktop.

The desktop file stores a list of only those windows that have been previously saved to disk. Any windows which are open in the program and have not been saved will not be recorded in the desktop file.

Directions:

To create a desktop file:

- (1) Load each of the files you wish to have in your desktop
- (2) Save any files currently on the screen which have not been previously saved
- (3) Choose *Files/Save Desktop As...* and name the desktop file

Note: you can get a list of all desktops saved in the MQDATA directory just by right clicking in Midi Quest's frame/background window

See Also: [Files Menu](#)

Group Member (Driver List)

Purpose:

Group Member is a toggle parameter. If set, the driver becomes part of the instrument's Group. When the Instrument Bar is selected and the 'Edit' or 'NEdit' buttons are pressed, the program loads or creates a new group using the instrument's group drivers.

For more information, please see [Groups](#).

Directions:

To add a driver to an instrument group:

- (1) Select the driver in the Driver List that is not currently part of the instrument's group
- (2) Choose *Drivers/Update Settings/Group Member*
- (3) 'Grp' will appear in the driver's status line

To remove a driver from an instrument group

- (1) Select the driver in the Driver list that is currently part of the instrument's group
- (2) Choose *Drivers/Update Settings/Group Member*
- (3) 'Grp' will disappear from the driver's status line

See Also: [Group Window](#)

See Also: [Driver List Window](#)

Open MidiX (Drivers List)

Purpose:

Open MidiX attempts to import SysX data currently stored in MidiX format disk file directly into a Patch or Bank Editor. To accomplish this, you must select the driver in the program which represents the type of data stored in the MidiX file. For example the Kawai K1 Patch Bank driver. Once selected, choose *Drivers/Open/MidiX* to select the file to import and convert.

Note1: this function will not work for all MidiX files. For the importation to work, the MidiX file must contain only the type of data you are trying to load. For example, if you are trying to import a Kawai K1 Patch Bank but the MidiX file contains data other than a Patch Bank or both a Patch Bank and a Multi Bank, the data is not imported correctly. In these situations, you will need to use the File Conversion Window to write a macro to perform the conversion.

Note2: This direct MidiX importation will NOT work with most Roland MidiX data dumps because of the inconsistent way that Roland instruments dump SysX data.

Directions:

To read a MidiX format file and open a Patch or Bank editor for it:

- (1) Select the driver in the program which corresponds to the MidiX data on disk
- (2) Choose *Drivers/Open/MidiX* and use the file selector to choose the MidiX file to convert
- (3) An editor window will open containing the MidiX data

See Also: [File Conversion Window](#)

See Also: [Driver List Window](#)

Play Chord (Midi Menu)

Purpose:

Play Chord allows you to listen to a Patch under development in a Patch editor window or any Patch you want from a Bank Editor, DBase, Library, or Group. Play Chord automatically picks up the Midi Channel assigned to the window so you should always hear the chord on the correct Midi Port and channel. The chord played by Play Chord can be defined in the Tones Window.

Directions:

(1) Choose *Midi/Play Chord* or press F10

See Also: [Tones Tab](#)

See Also: [Midi Menu](#)

Play Sequence (Midi Menu)

Purpose:

Play Sequence allows you to listen to a Patch under development in a Patch editor window or any Patch you want from a Bank Editor, DBase, Library, or Group. Play Sequence automatically picks up the Midi Channel assigned to the window so you should always hear the chord on the correct Midi Port and channel. The chord played by Play Sequence can be defined in the Tones Window.

Directions:

(1) Choose *Midi/Play Sequence* or press F11

See Also: [Tones Tab](#)

See Also: [Midi Menu](#)

Midi Quest (Cakewalk) MFX Plugin

Using the Midi Quest MFX Plugin

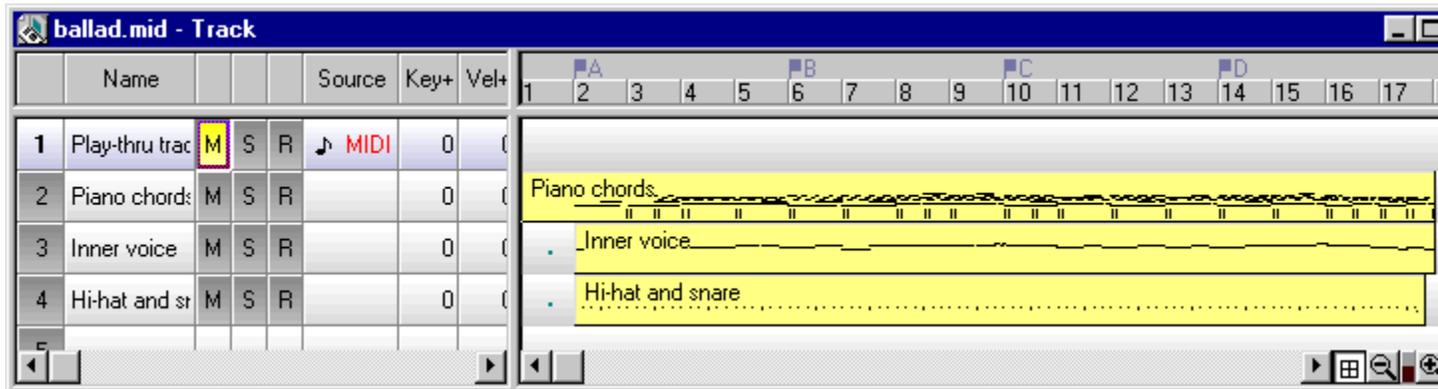
The Cakewalk MFX version of Midi Quest is designed so that the five main display/editing windows (Group, Library, Bank, Patch, DBase) can each be used entirely from within Cakewalk. This feature allows you to perform all of the functions that you normally use Midi Quest for from within Cakewalk. In addition, the plugin can be configured to automatically send the SysX data in response to Cakewalk events. Essentially, this automates Midi Quest's operation so that you no longer need to go through the time consuming process of exporting data as a MIDIX file and importing it into Cakewalk.

Installing the Cakewalk MFX Plugin

If Cakewalk is already installed when you install Midi Quest then the Midi Quest MFX plugin will automatically be installed. If Cakewalk is not installed or you have an older version (less than v9) then you may need to reinstall Midi Quest in order for the plugin to be correctly configured.

Creating Your First Midi Quest MFX Plugin

First, open a Cakewalk sequence as in the following example:



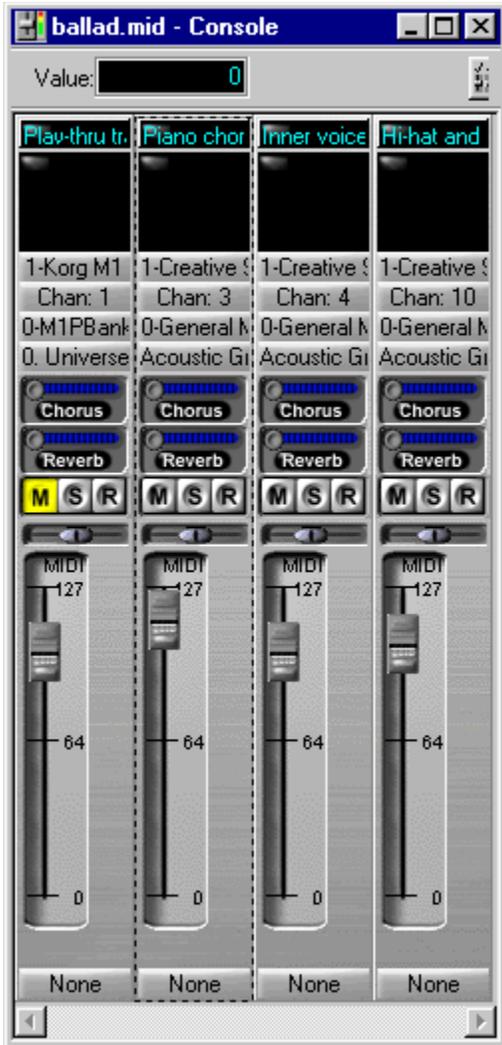
Next, choose View|Console from the menus and you will see the console window:

-  Piano Roll
-  Event List
-  Staff
-  Audio
- Oh- Lyrics
-  StudioWare

-  Console
-  Video
-  Big Time
-  Markers
-  Tempo
-  Meter/Key
-  Sysx

-  StudioMix

- Layouts...
- Toolbars...



Right clicking in the black area just below the track name will display a popup menu similar to this one:

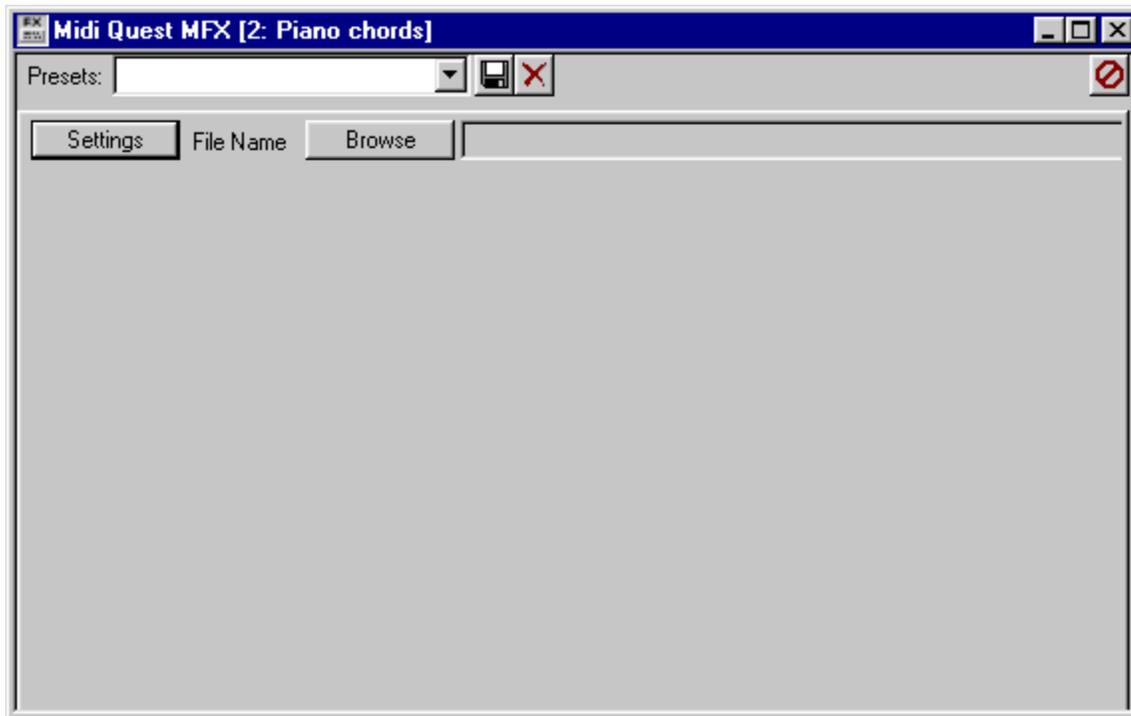


If you do not see "Midi Quest MFX" in the list you will need to go back and reinstall Midi Quest. Assuming "Midi Quest MFX" is displayed, select it and you will see the following in the track you selected:



The Midi Quest MFX Window

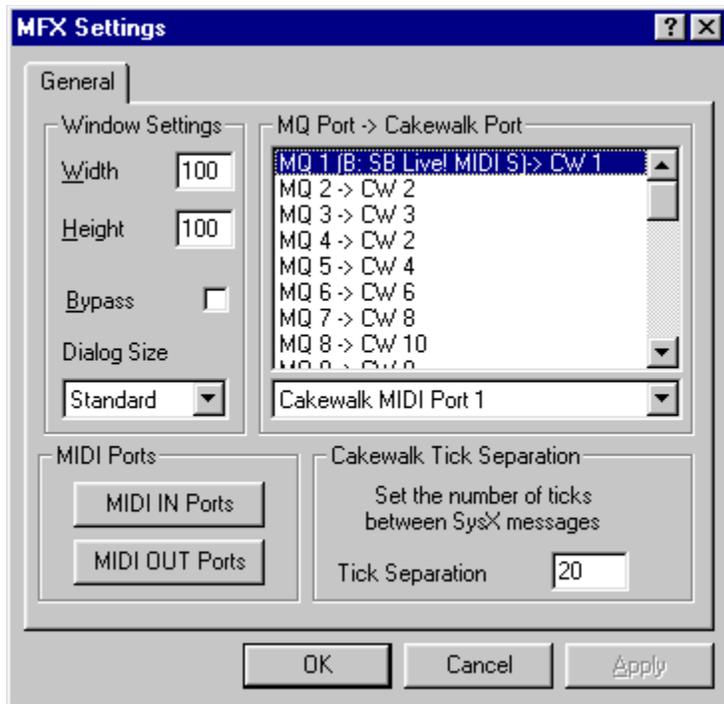
Double click on "Midi Quest" and you will see the following window:



This is the window where you will perform all of your Midi Quest activities.

Selecting MIDI Ports

The first thing you will need to do is select the MIDI ports that Midi Quest will use. While Midi Quest allows you to select a different set of MIDI interfaces for MFX, in most cases you will probably use the same configuration. To select MIDI Ports, press the **Settings** button and you will see the following dialog:



To select MIDI OUT ports, click on the **MIDI OUT Ports** button. Use the displayed [Midi Out Ports](#) dialog to select the ports you wish to use.

To select MIDI IN ports, click on the **MIDI IN Ports** button. Use the displayed [Midi In Ports](#) dialog to select the ports you wish to use. Remember that virtually all activity in MFX will be in the form of sending MIDI data, not receiving it so you will probably not need to open any MIDI IN ports.

Important: It is also important to note that both Midi Quest and Cakewalk will need to have these ports open simultaneously so you need to use a MIDI interface with "multi-client" drivers (or constantly enable and disable MIDI drivers in Cakewalk and Midi Quest MFX). Multi-client drivers allow more than one MIDI application to access the interface at the same time. Companies which offer multi-client drivers include MIDI Man, Opcode, MOTU, and Turtle Beach. You will find that the driver which come with your standard sound card, including those made by Creative Labs, are **not** multi-client.

Configuring Midi Quest MFX for Output in Cakewalk

You have now reached the last stage of configuring Midi Quest MFX for use in Cakewalk.

In this stage we need to map Midi Quest MIDI port numbers to Cakewalk MIDI port numbers. Why do we need to do this? All of your Midi Quest data files, be they Groups, Banks, Patches, Libraries, or DBases, all have port numbers stored in them so that Midi Quest knows where to send the file. In Cakewalk, you may have additional MIDI ports selected and the ports may be in a different order. What must be done here is create a "map" so that the SysX data is sent to the correct MIDI port.

This map is created in the General tab of the MFX settings in the **MQ Port -> Cakewalk Port** section. This list is used to convert a Midi Quest port number to a Cakewalk port number. On the left hand side of the arrow is the port number as it appears in Midi Quest. If you opened a MIDI port for that port number in the previous section, **Selecting MIDI Ports**, then you will see the name of the MIDI port as well. On the right hand side of the arrow is the MIDI port number that the SysX data will be sent to by Cakewalk.

The idea is that these ports should be the same. So if the port on the left is "Creative Labs MIDI Out" for Midi Quest, then on the right hand side you should select the port number when "Creative Labs MIDI Out" is open in Cakewalk. In an ideal world, we would be able to display the names of the ports open in Cakewalk and do the matching for you. Unfortunately, Cakewalk doesn't provide this information to plugins so we can't.

Now that you, hopefully, understand what you need to do, lets do it. Click on first line of the list, **MQ 1 -> CW 1**. If the first port you have open in Midi Quest is not the first port in Cakewalk, click on the drop-down list where it says **Cakewalk Midi Port 1** and select the Cakewalk MIDI port where you would like SysX assigned to port 1 in Midi Quest to be sent.

After you have made the assignment, move on the the second assignment in the list, **MQ 1 -> CW 2** and repeat the process in the previous paragraph. Continue making the assignments until you have assigned all of the ports that you use in Midi Quest to ports you use in Cakewalk.

Remember, this port map only applies to the automation features of Midi Quest MFX. That is, when you have Cakewalk automatically send SysX data to an instrument. Any editing or auditioning that you do in the Midi Quest MFX windows is sent to the MIDI ports that you opened in **Selecting MIDI Ports** above.

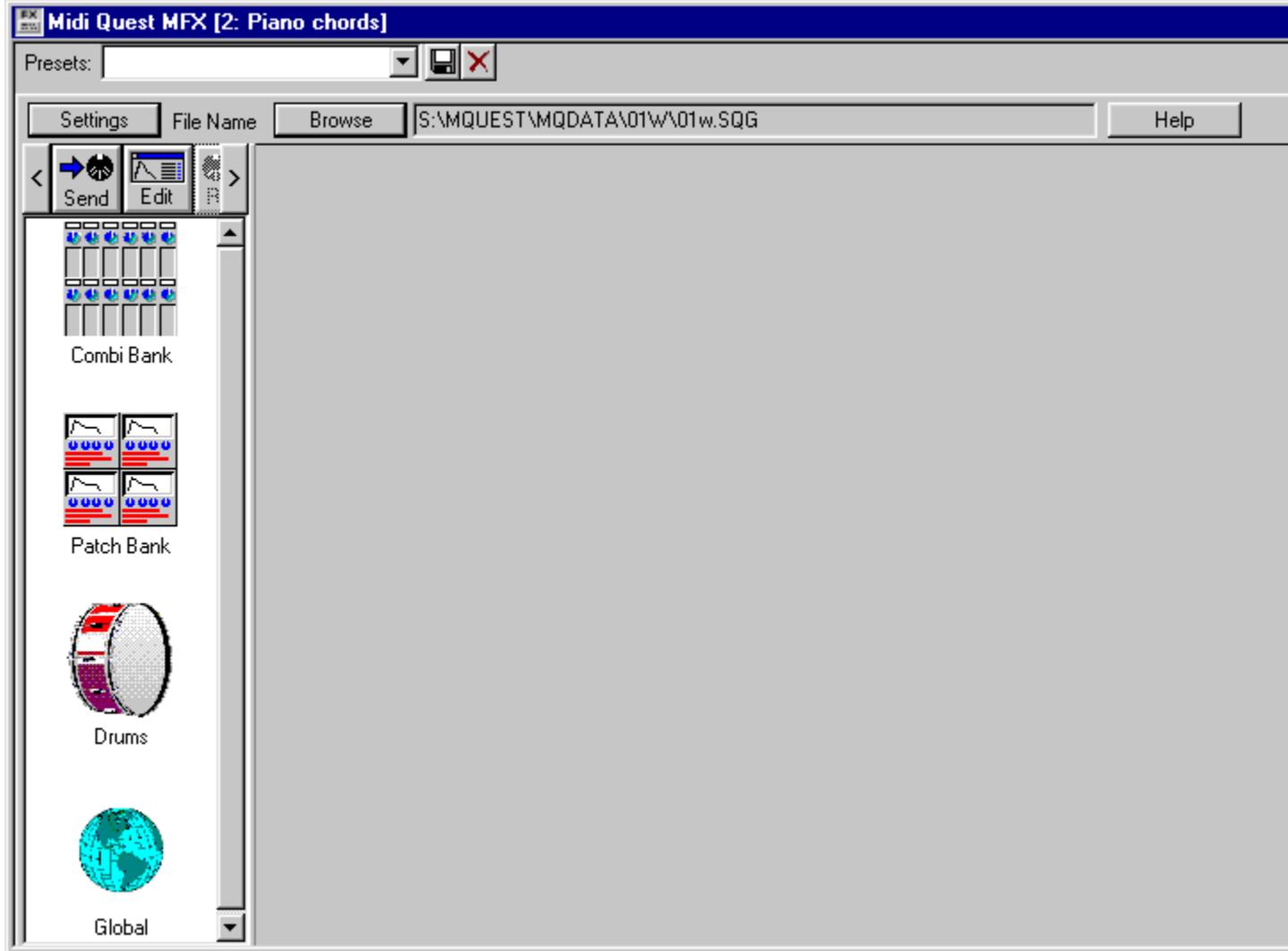
For more information on MFX automation, when you are finished with this page go to [MFX./VST Settings](#).

After setting up your port map, press the OK button to close the MFX settings dialog.

Selecting a Midi Quest File

The next thing you will want to do is select one of the Midi Quest files that is used by the current song. To do this, press the **Browse** button and a file selector is displayed. Use this selector to choose a Midi Quest file to load. You can choose any Group (.sqq), Library (.sql), DBase (.sqd), Bank, or Patch file. After selecting the file, press the OK button to load it.

If you selected a Group file, you will see something like this:



You will notice that this window look exactly like Group in Midi Quest, and it is. You have all of the same functionality and can edit banks and patches from within the Group. Just remember the virtually all of a particular window's functionality can be accessed by right clicking the mouse button in that window.

That's it. You're now set up to work with Midi Quest MFX in Cakewalk just as if you were using Midi Quest as a separate program.

For information on all of the options found in the MFX Settings dialog, please click [here](#).

See also: [Midi Quest VST](#)

See also: [MFX./VST Settings](#)

Play Last Note (Midi Menu)

Purpose:

Play Last Note will literally replay the last note played by the program. This is particularly useful if you are using the right mouse button to play notes in the Patch Editor window. If you play a particular note you would like to continue using, switch to the Play Last Note, easily triggered from the computer keyboard's F12 key.

Directions:

(1) Choose *Midi/Play Last Note* or press F12

See Also: [Tones Tab](#)

See Also: [Midi Menu](#)

Play Note (Midi Menu)

Purpose:

It is possible to play any of the first eight notes defined in the Tones Window as independent pitches. This is particularly useful if you are developing a drum configuration. In the Tones window, assign the first eight notes in the window to the different drum notes you are working on. If you switch over to the Patch Editor Window where you are editing your drum setup, you can use the function keys F2 through F9 to play the different notes.

Directions:

(1) Choose *Midi/Play Note/Note* or press F2 - F9

See Also: [Tones Tab](#)

See Also: [Midi Menu](#)

Cut (Edit Menu)

Purpose:

Cut is used to remove one or more items from a Group, DBase, or Library window and place them in a program clipboard so that they may be pasted into the program at a later time.

Please see the window specific "Cut" for more information.

Directions:

- (1) Choose the items in the window to cut
- (2) Choose *Cut* from the edit menu

See Also: [DBase Cut](#)

See Also: [Library Cut](#)

Copy (Edit Menu)

Purpose:

Copy is used to copy one or more items from a Group, DBase, Library, Bank, or Patch window and place them in a program clipboard so that they may be pasted into the program at a later time.

Please see the window specific "Copy" for more information.

Directions:

- (1) Choose the items in the window to copy
- (2) Choose *Copy* from the edit menu

See Also: [Bank Editor Copy](#)

See Also: [Patch Editor Copy](#)

See Also: [Group Copy](#)

See Also: [DBase Copy](#)

See Also: [Library Copy](#)

Paste (Edit Menu)

Purpose:

Paste is used to move the items currently in a program clipboard into a Group, DBase, Library, Bank, or Patch Window.

Please see the window specific "Paste" for more information.

Directions:

To paste items into a window:

- (1) Items must have been previously cut or copied to the appropriate clipboard
- (2) Choose *Paste* from the edit menu

See Also: [Bank Edit Paste](#)

See Also: [Patch Editor Paste](#)

See Also: [Group Paste](#)

See Also: [DBase Paste](#)

See Also: [Library Paste](#)

Clear (Edit Menu)

Purpose:

Clear is used to permanently remove one or more items from a Group, DBase, or Library window.

Please see the window specific "Clear" for more information.

Directions:

To clear items in a window:

- (1) Choose the items in the window to clear
- (2) Choose *Clear* from the edit menu

Note: once items are cleared from a window, then can not be recovered!

See Also: [Bank Editor Clear](#)

See Also: [DBase Clear](#)

See Also: [Library Clear](#)

Open Driver List (Driver List)

Purpose:

Open Driver List loads a Driver List into the Driver List Window. This function is similar to choosing *Files/Open...* except that when chosen from the Drivers menu, the File Selector is automatically placed in the correct directory with the correct extension for loading a Driver List.

Directions:

To load a new Driver List into the Driver List Window

- (1) Choose *Drivers/Open Drv List*
- (2) Use the File Selector to load a different Driver List

See Also: [Driver List](#)

Auto Load (Driver List)

Purpose:

Auto Load is a toggle parameter. It only affects the drivers for which the Grp is already checked.

This parameter helps control how a Group is created for the instrument. By default, Auto Load is checked for each driver that is part of the instrument's Group. When Edit is pressed to load a new Group from an instrument, the program collects data from the instrument for each driver where both Grp and Auto Load are set. If a driver is part of the group but does not have Auto Load set, an uninitialized copy of the data is inserted in the new Group.

For more information, please see [Groups](#).

Directions:

To make a driver in an instrument Group auto-loading:

- (1) Select the driver in the Driver List that is not currently part of the instrument group
- (2) Choose *Drivers/Update Settings/Auto Load*
- (3) 'Auto' will appear in the drivers status line

To cancel the auto-loading for a particular driver

- (1) Select the driver in the Driver list that is currently part of the instrument group
- (2) Choose *Drivers/Update Settings/Auto Load*
- (3) 'Auto' will disappear in the drivers status line

See Also: [Group Window](#)

See Also: [Driver List Window](#)

Blend (Library)



Purpose:

Blend creates new Patches by randomly picking parameter values from two selected Patches.

The first Patch created is largely biased toward the first selected Patch. As new Patches are created, the bias increases toward the second selected Patch. The last Patch created is largely biased toward the second selected Patch. This produces the aural effect of gradually moving from the first Patch through to the second Patch. Only values actually present in either of the two Patches are used.

Directions:

To Blend Patches:

- (1) Select the two Patches you wish to Blend
- (2) Choose *Library/Randomize/Blend*
- (3) Why not try auditioning your new Patches?

See Also: [Bank Edit Blend](#)

See Also: [Library Window](#)

Driver List Contents

The Driver List contains a list of all of your instruments. It is organized hierarchically, a thick bar represents each instrument and is called an Instrument Bar. On the left of each bar is an icon representing the instrument. To the right is the instrument's name, Comm Channel, Cntrl # (optional), Midi Channel, MIDI Port In, and MIDI Port Out. A thin bar under each instrument bar represents a particular driver or data type for the instrument.

To expand the instrument to see each of its drivers, double click anywhere in the text portion of the instrument line. Double clicking on the text of an already expanded instrument listing will shrink the listing back down.

For more information on the Comm Channel, Midi Channel, Cntrl#, Port In, and Port Out, please see the [Settings Dialog](#).

Each instrument driver has four parameters:

Driver Name

The first parameter is the driver's name. This parameter is not editable.

Pch#

The Pch # is only visible when it is used by the driver. It is used to select which Patch or Bank is loaded from the instrument when the driver is designed to make a selection. The Fast Tips help for the drivers will explain what the Pch # is used for in each particular driver's instance.

Group

When "Grp" is displayed in the third column, it indicates that the driver is part of the instrument's Group. When a new Group is created or loaded from an instrument, the data file for this driver is included.

Auto Load

When "Auto" is displayed in the fourth column, it indicates that when a new Group is created by loading data from the instrument, the data file created by this driver is part of the Group. If the driver is part of the Group but Auto is NOT displayed, the data file is added to the Group but it is zeroed and not loaded from the instrument.

Group



Purpose:

A "Group", by default, holds a complete dump of an instrument's permanent memory. In the case of the M1, the instrument's permanent memory is made up of a Patch Bank, Combi Bank, Sequence, and Global data. When a Group is loaded or a new Group created, it will hold each of these data files. The purpose of the Group is to allow you to treat your instrument as a whole rather than as a collection of individual parts.

Groups are extremely important because they link the individual data files of an instrument together, and treat them as a whole. In a Group, Midi Quest can provide intelligent Bank editing and organizing features when compared with individual Bank data files.

For example, assume you are editing two M1 Combi Banks, one from each of two Groups. If you copy Combis from one Bank to the other, Midi Quest will automatically copy the Combi's Patches as well. (There must be empty locations in the bank for the Patches to be copied!). This is impossible with individual data files. The Combis are considered "Parents" because they call, or make use of, other types of data in the instrument, in this case Patches. The Patches are considered "Children" since they are lower in the hierarchy. Additionally, swapping "Child" patches within a bank will automatically adjust the Parent Multi/Combi patches above to reflect the changes.

An instrument's Group is completely defined from the Driver List. Each driver for an instrument which has "Grp" set is considered part of the Group. If you wish to add or remove a data file from the Group definition, simply select the driver to change in the Driver List and choose Drivers/Update Settings/Group Member to toggle Group membership. The next Group created will have only those data files you have selected. For more information, see the Driver List chapter.

The other notable feature which is controlled from the Driver List is Auto Load. By default, Auto Load is active for each driver that is a Group member. However, there may be circumstances where you do not want to load a particular data file at the same time as the other data files in the Group. If you turn Auto Load off, a data file for the driver is placed in the Group but it is not activated until data is loaded into the Group from the instrument at some future time.

Pressing the Edit button on the icon bar of the Group window alternately shows and hides the Group's split window. The split window edits the data file currently selected in the Group. When a Group is created, it starts in split mode. Clicking on any data file in the Group will automatically open an editor in the right hand side of the window. If there is no editor for the data file, the editing side of the split window remains empty. Any changes made to a data file in the edit window are automatically stored in the Group. To access the Patch or Bank menu functions use the right mouse button.

A useful function, particularly if you keep your Library sorted, is the Quick Search. After selecting any

patch in the Library, you can press any alphanumeric key on the computer keyboard the you will automatically jump to the next Patch that starts with the selected letter. If you know that the patch you are looking for starts with an H, this is an extremely fast method of getting there.

It is possible to have as many Groups as desired displayed simultaneously on the screen (limited only by available RAM) so it is even possible to cut or copy components of one DBase into another.

For information on each column in the DBase, see [Group Contents](#)
To send the entire contents of a DBase to the instruments, see [Sending a Group](#).
To send one or more data files in a DBase, see [Sending Data Files](#).

Group Menu

The Group contains 16 menu functions with which to organize the data files in the Group. These functions are all contained in the Group menu. The Group window must be activated before you can see and access the Group menu.

Cut	Cut the selected Data File(s) to the DBase clipboard
Copy	Copy the selected Data File(s) to the DBase clipboard
Paste	Paste the Data Files in the clipboard to the current DBase
Clear	Remove selected Data Files from the DBase
Select All	Select all entries in the DBase for processing
Open/Editor	Open the Data File into a Patch or Bank Editor
Open/SysX View	Open the Data File into a SysX View Window
Open/Update Settings...	Open the Update Settings Dialog for the selected data file
Sort By	Sort the DBase by one of 5 methods
Disk/Load Data File	Load a Data File from disk into the DBase
Disk/Save Data File	Save a Data File in the DBase to disk as a separate file
Disk/Link a Data File	Link a Data File from disk into the DBase
Disk/Save Data File & Link	Save a Data File to disk and link the DBase to it
Disk/Resolve Links	Make any linked Data Files into files stored in the DBase
Receive From Instrument	Add data for the Driver List drivers into the DBase
Move To...	Add the selected Data File to a different window

Open Toolbar (Options Menu)

Purpose:

Open a new toolbar in the position indicated in the Preferences [Misc Tab](#).

Directions:

To open a new toolbar:

(1) Choose *Options/Open Toolbar*

See also: [Options Menu](#)

Drag and Drop (Common Elements)

Purpose:

With Drag and Drop It is possible to transfer data between any of the following windows: Bank Editor, Patch Editor, Group, DBase, and Library. Choose the data files to transfer in the source window, next reselect one of the selected data files, hold the mouse button down, drag the cursor over the destination window and release the mouse.

This feature makes transferring data files between windows fast and easy. Basically it provides the same capabilities as the Move To Dialog without requiring the dialog.

The transfer will behave somewhat differently depending on what the source and destination window types are. This information is laid out in the chart below. The one important factor to remember when dealing with Bank Editors, Patch Editor, Groups, and Libraries is that the data in the source and destination windows is compatible. For example, you can't drag and drop M1 Patches into an M1 Combi Library and you can't drag and drop a DX7 Voice into an M1 Group.

Bank Editor Window to:

- Bank Editor Window
 - for 1 patch in a Bank
 - when dropped in the same bank
 - source is pasted, swapped, or inserted at the destination patch based on the setting in *Options/Properties/Bank Editor* tab
 - when dropped in a different bank
 - source is pasted or inserted at the destination patch based on the setting in *Options/Properties/Bank Editor* tab
 - otherwise:
 - the selected Patches overwrite the selected Patches in the destination Bank
 - if there are no selected patches, Patches are placed in empty locations
- Patch Editor Window
 - the first selected Patch is transferred into the editor
 - the data must be compatible to transfer
- Group Window
 - the entire destination Bank is overwritten by the source Bank
 - if a Bank of the same type does not exist, the data is ignored
- DBase Window
 - entire Bank is moved to the DBase
- Library Window
 - dialog is displayed to choose between moving all Patches or only selected

Patch Editor Window to:

- Bank Editor Window
 - the Patch overwrites the first selected Patch in the Bank
 - the data must be compatible to transfer
- Patch Editor Window
 - the entire Patch is transferred from one Patch to the other
 - the data must be compatible to transfer
- Group Window
 - the Patch overwrites the identical Patch in the Group
 - if there is no compatible data in the group the data is ignored
- DBase Window
 - the Patch is added to the DBase
- Library Window

the Patch is added to the Library
the data must be compatible with the Library to transfer

Group Window to:

Bank Window

the source Bank overwrites the entire destination Bank
if the Banks are not of identical type the data is ignored

Patch Editor Window

the selected Patch is transferred into the editor
for a successful transfer the data must be the same as in the editor

Group Window

the selected data files overwrite the same data files in the destination Group
if an identical data file is not found in the group it is ignored

DBase Window

the selected data files in the group are transferred into the DBase

Library Window

if the data is a Patch it is transferred into the Library
if the data is a Bank each Patch in the Bank is transferred into the Library
the data is only transferred if it is compatible with the Library

DBase Window to:

Bank Window

the source Bank overwrites the entire destination Bank
if the Banks are not of identical type the data is ignored

Patch Editor Window

the selected Patch is transferred to the Patch Editor
for a successful transfer the data must be the same type as in the editor

Group Window

the selected data files of the DBase overwrite the same data files in the Group
if the identical data is not found in the group it is ignored

DBase Window

the selected data files in the DBase are transferred into the destination DBase

Library Window

if the data is Patch it is transferred into the Library
if the data is a Bank each Patch in the Bank is transferred into the Library
the data is only transferred if it is compatible with the Library

Library Window to:

Bank Window

the selected Patches are transferred to the library
for a successful transfer the data must be the same type as the Bank
the Patches are first added to selected Patches then empty Patches in the Bank

Patch Editor Window

the selected Patch is transferred into the editor
for a successful transfer the Patch must be the same type as in the editor

Group Window

the first selected Patch in the library is transferred to the Group
there must be an identical data type in the Group of the Patch is ignored

DBase Window

the selected Patches are transferred to the DBase

Library Window

the selected Patches are transferred to the destination library
the data is only transferred if it is compatible with the Library

Explorer to:

Group Window

if data of that type exists in the Group, it is replaced by the new file

DBase Window
that data file is added to the database
MIDIX files are also added to the database

Library Window
if the data types match, the data file is added to the library

Background Window
the file is opened into its own window
any MIDIX (.SMF) file is converted to a Midi Quest file and loaded in a SysX View

Right Click

right clicking in the Midi Quest frame window displays the MRU file list
it also displays a list of desktop files saved in the MQData directory

Directions:

To transfer a Bank in a Group to a Library:

- (1) The Bank to transfer must be of the same data type as the Library
- (2) Select the Bank in the Group to transfer to the Library
- (3) Click again on the Bank in the Group
- (4) Drag the mouse over the Library and release the mouse button
- (5) Each of the Patches will be transferred from the Bank in the Group to the Library

See Also: [Bank Editor Window](#)

See Also: [Patch Editor Window](#)

See Also: [Group Window](#)

See Also: [DBase Window](#)

See Also: [Library Window](#)

Group Contents (Group)

Data Files can be dragged and dropped between the Group and other windows in the program. For more information, please see [Dragging and Dropping](#).

Individual entries in the Group can be reordered by following these steps:

- 1) Choose the data file to move;
- 2) hold down the ALT key on the keyboard;
- 3) use the up and down arrow keys to move the data file to the desired location.

Each data file is shown as a row in the Group. The first row of the Group contains the column title: Type. The first column has no title but will show a disk icon if the entry is linked to a disk file instead of actually being stored in the Group.

Link

The first column is untitled, but will show a disk icon if the row contains a disk linked data file. Disk linked files are references in the Group to other Sound Quest data files stored on disk. These files are only referenced by the Group and are not actually stored in the Group. This allows any Group, DBase or Library to have access to the same Patches and Banks. Any changes to the disk based data file will also be reflected in the Groups, DBases and Libraries that have links to the source data file. All of the text in the row will appear in blue instead of black. For more information on linking to data files, please see [Link a Data File](#), [Save a Data File and Link](#), and [Resolve Links](#).

Type

The Type column shows the type of SysX data stored in the row. For example, Patch Bank. This parameter is not editable and gives you a list of the different types of data contained within the Group.

State

A Group can have two different types of data files, those which are [active](#) and those which are not. [Active](#) data files are those which have been loaded from the instrument or added to the Group from disk. Active data files show active in the State column and can be sent to the instrument. Those data files in the Group which are inactive have no indicator in the State column and highlight in gray. These data files can't be sent to the instrument as the program considers them to be uninitialized place holders.

The concept of a Group is to represent the entire contents of an instrument. However, you will often find that there are portions of your instrument where you never change the settings. In these instances, you can choose not to send and receive that data by making it inactive.

See also: [Group Window](#)

Contract All (Driver List)

Purpose:

Contract All is used to automatically contract the Driver List so that only the instrument bar for each instrument is displayed. Use this function if you would like to hide all of the drivers that are currently displayed.

Directions:

(1) Choose *Drivers/Contract All*

See also: [Driver List Window](#)

Link A Data File (Group)

Purpose:

Link A Data File is very similar to [Load A Data File](#). Both are used to bring a particular data file into the Group and replace the existing data file. Link a Data File, however, does not actually store the data file in the Group, it just stores a reference to the file. This means that any changes made to the source files on disk are automatically reflected in the linked data in the Group. **It also means that if the data file on disk is erased, it will not be available to the Group and will be removed!**

For additional information on the advantages of linking data files, please see [DBase Link A Data File](#).

Directions:

To link one or more data file:

- (1) Choose *Group/Link a Data File*
- (2) A file selector is displayed
- (3) Choose the one or more data files to be linked to the Group

See Also: [Group Window](#)

Copy (Group)



Purpose:

Copy duplicates the selected data files and places the copies in the Group Clipboard. The data files can now be pasted into a different Group.

Note: Before adding the data files to the Group Clipboard, the program deletes any data files currently stored there.

Directions:

To Copy a data file and place it in the Group clipboard:

- (1) Select the data file(s) to copy
- (2) Choose *Group/Copy*

See Also: [Group Window](#)

Load Data File (Group)

Purpose:

Load Data File replaces one or more of the data files in the Group with a data file of the same type on disk.

If a new Group is being built, some of the files required may be currently stored on disk. Instead of opening the files as editors and transferring them into the Group, the files can be loaded directly in from disk using Load Data File.

In some cases, data files for certain instruments are not editable. Because these files can not be opened as editors, there is no means available to transfer the data into the Group from within the program. Use this feature to load the data directly from disk.

Directions:

To load one or more data files from disk into a Group:

- (1) Choose *Group/Disk/Load Data File*
- (2) Use the File Selector to select the one or more data files you wish to load

See Also: [Group Window](#)

Move To (Group)



Purpose:

Move To moves one or more data files into any of the following destination windows: Bank Editor, Patch Editor, Group, DBase, or Library window.

Directions:

To move one or more data files into another window:

- (1) Choose the data files to move
- (2) Choose *Group/Move To...*
- (3) The selected data files are moved to the destination window

See Also: [Move To Dialog](#)

See Also: [Group Window](#)

Open SysX View (Group)

Purpose:

Open SysX View moves the selected data file's SysX data into the SysX View Window. From there, the data can be viewed in its raw hexadecimal form. This feature is useful for viewing data while developing and testing instrument drivers in MIDI QUEST.

Directions:

To view the data file in hexadecimal format:

- (1) Select the data file to view
- (2) Choose *Group/Open/SysX View*

See also: [SysX View Window](#)

See also: [Group Window](#)

Open Editor (Group)

Purpose:

Open Editor opens any editable data file in the Group. If the selected data file is a Bank, a Bank Editor Window is opened to edit the data file. Otherwise a Patch Editor Window is opened to edit the data file. For additional information on editing Banks and Patches, see the [Bank Editor Window](#) and the [Patch Editor Window](#).

Directions:

To edit a Patch or Bank from a Group:

- (1) Select the data file to edit
- (2) Choose *Group/Open/Editor*.

See Also: [Group Window](#)

Receive From Instrument (Group)



Purpose:

Receive From Instrument loads one or more data files directly from the instrument into an existing Group.

This feature allows one or more components of the group to be overwritten with the current contents of the instrument. You can use this function if you have made changes to a portion of your instrument's memory and want to update the Group to match it.

Note: any file loaded from the instrument will replace the contents of that same data file in the group. This operation can not be undone.

Directions:

To load data files from an instrument to a Group:

- (1) Select one or more of the data files in the Group
- (2) Select the `Recv` button or *Group / Receive From Instrument*
- (3) The program will load data files from the instrument directly into the Group

See Also: [Group Window](#)

Paste (Group)



Purpose:

Paste copies the data files from the Group Clipboard into the active Group.

Before data files can be pasted into a Group, data files from another Group must be Copied. This places the data files in the Group clipboard. Once in the clipboard, the data files can be pasted to the Group.

The primary use for Copy/Paste is to move data files between Groups.

Note: if you copy data files from one Group and then attempt to paste them to a different, incompatible Group, the data files will be ignored.

Directions:

To paste the contents of the Group Clipboard into a Group:

- (1) Copy one or more data files from a Group
- (2) Click on the Group Window to paste the data into
- (3) Choose *Group/Paste*

See Also: [Group Window](#)

Save Data File (Group)

Purpose:

Save Data File saves a Group data file to disk.

This is similar to [Save Data File and Link](#). Both functions save the selected data file to disk, however, with Save Data File and Link the data file in the Group becomes a linked disk file. The data file itself is no longer stored in the Group.

Directions:

To save a Group data file to disk:

- (1) Select the data file you wish to save
- (2) Choose *Group/Disk/Save Data File*
- (3) Use the File Selector to name and save the data file

See Also: [Group Window](#)

Resolve Links (Group)

Purpose:

Resolve Links converts all linked disk files into data files that are stored in the Group.

For instance, if you wish to take a Group which references linked disk files to a studio, you will have to take the Group and all of the linked disk files unless you use Resolve Links before you leave for the studio. Resolve Links places any linked disk files directly into the Group so you can just take your Group to the studio. Everything you need will be there.

Directions:

(1) Choose *Group/Disk/Resolve Links*

See Also: [Link a Data File](#)

See Also: [Save a Data File and Link](#)

See Also: [Group Window](#)

Group Banks and Singles (Driver List)

Group Banks

Group Singles

Purpose:

These three options are used to automatically select various drivers in the Driver List.

"Group Banks and Singles" selects all of the drivers which are part of each instrument's Group, either Banks or singles. This is a quick way to select all of the drivers in the Driver List which are necessary to take a full snapshot of your system's configuration. These drivers will collect all of the stored parameter settings from each of your MIDI instruments. Once selected, choose the DBase button or *Drivers/Instrument to/DBase* to load the data into a DBase.

"Group Banks" selects all of the drivers which are part of each instrument's Group that are Banks. This function selects all of the Bank drivers for each of your instruments, giving you a quick way to save all of your Banks. Once selected, choose the DBase button or *Drivers/Instrument to/DBase* to load the data into a DBase.

"Group Singles" selects all of the drivers which are part of each instrument's Group and are single blocks of SysX data (not Banks). This function is used to quickly select all of the single drivers for each of your instruments, giving you a quick way to save all of your instrument's individual settings. Once selected, choose the DBase button or *Drivers/Instrument to/DBase* to load the data into a DBase.

Directions:

Choose either *Drivers/Select/Group Banks and Singles*, *Drivers/Select/Group Banks*, or *Drivers/Select/Group Singles* depending on which drivers you would select.

See also: [Driver List Window](#)

Select All (Group)

Purpose:

Select All is a quick way to select the entire Group.

Directions:

Choose *Group/Select All*.

See Also: [Group Window](#)

Save Data File and Link (Group)

Purpose:

Save Data File and Link saves the selected Group data file to disk. It then modifies the Group so that the data file becomes a linked disk file. The data file itself is no longer stored in the Group.

This is similar to [Save Data File](#). Both functions save the selected data file to disk, however, with Save Data File the data file remains stored in the Group.

For a complete description of the advantages of file links, please see [Link a File](#)

Directions:

To save a data file and link:

- (1) Select the data file to save to disk and make a linked file
- (2) A file selector is displayed
- (3) Choose an appropriate directory for the file (if necessary) and give the file a name
- (4) Click on the save button.

See Also: [Group Window](#)

Sending a Group (Group)

Purpose:

Sending a Group sends each data file in a Group to the instrument. In most cases, this completely reconfigures your instrument. Only Group member which are currently active are sent to the instrument.

Use Sending a Group when you begin editing a different Group and would like update the instrument's memory to match the Group. This makes it easy to switch between completely different instrument configurations.

Directions:

To send all of the data stored in a Group to your instruments:

- (1) click on the Window containing the Group you wish to transmit
- (2) Choose *Midi/DBase/Group to Instrument*

See Also: [Group Window](#)

Sending Data Files (Group)



Purpose:

Click once on the Send button to send selected data files in a Group to the instrument. Double click on the Send button to send all of the data file in the Group. Any Group member which is not active (highlighted in gray) will not transmit. The program considers these files to be uninitialized and not part of the Group until data of that type is added to the Group.

Directions:

To send one or more files in a Group to the instrument:

- (1) Select the data file(s) to send
- (2) Select `Send' button or choose *Midi/Data to Instrument*

See Also: [Group Window](#)

Update Settings (Group)



Purpose:

Update Settings opens the Settings Dialog which is used to edit the basic settings of the Group. Basic settings include: Comm Channel, MIDI channel, MIDI In Ports, MIDI Out Ports, Patch #, Cntrl #, and key words assigned to the Group's data files.

For more information, you should see the [Settings Dialog](#).

Directions:

To change the basic settings for the Group

- (1) Choose *Group/Open/Update Settings...*
- (2) Make any necessary changes to the basic settings

See also: [Settings Dialog](#)

See also: [Group Window](#)

Custom Set (Driver List)



Purpose:

Custom Set selects a preset combination of drivers in the Driver List. To utilize this function, you must have previously created one or more Custom Sets. Choose *Drivers/Select/Custom Set...* and your Custom Sets are displayed in a dialog. Choose one. Each Driver Listed in the Custom Set is selected.

In the Custom Set Dialog, there is one selectable option called "Use Comm Channel". You should only check this option if you have two or more instruments of the same type such as two M1s or two or more instruments from the same family (JVs, DXIIs, D-10/20/110, U-20/220, etc). In these instances, it is necessary to also use the Comm Channel in order for the program to determine whether the driver should be selected or not.

To create a custom set, select the drivers that are to be part of the set and choose Drivers/Select/Make Custom Set.

Directions:

- (1) Choose *Drivers/Select/Custom Set...*
- (2) Use the dialog to select the custom set to load
- (3) Check "Use Comm Channel" to select the driver only if its Comm Channel matches
- (3) The program will select the drivers defined in the Custom Set

See Also: Make Custom Set

See also: Driver List Window

Make Custom Set (Driver List)

Purpose:

Make Custom Set creates a custom list of drivers which may be quickly selected together at some time in the future.

This function saves you time when creating DBases. If you are commonly selecting a particular set of drivers before creating a DBase, you can save these drivers as a Custom Set. Give your Custom Set a name. In future, you need only choose the Custom Set by name and the appropriate drivers will be automatically selected. Once selected, your Custom Set is easily loaded into a new DBase by pressing the DBase button.

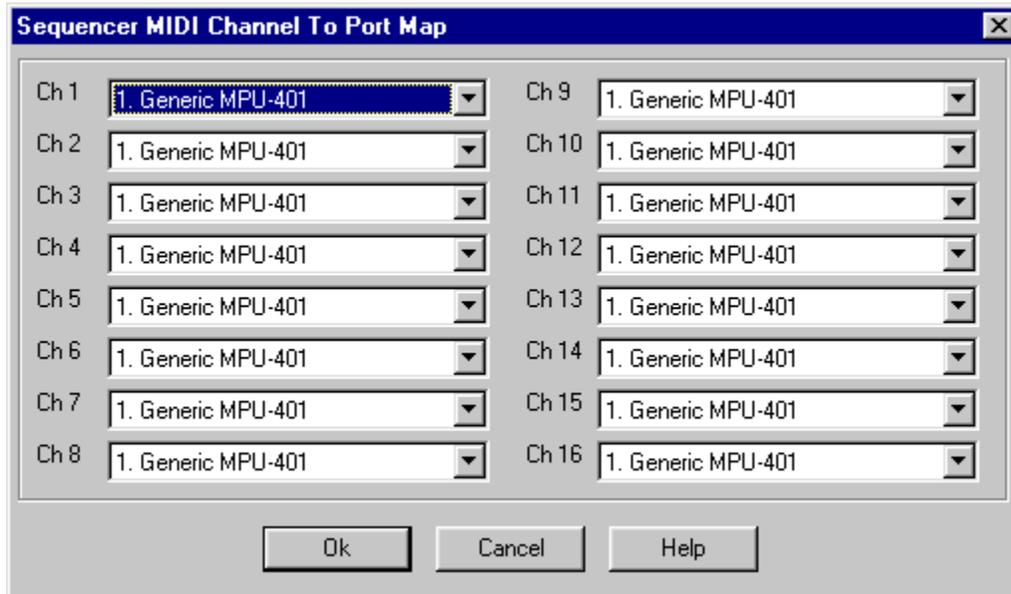
Directions:

To create a custom set:

- (1) Select the drivers in the Driver List that you would like to be part of the set
- (2) Choose *Drivers/Select/Make Custom Set...*
- (3) Use the displayed dialog to enter a name for the set and press OK.

See also: [Driver List Window](#)

Sequencer Ports (Sound Checker)



Purpose:

This window is used to assign each MIDI channel playing in the sequence file to a particular MIDI port in the software. This dialog is opened from the Sequencer Window.

Directions:

To change the assigned port of a particular MIDI channel, click on the port selector beside the MIDI channel and choose a new port.

See Also: [Sequencer Window](#)

See Also: [Sound Checker](#)

Active (Group)

Purpose:

Active selects whether the selected data file is transmitted to the instrument as part of the Group or not. In certain instances, there may be parts of an instrument that you never change but are still part of a Group in this case, you can make that data file inactive in the Group so it is never sent to the instrument.

Directions:

To change the active state of a data file:

(1) choose *Group/Active*.

See Also: [Group Window](#)

Edit Mode (Group)

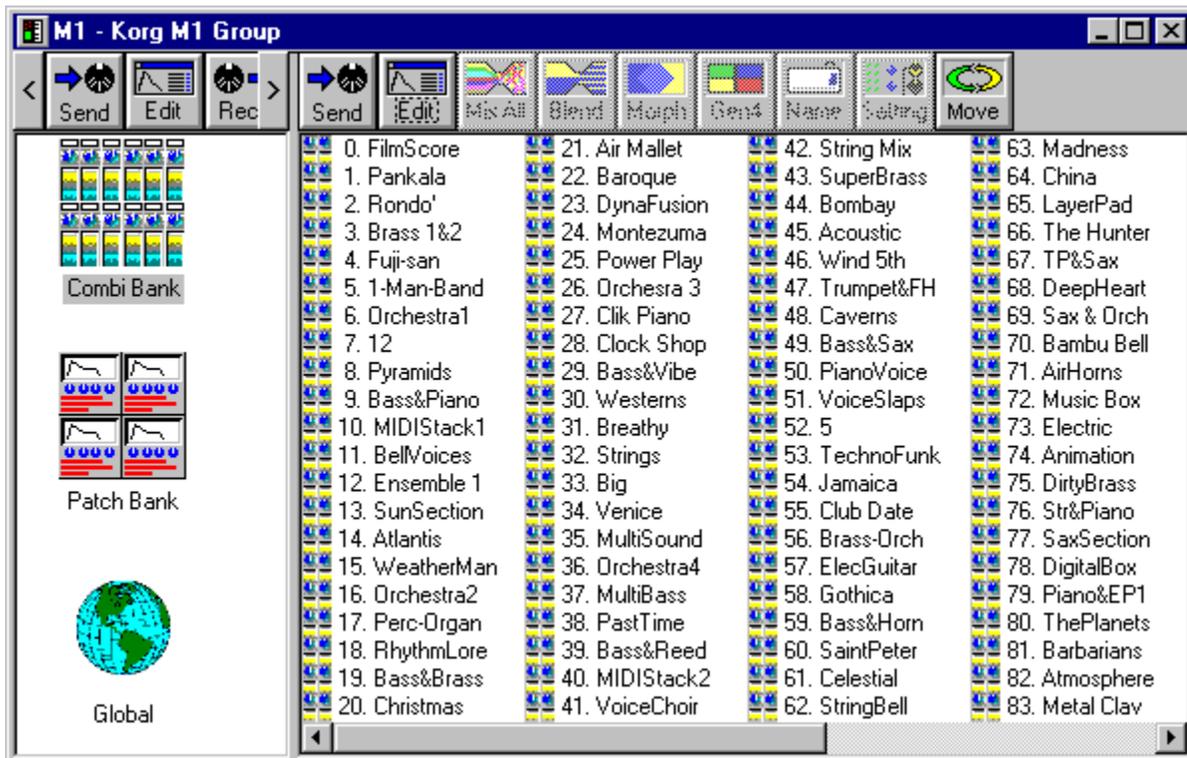


Purpose:

The Edit button controls whether the Group is displayed in split or regular mode. In regular mode, the Group window shows each of the different types of data in the Group and whether the data is active or not, as displayed below:



The second, and default mode for new Group is the split mode. In this mode, the Group members are displayed in the left hand portion of the window. Selecting a Group member displays its contents for editing in the right hand portion of the window. To edit the data, it must be either a Bank or a Patch with an editor. Following is an example of a Group in split mode.



Note: To access the menu for the Patch or Bank use the right mouse button speed menus. Disk Linked Patches can't be edited in the Library.

Directions:

To switch the Group Window between regular and split mode, press the Edit button.

Edit Mode (DBase)

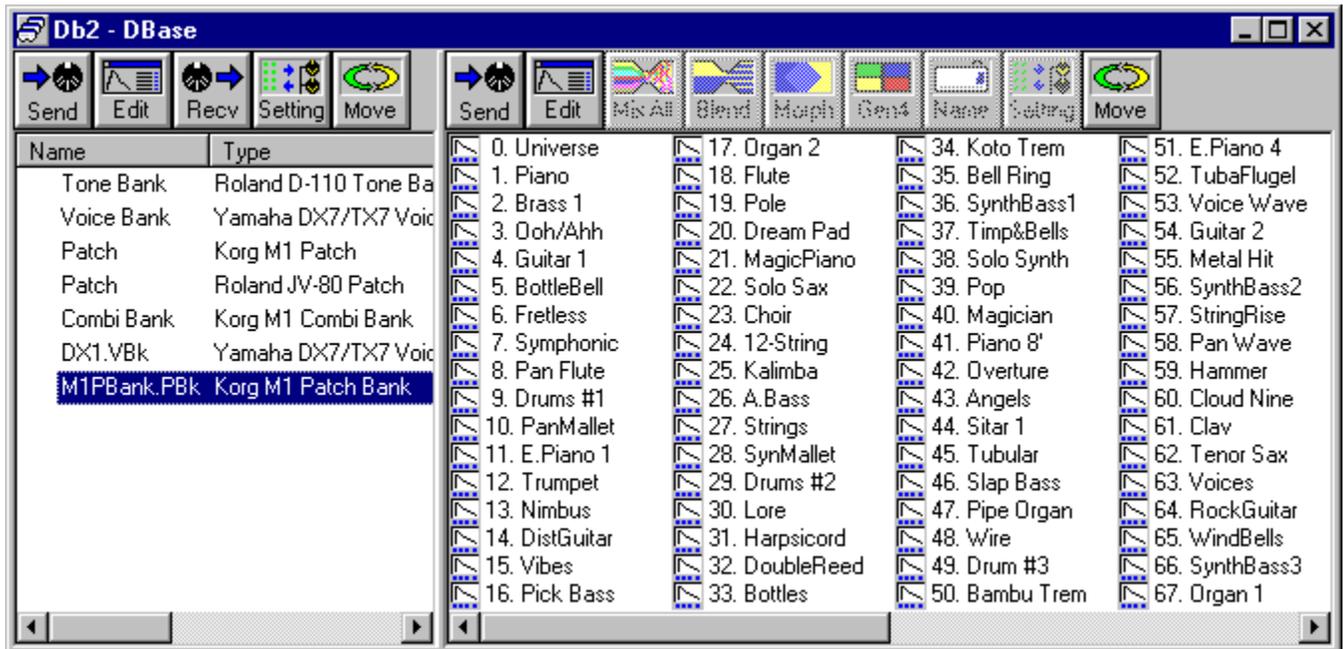


Purpose:

The Edit button controls whether the DBase is displayed in split or regular mode. In regular mode, the DBase window shows each of the different types of data in the DBase, as displayed below:



The second, and default mode for new DBase, is the split mode. In this mode, the DBase members are displayed in the left hand portion of the window. Selecting a DBase member displays its contents for editing in the right hand portion of the window. To edit the data, it must be either a Bank or a Patch with an editor. Following is an example of a DBase in split mode.



Note: To access the menu for the Patch or Bank use the right mouse button speed menus. Disk Linked Patches can't be edited in the Library.

Directions:

To switch the DBase Window between regular and split mode, press the Edit button.

Edit Mode (Library)

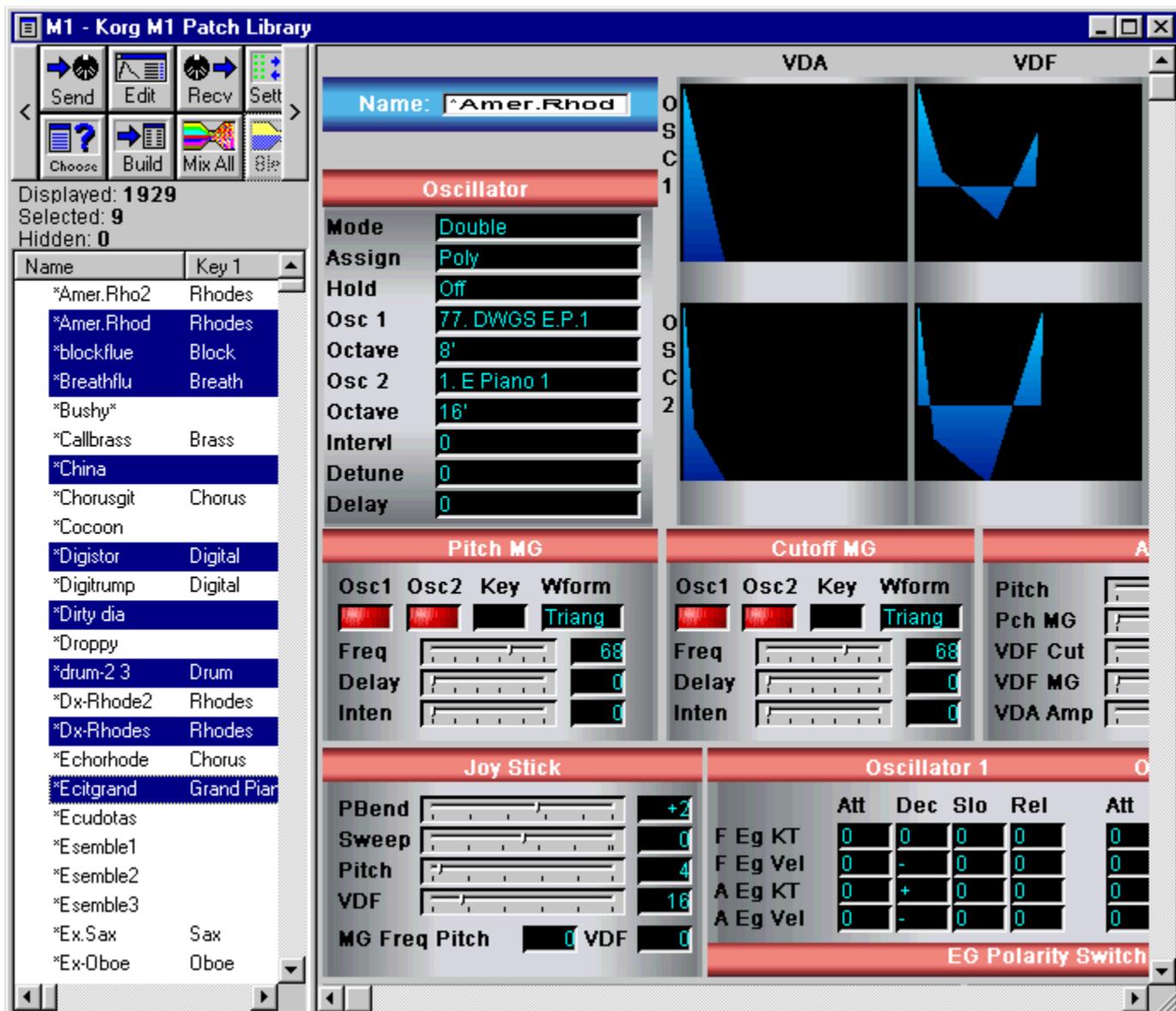


Purpose:

The Edit button controls whether the Library is displayed in split or regular mode. In regular (the default) mode, the Library window shows each of the Patches in the Library, as displayed below:



The second is the split mode. In this mode, the Library Patches are displayed in the left hand portion of the window. Selecting a Library Patch displays its contents for editing in the right hand portion of the window. To edit the data, there must be an editor available for it. Following is an example of a Library in split mode.



Note: To access the menu for the Patch editor use the right mouse button speed menus. Disk Linked Patches can't be edited in the Library.

Note: this feature is only available in Midi Quest and Solo Quest.

Directions:

To switch the Library Window between regular and split mode, press the Edit button.

Edit Mode (Bank)



Purpose:

The Edit button controls whether the Bank is displayed in split or regular mode. In regular (the default) mode, the Bank window shows each of the Patches in the Bank, as displayed below:



The second is the split mode. In this mode, the Bank Patches are displayed in the left hand portion of the window. Selecting a Bank Patch displays its contents for editing in the right hand portion of the window. To edit the data, there must be an editor available for it. Following is an example of a Bank in split mode.

OP	Frequency	L	Level	Scaling	R	Sensitivity & Output							
Mute	Coars	Fin	Mod	Det	Dpt	Crve	BkPt	Crve	Dpt	Scal	AMS	Sens	Output
1	3.00	0	R	0	0	-lin	A 4	-lin	0	7	0	5	
2	7.00	37	R	0	26	-lin	C 4	-lin	0	6	0	7	
3	1.00	0	R	0	0	-lin	G 3	+lin	0	5	0	3	
4	8.00	0	R	-1	0	+lin	C 3	-lin	30	4	0	3	
5	1.00	0	R	+1	30	+lin	A 2	-lin	24	4	0	5	
6	9.00	77	H2	+3	10	-lin	E 4	-lin	0	5	0	1	

Op1	99	96	94
	0	32	70
		13	0
		67	0
Op2	99	94	99
	0	69	23
		71	75
		78	0
Op3	99	80	99
	0	28	98
		11	0
		55	0

Note: To access the menu for the Patch editor use the right mouse button speed menus.

Note: this function is only available in Midi Quest and Solo Quest.

Directions:

To switch the Bank Window between regular and split mode, press the Edit button.

See Also: [Bank Editor Window](#)

Delete Hidden (Library)

Purpose:

Delete Hidden deletes any patches which are currently hidden (not displayed) in the Library.

Directions:

To permanently remove the currently hidden patches from the library:

(1) Choose *Library/View/Delete Hidden*

See also: [Library Window](#)

File Conversion Window (Utilities Menu)

No Header Conversion



Purpose:

The File Conversion Window (for no header files) is used to convert a No Header SysX data file into Sound Quest file format. If the files are in this format then the software is capable of creating a macro which can convert from the existing format into Sound Quest's format.

Directions:

The steps to converting a file are as follows:

1. Open the File Conversion Window.
2. Load the Sound Quest Driver that matches the type of data in the file.
3. Press the "No Header/Unisyn Macro" button to create the necessary macro.
4. Select "Convert File to SQ" or "Convert Directory to SQ".
5. Load the new data file to ensure proper conversion.
6. "Save" the new conversion macro to disk for later use.

NOTE: With an existing conversion macro replace 2 and 3 with 7.

7. "Open" an existing file conversion macro.

Step 1. Opening the File Conversion Window.

Use the following steps to access the File Conversion Window:

1. Bring down the "Utilities" menu.
2. Select "File Conversion Window" (item 8).

Step 2. Loading the Sound Quest Driver

The Sound Quest Driver must be loaded into the File Conversion Window because this information is an integral component of every Sound Quest Data File. It also tells the File Conversion Window what kind of data is being converted. A driver can be obtained from two locations, directly from disk or from the Driver List Window. This driver **MUST** represent the data stored in the file or the program will most likely crash while performing conversions!

Loading the Driver from disk:

1. Select on the File Conversion Window to activate it.
2. Display the Convert menu.
3. Select the "Load Driver From Disk" item.
4. Use the displayed File Selector to select the driver.
5. Select OK to load the driver.
6. The driver name will appear below "Convert to:".

Loading the Driver from the Driver List:

1. Display the Driver List Window (if necessary).
2. Select the Driver to load into File Conversion.
3. Select on the File Conversion Window to activate it.
4. Display the Convert menu.
5. Select the "Load Driver From List" option.
6. The selected driver will appear below "Convert to:".

Step 3 - Create the Macro

Press the "Create MIDIX Macro" button to create the macro that will convert your files into that used by Sound Quest. The program will do this automatically meaning that you will require no knowledge of the MIDIX file format or Sound Quest's. The one exception to this general rule is that of Roland. Because their instruments do not dump SysX in a consistent fashion, in virtually all cases, it is necessary for you to write a conversion macro.

After the macro appears, you may type in a comment so that you can remember the type of conversion that this file performs.

Step 4. Convert the File

To convert the file, select either "Convert File to SQ" or "Convert Dir to SQ". The program takes the following steps:

1. Opens a File Selector
2. Choose the file to convert and Select "Ok"
3. A data file is allocated for the Conversion Driver
4. The selected disk file is opened and prepared.
5. The macro is executed.
6. The Sound Quest file is saved to disk with the extension currently in the "Extension" box in the window.
7. If directory processing Loop to #3 for all files.

If you wish to convert only those files with a particular extension, enter this extension in the "Ext Select" entry area. For instance, if you only wished to convert files with the ".DDF" extension, you would enter "*.DDF".

WARNING: If a directory of files is being converted, ALL of the files in that directory MUST be of the same type.

WARNING: If there are two or more files with the same name but different extensions, all of the converted files will be lost except the last one converted. The files should be renamed before the conversion process is started.

Step 5 - Load the New File to Verify

After the files have been converted, the new files should be loaded to ensure that the process proceeded properly. This is particularly important when you have just written your own conversion macro.

Take the following steps:

1. Bring down the "Windows" menu.
2. Select the "Edit Window" item.
3. A File Selector will appear.
4. Select the newly created data file and press "Ok".
5. Verify that the data has been converted properly.
6. Transmit the data to your instrument.
7. Verify the data aurally.

Congratulations, you have successfully converted file formats.

Step 6 - Save the Conversion Macro

Once the conversion macro has been tested, you can save it to disk for later use. When the macro is saved to disk, the Driver file in use is also saved with the macro and is automatically loaded with the macro. Before saving the file if you wish, you may add a comment in the comment box so that you may easily keep track of what this conversion macro does in the future.

Take the following steps to save the macro:

1. Activate the File Conversion Window (click mouse on it).
2. Display the "Files" menu.
3. Select the "Save As..." item.
4. Enter a name for the macro.
5. Select "OK" to save the macro.

Step 7 - Load a Conversion Macro

If you want to use a conversion macro you have written and saved or received from someone else, follow these steps to load it:

1. Select on the File Conversion Window to activate it.
2. Display the "Files" menu.
3. Select the "Open" item.
4. Use the File Selector to select the macro.
5. Press "OK" to load the macro.
6. MIDI QUEST will load the selected macro for use.

Note 1: the conversion macro file does not actually save the Sound Quest Driver it is using, only a path to where that driver is located on disk. If you obtain a conversion macro from someone else, make sure that you either have the requisite driver yourself or you obtain the driver as well.

Note 2: when converting a directory of files with the 32 bit version of the software, the program will use only the 8.3 version of the file name. It will not use the long file name for the conversion. If any of the files use long file names, not only will you see a change of extension but also a change of name. If you wish to retain the full long name version, convert the files one at a time.

See also: [The Utilities Menu](#)

Floating Editor (Patch Editor)



Purpose:

Open Floating Editor is where you open your floating editors in order to do editing outside of the main program window. The function displays a dialog showing up to 30 available floating editors with a description or comment for each available editor. Double click on the floating editor you want and the program will open it.

[For extended information on Floating Editors, please click here.](#)

Directions:

- (1) Choose *Sngl Edit/Open/Floating Editor...*
- (2) Use the displayed dialog box to select the floating editor you wish to use
- (3) Use the displayed editor for your editing

See also: [Floating Editors](#)

See also: [Save Template as a Floating Editor](#)

See also: Properties [Patch Editor Tab](#) to set the floating editor refresh rate

See also: [Patch Editor](#)

File Conversion Window (Utilities Menu)

Converting SMF Files



Purpose:

The File Conversion Window (for SMF files) is used to convert SysX stored in SMF (Standard MIDI Format) files into Sound Quest file format. If the files are in this format then the software is capable of creating a macro which can convert from the SMF format into Sound Quest's format.

Directions:

The steps to converting a file are as follows:

1. Open the File Conversion Window.
2. Load the Sound Quest Driver that matches the type of data in the SMF file.
3. Select "Extract from SMF" or "Extract Dir of SMFs".
4. Load the new data file to ensure proper conversion.

Step 1. Opening the File Conversion Window.

Use the following steps to access the File Conversion Window:

1. Bring down the "Utilities" menu.
2. Select "File Conversion Window" (item 8).

Step 2. Loading the Sound Quest Driver

The Sound Quest Driver must be loaded into the File Conversion Window because this information is an integral component of every Sound Quest Data File. It also tells the File Conversion Window what kind of data is being converted. A driver can be obtained from two locations, directly from disk or from the Driver List Window. This driver **MUST** represent the data stored in the SMF file or the program will most likely crash while performing conversions!

Loading the Driver from disk:

1. Select on the File Conversion Window to activate it.
2. Display the Convert menu.
3. Select the "Load Driver From Disk" item.
4. Use the displayed File Selector to select the driver.
5. Select OK to load the driver.
6. The driver name will appear below "Convert to:".

Loading the Driver from the Driver List:

1. Display the Driver List Window (if necessary).
2. Select the Driver to load into File Conversion.
3. Select on the File Conversion Window to activate it.
4. Display the Convert menu.
5. Select the "Load Driver From List" option.
6. The selected driver will appear below "Convert to:".

Step 3. Convert the File(s)

To convert the file, select either "Extract from SMF" or "Extract Dir of SMFs". The program takes the following steps:

1. Opens a File Selector
2. Choose the file to convert and Select "Ok"
3. A data file is allocated for the Conversion Driver
4. The selected disk file is opened and prepared.
5. The SMF extraction process is executed.
6. The Sound Quest file is saved to disk with the extension currently in the "Extension" box in the window.
7. If directory processing Loop to #3 for all files.

If you wish to convert only those files with a particular extension, enter this extension in the "Ext Select" entry area. For instance, if you only wished to convert files with the ".DDF" extension, you would enter "*.DDF".

WARNING: If a directory of files is being converted, ALL of the files in that directory MUST be of the same type.

WARNING: If there are two or more files with the same name but different extensions, all of the converted files will be lost except the last one converted. The files should be renamed before the conversion process is started.

Step 4 - Load the New File to Verify

After the files have been converted, the new files should be loaded to ensure that the process proceeded properly. This is particularly important when you have just written your own conversion macro.

Take the following steps:

1. Bring down the "Windows" menu.
2. Select the "Edit Window" item.
3. A File Selector will appear.
4. Select the newly created data file and press "Ok".
5. Verify that the data has been converted properly.
6. Transmit the data to your instrument.
7. Verify the data aurally.

Congratulations, you have successfully converted file formats.

Note: when converting a directory of Standard MIDI files with the 32 bit version of the software, the program will use only the 8.3 version of the file name. It will not use the long file name for the conversion. If any of the files use long file names, not only will you see a change of extension but also a change of

name. If you wish to retain the full long name version, convert the files one at a time.

See also: [The Utilities Menu](#)

INI File Format (Technical)

This document explains how the .ini files are used in Sound Quest's editor/librarian products.

The .ini file has one block called [Instrument]. All of the settings can appear in any order although for consistency, you might want to maintain the same order as Sound Quest uses.

The first 5 parameters must always be:

Name - This is the name that appears on the Instrument bar in the program. It can be any name you wish. In this example, you are looking at the file for the Korg 01/W.

Dir - this is the name of the directory where the instrument's driver and editor files are located. This directory must be within the INSTR directory.

NumDrivers - This is the number of drivers used with the instrument

ResFile - This parameter holds the name of the file which contains the icon to be displayed in the instrument bar. Unless you know what you are doing, leave this set to "MQInsBmp.dll". If you have programming experience, you can create a .dll file that contains a .BMP icon. To match the other icons, the graphic should be 64 x 40 pixels.

ResNum - This parameter holds the Resource Number of the BMP icon in the file specified in ResFile. To use the default icon in Midi Quest or Solo Quest, set this parameter to 0.

ID - The ID parameters list the file names of each of the instrument's drivers. If there are 8 drivers then there should be IDs ID0 through ID7. The drivers will appear in the order listed. This allows you to choose the order of the instrument's drivers. You should attempt to order the drivers if the instrument is hierarchical. Multi/Combi/Performance type drivers should be at the top with the other drivers below. In the example below, the drivers are loaded in the following order: Combi.Bnk, Combi, Patch.Bnk, Patch, Drums, Global, Sequence, and AllData.

Ins - This parameter controls which drivers are automatically considered part of the group when the instrument is loaded. Again, if there are 8 drivers then Ins should be set for Ins0 through Ins7. To make a driver part of the group, set it to a value of 1. In the example below, the following drivers are automatically part of the group: Combi.Bnk, Patch.Bnk, Drums, and Global.

Auto - the Auto option is currently unused.

The following is a sample .ini file. In this case it is the 01/W ini file.

```
[Instrument]
Name=Korg 01/W
Dir=01W
NumDrivers=8
ResFile=MQInsBmp.dll
ResNum=0
```

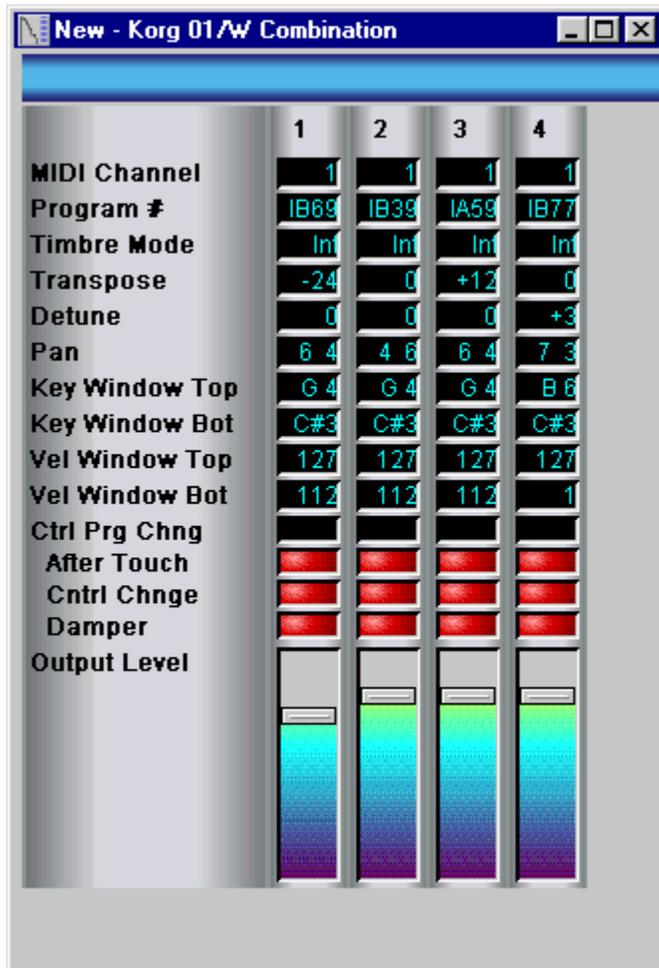
```
ID0=Combi.Bnk
ID1=Combi
```

ID2=Patch.Bnk
ID3=Patch
ID4=Drums
ID5=Global
ID6=Sequence
ID7=AllData

Ins0=1 ; if = 1 then is part of the Instrument
Ins1=0
Ins2=1
Ins3=0
Ins4=1
Ins5=1
Ins6=0
Ins7=0

Auto0=1
Auto1=0
Auto2=1

Floating Editors (Patch Editor)



Purpose:

Floating editors are a new innovation which allow you to more easily perform your editing while working with a sequencer. Floating editors provide two major features. First, the editors float, that is, they are not contained within the program's frame window and they stay on top of all of the other windows on the screen. The second is that they may contain only a portion of the editing parameters. This allows the creation of smaller, more effectively used editor windows.

One of the most obvious possibilities for this feature is to create a mini editor which contains only the most frequently used parameters. This editor could then be placed over your sequencer (or other application) so that you can easily make changes to a patch, combi, multi, or other data while working with the other application.

It is possible to have 30 different floating editors for each regular editor in Midi Quest or Solo Quest. These floating editors are simply created by making changes to the standard editor and then saving the modified editor as a floating editor. Each editor has a comment assigned to it for easy selection in the future while the program maintains the actual name of the editor file. As you use your floating editor, the program also maintains the last position and size of the floating editor so you can have it appear with an optimal location and size.

Creating a Floating Editor

The creation of a floating editor is a two part process. First the editor must be designed. This is accomplished by taking an existing editor and making modifications to it. Once the editor has only those components you require, it must be saved to disk as a floating editor.

To begin the process, choose *Options/Patch Editor Font/Small*. To modify an editor, it is necessary to open the editor with a small font in order for the correct spacing and sizing information to be generated when saving the file. This may result in a smaller font than you are used to, if you find it distracting, we suggest that you reduce your monitor's resolution from, for example, 1024x768 to 800x600. This will give you larger characters to work from and you can return to the resolution you are accustomed to when editing is finished.

Ensure that *Patch/Numeric Equivalents* is unchecked in the Options/Preferences dialog.

Now, open the editor that you would like to create a floating editor for. If you would like to create a floating editor which is identical to the full editor, all that is required is to save it. To do this, choose *Template Edit/Save Template as Floating Editor....* This will bring up a dialog box where you select which of the thirty locations will be used to store the editor and enter a description of the floating editor. In this case you might call it "Complete Editor" and press the OK button to save it.

Using a Floating Editor

To open this floating editor, choose *Open/Floating Editor...* and select the editor you just created. It will be opened in a small frame window that is separate from the main program but it is linked to the standard editor. If you close the standard editor, the floating editor will automatically close. Making changes in either window will automatically update the other windows. This updating is not immediate but is performed with the frequency set by the "Floating Editor Refresh" parameter in the Properties Patch Editor Tab. By default, updating occurs every 30 seconds.

Creating a Custom Floating Editor

To create a custom floating editor, open the editor you would like to modify. Let us assume that you want a Multi/Performance style editor which only displays the parameters for the first four instruments of a multi-timbral setup. All remaining controls are not necessary. To remove these parameters, select them by encircling as described in Selecting Parameters. Now, choose *Template Edit/Delete Selected Controls* to remove them. If you wish to delete the unnecessary controls in several groups that's fine.

If there any parameters which you want that are substantially separated from the rest, you might want to move them closer to the main group. This can be accomplished by choosing *Template Edit/Enable Control Drag* to check it. If you click in the upper left corner of a control, it can now be dragged to a new location.

It is also possible to move controls as a group. Select the parameters you wish to move and choose *Template Edit/Move Selected Controls....* Use the dialog to move the controls to a new position. You may, of course, perform any other editing you desire as well by change colors, borders, and text information in the editor. Once you have finished creating the custom editor, choose *Template Edit/Save Template as Floating Editor....* This will bring up a dialog box where you select which of the thirty locations will be used to store the editor and enter a description of the floating editor. In this case you might call it "My New Editor" and press the OK button to save it.

To open this new floating editor, choose *Open/Floating Editor...* and select the editor you just created. It

will be opened as a separate floating editor.

Enable Control Drag (Patch Editor)

Purpose:

When active, Enable Control Drag allows the active control to be moved around in the editor by clicking in the upper left corner, holding the mouse down and dragging to a new location. Any text or border settings that are attached to the control are carried as well. This function is part of the template editing tools which allow you to personalize your editors to match your working preferences.

Directions:

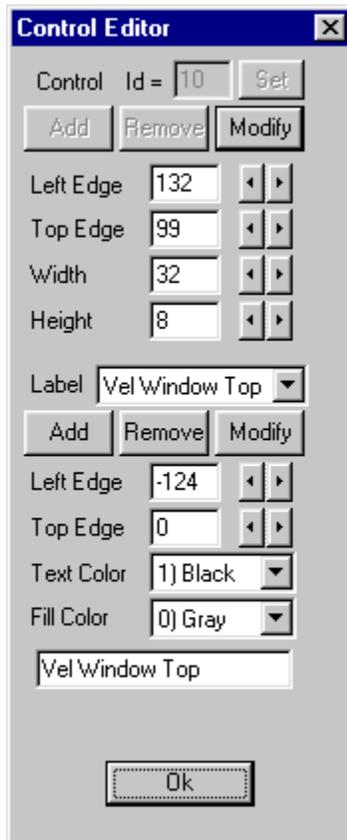
- (1) Choose *Sngl Edit/Template Edit/Enable Control Drag*
- (2) This will toggle the function on or off depending on its current state
- (3) If the option is checked, click in the upper left corner of a control and drag it to the new, preferred location

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties [Patch Editor Tab](#).

Note: not available in MQ Jr.

See also: [Patch Editor](#)

Control and Text (Patch Editor)



Purpose:

This dialog is used to change a control's basic size and position as well as modify the text assigned to it.

Editing controls are central to Midi Quest. There are currently 8 different ways in which you can edit data using Midi Quest. Numeric, Toggle, Slider, Level, Knob, Envelope, String, and Character String. Each of these controls allows important information to be presented in distinctly different ways. Keep this in mind as you begin modifying and developing your own templates.

Directions:

Control Parameters

Control parameters provide the functions necessary to add, remove and modify the editor controls, as well as position the controls in the editor and give each an ID.

Add

The "Add" function will add a control to the Template Window in the position and with the size specific by Left Edge, Top Edge, Width, and Height. After this function is selected, a dialog appears so that you can select the type of control to add. You may choose from any of the listed types. The program will then place the new control on the screen. In addition to manually setting the location of a control, it is possible

to double click on a location in the Template Window, the software will automatically use the location as the position for the new control. The control type dialog will appear and you can select the type to add.

Note: This function is not available in Midi Quest.

Remove

The "Remove" function will remove the currently selected control from the Template Window.

WARNING: This operation can not be undone. Once the control is deleted, it can not be retrieved.

Modify

The position and size of the currently selected control is modified to reflect the values currently displayed in Left Edge, Top Edge, Width, and Height as well as alter the current control ID number.

Control ID

At the top of the dialog is the ID number assigned to the control. This value shows the ID of the currently selected control and should be unique for each control. The IDs allow interaction between various controls in the editor. This value is automatically incremented as new controls are added to the Template Window. A new control number for any control can be entered by using the "Modify" function described above.

Left Edge, Top Edge, Width, Height

Left Edge - the x position of the upper left hand corner of the control
Top Edge - the y position of the upper left hand corner of the control
Width - the width of the control
Height - the height of the control

These four parameters describe the position and size of the currently selected control. To modify these, either click in the text area, enter a new value, and select "Modify" to update the control display, or use the directional arrows to fine tune the gadget position.

Label Parameters

The Label Parameters area contains the controls necessary to add, remove, and modify the text attached to each control, as well as select the text string currently displayed for editing.

Label

This drop down selector allows you to select the current label (text) that is being edited. If there no labels currently attached to a control, the drop down list will be empty.

Add

The "Add" function will add a label to the currently selected control in the position specified by Left Edge and Top Edge, with the colors specified in Text Color and Fill Color.

Remove

The "Remove" function will remove the currently selected label from the currently selected control.

WARNING: this operation can not be undone. Once the text is deleted, it can not be retrieved.

Modify

All of the parameters relating to the currently selected label of the currently selected control are updated to reflect the values in the Left Edge, Top Edge, Text Color, Fill Color, and Text.

Left Edge, Top Edge

These two parameters describe the position of the label relative to the control. To modify these, either click in the text area, enter a new value, and selected "Modify" to update the gadget display, or use the directional arrows to fine tune the text position.

NOTE: text positions are displayed relative to the position of the control. If a control is moved to a new location in the window, the label will also be moved.

Text Color / Fill Color

These two parameters control the text and background color of the text. Use the drop down selectors to choose the colors you would like to use.

Text

The final entry is used to edit the displayed text.

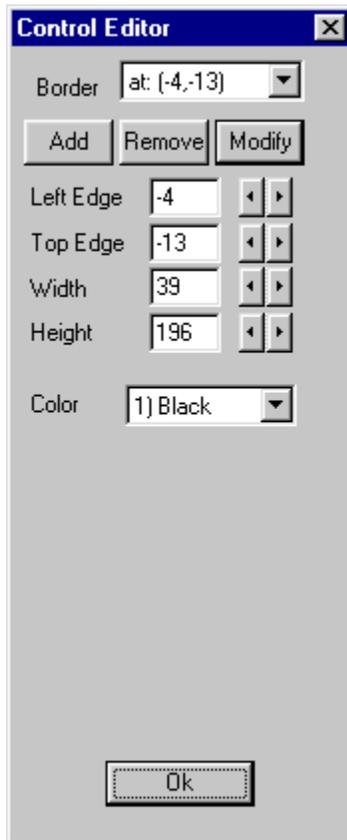
Note 1: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties [Patch Editor Tab](#).

Note 2: this component is drawn from Sound Quest's Tech Quest program although its functionality is significantly reduced

Note: not available in MQ Jr.

See also: [Patch Editor](#)

Border Editor (Patch Editor)



Purpose:

The Border Editor is used to add borders to individual controls. A border is a rectangle or string line (a rectangle of zero width or height) and is commonly found in editing templates which are column oriented. Any number of borders may be added to each control. The position of the border is ALWAYS relative to the control it is attached to, with the upper left corner of the control being coordinate (0,0). If a control with borders is moved within the Template Window, all borders associated with the control are also automatically moved.

Note 1: a control must be currently selected to add, modify, or remove borders.

Note 2: a border can not be added to a Slider control

Note 3: a border can not be filled with a pattern or solid color. Use the Xtras Control to add this type of display enhancement.

Directions:

Border

The Border drop down list is used to select the border which is currently being edited. Each border is identified by its Left Edge and Top Edge positions. The list will be empty if there are no borders attached to the currently selected control

Add

The "Add" function will add a border to the Template Window in the position and with the size specified by Left Edge, Top Edge, Width, and Height.

Remove

The "Remove" function will remove the currently selected border from the control.

WARNING: this operation can not be undone. Once the border is deleted, it can not be retrieved.

Modify

The position and size of the currently selected border is modified to reflect the values currently displayed in Left Edge, Top Edge, Width Height, and Color

Left Edge, Top Edge, Width, Height

There four parameters describe the position and size of the currently selected border. To modify these, click in the text area, enter a new value, and select "Modify" to update the border display or use the directional arrows to fine tune the border position

Color

The color of the border is the final parameter. Select the appropriate color from the drop down list. Please note that Midi Quest does its best to automatically create a 3D appearance so the top and left borders are drawn using the selected color. The right and bottom borders are automatically drawn in white.

Example

To add a border to a control:

1. Select a control in the Template Window
2. Select the Sngl Edit/Template Edit/Border/Border... in Midi Quest
3. Enter border's Left Edge, Top Edge, Width and Height
4. Select the color of the border
5. Select "Add" to add the border
6. The border will be drawn in the Template Window
7. To add a second border repeat steps 1 - 6 or
8. To modify the position, repeat 1 - 4 then press "Modify"

NOTE: Step 3 can be performed using the mouse (in Tech Quest only). Use the following steps to automatically enter the Left Edge, Top Edge, Width, and Height:

1. Activate the Template Window
2. Click the mouse button where the border is to start
3. Hold the mouse button down.
4. Drag the mouse to where the border is to end
5. Release the mouse button
6. The Left Edge, Top Edge, Width, and Height are updated.

HINT: When a new control is selected in the Template Window while the Border dialog is opened, the parameters in the Border dialog are automatically updated. However, if the select gadget does not have any borders, the parameters are not changed. This feature allows the same border to be added to a number of different controls quickly and easily. This is particularly common with performance style editors

which use columns of the same controlling parameters. To accomplish this follow these steps:

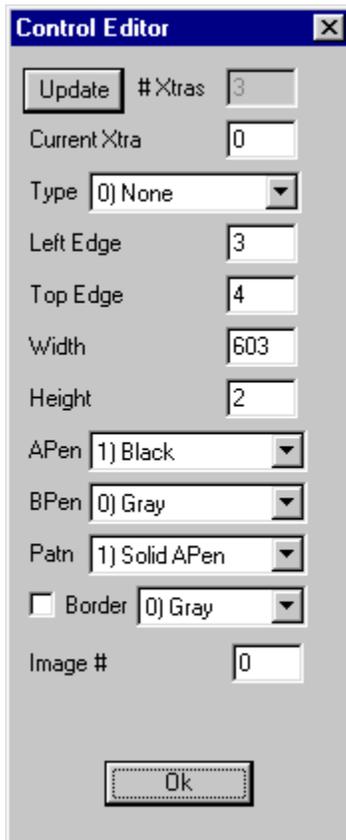
1. This feature is only available in Tech Quest.
2. Add the border to one of the controls.
3. Click on the next control
4. The border parameters will not change
5. Click on the Add button to add the parameters

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties Patch Editor Tab.

Note: not available in MQ Jr.

See also: Patch Editor

Xtra Editor (Patch Editor)



Purpose:

The Xtra editor is used to edit the template's Xtras. Xtras are additional graphic display elements which are used to improve the look of the edit template. Each of these elements is in a fixed location on the screen. They are not attached to a particular gadget.

Directions:

In Tech Quest, the position and size of the current Xtra can quickly be entered using the mouse. While the Xtra dialog is selected, click the mouse in the Template Window and drag a box on the window then release the mouse button. When the button is released, the Left Edge, Top Edge, Width, and Height parameters in the Xtra dialog are updated with the new values. Simply click the mouse in each of these parameters to "lock in" the values and press the Update button to update the visuals.

Note 1: While dragging the mouse in the Template Window, there is no visual indication the Xtra settings are being created (Tech Quest only)

Note 2: if the Xtra Type is set to 0, no Xtra is drawn

Note 3: the Xtras are drawn on the window in ascending order. If an Xtra is to appear on top of another Xtra, it must be higher in the Xtra list. For example, if the background Xtra is #2, then the foreground Xtra could be Xtra #3 and beyond.

Total Xtra

This parameter contains the total number of Xtras to be used. It can not be changed in Midi Quest so in Midi Quest it is possible to remove Xtras by setting the display type to 0 or to modify an Xtra but new Xtras can not be added.

Curr Xtra

This parameter displays the number of the Xtra currently being edited. The acceptable range is 0 -> (total number of Xtras - 1)

Type

The Type parameter indicates the type of Xtra being used. Valid entries are:

- 0 - inactive
- 1 - Unfilled Rectangle of color APen
- 2 - Unfilled Ellipse of color APen
- 3 - Filled Rectangle using all 2nd column parameters
- 4 - Filled Circle using all 2nd column parameters
- 5. Display the Image selected by 'Xtra Num'

Left Edge

Left Edge contains the x co-ordinate of the currently selected Xtra.

Top Edge

Top Edge contains the y co-ordinate of the currently selected Xtra.

Width

Width contains the width of the currently selected Xtra.

Height

Height contains the height of the currently selected Xtra.

Update

The "Update" button causes the Template Window to be redrawn to redraw any changes made to the xtras being edited.

APen

One of two color selectors for controlling the color of patterns in both filled rectangles and filled circles as well as the color used for ellipses and rectangles drawn in the Template Window.

BPen

The second of two color selectors for controlling the color of patterns in both filled rectangles and filled circles

Border Pen

Border Pen is used to filled rectangles and filled circles to add a border around the Xtra. Border Pen contains the color of the border to draw. This is now automatically given a 3D look.

Fill Patn

Filled rectangles and filled circles are filled using one of eight patterns as selectable in the drop down list

Xtra Num

Xtra Num contains the number of the image to display in the selected location (Left Edge, Top Edge). This number matches the image's position in the image list of the Image Control Window. This parameter is currently used solely to select which image the Xtra is to display. Images can only be added and removed in the AMIGA version of Tech Quest.

Xtra Tutorial (for Tech Quest)

1. In Tech Quest, click Xtra in the Control dialog
2. Enter '2' in 'Total Xtra' to select total number of Xtras
3. Enter '0' in 'Curr Xtra' to select the edited Xtra
4. Enter '3' in 'Type' to select a filled rectangle
5. Enter '1' in 'APen' to select a black on grey pattern
6. Enter '1' in 'Border Pen' to create a black border
7. Enter '3' in 'Fill Patn' for a high density black
8. Check the Border box to draw the border
9. Hold the mouse button down in template window
10. Drag the mouse to the lower right corner
11. Release the mouse button and the Xtra is drawn
12. Enter '1' in 'Curr Xtra' to select the second Xtra
13. Repeat steps 4 through 11
14. To fine tune the position of the second Xtra
15. Update Left Edge, Top Edge, Width, and Height
16. Press the Update button to redisplay the Xtras

Xtra Tutorial (for Midi Quest)

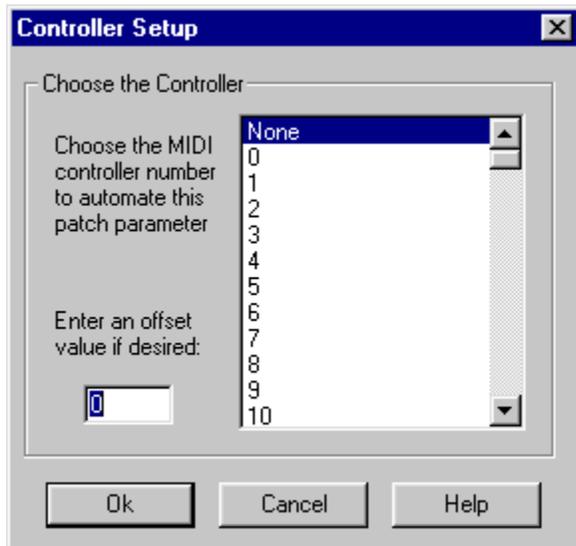
1. In Midi Quest, choose *Sngl Edit/Template Edit/Xtras...*
2. If the function is disabled, open a template with Xtras
3. Enter various values in 'Curr Xtra' to see the different Xtras
4. For an Xtra, change the Left Edge and Top Edge parameters
5. Press the Update button to see the changes in the window

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties Patch Editor Tab.

Note: not available in MQ Jr.

See also: Patch Editor

Controller Setup (Patch Editor)



Purpose:

Each control in an editor may have a MIDI controller number attached to it. If the program receives controller information at one of its open MIDI In ports on the MIDI channel assigned to this editor, the program will see if the controller number is assigned to any of the editor's controls. If there is a match, the program sets the value of that control based on the MIDI controller's current value. The software also sends out the equivalent MIDI parameter change message to the assigned port so that the instrument is automatically updated with the change.

You can think of this feature as "virtual NRPN" where you can have any MIDI controller editing one or more controls in an editor. Additionally, there are no limits on which parameters can be controlled as is the case with the internal NRPN implementation of most instruments.

This is also an important feature if you have a sequencer that does not store SysX data within its sequence tracks (such as Cakewalk). In these situations it is difficult to make real time changes to your instruments. With this feature, it's easy. Simply assign controller numbers to the parameters you wish to change in real time and have the editor open before running your sequence. Next, insert controller values in your piece create the SysX changes you wish to make. Finally, output your sequencer's controller track to a virtual MIDI port that MIDI QUEST or SOLO QUEST is receiving. The software will receive the controller values and output the correct SysX information to the instrument.

Also remember that more than one control can be assigned to a particular controller number so it is possible to have one controller making changes to a series of different editor controls.

Directions:

To assign a controller number to an editor control:

- (1) The editor must have been opened with the *Options/Patch Editor Font/Small Font* selected
- (2) Select the control in the editor
- (3) Choose *Sngl Edit/Template Edit/Controller Assign...*
- (4) In the dialog select the controller number to be used by the editor

(5) If desired, enter an offset value. The offset value can be positive or negative. For example, if you wish a controller value of 0 to set the editor control to a value of 2 then the offset should be 2

(6) Press the OK button to accept the settings

(7) To make the controller assignment permanent, choose *Sngl Edit/Template Edit/Save Template*

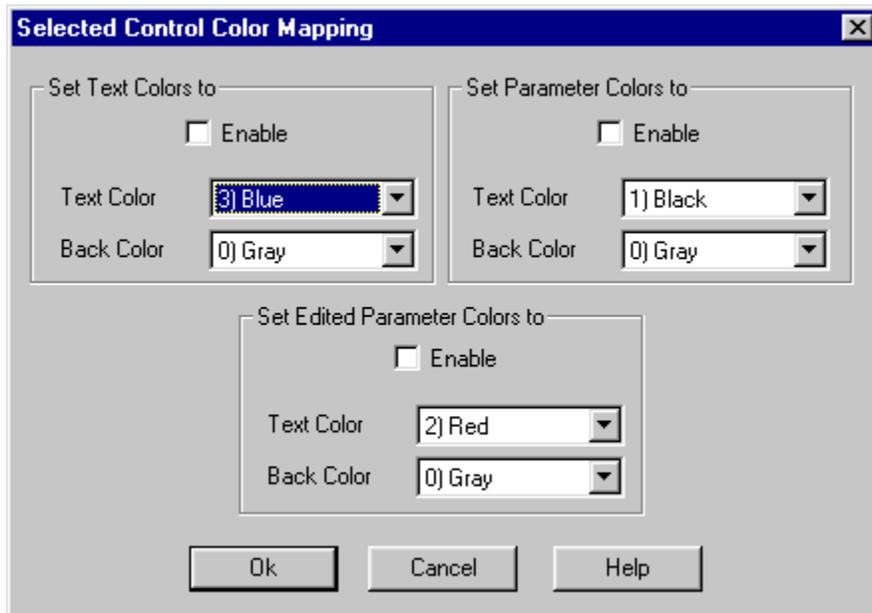
Note: After assigning the controller value, you should save the template using *Sngl Edit/Template Edit/Save Template* if you wish to keep this setting for future sessions with the program. The software saves these settings directly into the editor template. If you close the editor without saving your settings, the next time you use the editor, the controller settings will have to be made again.

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties Patch Editor Tab.

Note: not available in MQ Jr.

See also: Patch Editor

Assign Color to Selected (Patch Editor)



Purpose:

The Assign Color to Select dialog acts on all editor controls which are currently selected. [Press here](#) for control selection information. The dialog allows you to change the colors assigned to controls. There are three important and common colors for most controls: the color of text (labels) attached to a control, the color displayed by the control parameter while it has not been edited and the color used while it is in an edited state. These are the three colors set by this dialog.

Note 1: the names of the colors are based on the default colors used by Sound Quest. If you have used the *Options/Set Patch Editor Colors...* to change the default colors for the editors, the names of the colors displayed may not accurately reflect the colors appearing on the screen! In such conditions, you must be careful about your selections.

Note 2: When the dialog opens the colors currently selected are the defaults found in most Sound Quest editors for most controls. There may however, be some variation. For example, most labels in editors are Blue as indicated in the dialog.

Note 3: Even though knobs and levels do not display text, they still use the Text and Back colors to generate their displays and you can use this dialog to make changes to their colors.

Directions:

To change colors,

- (1) Select the controls in the editor which you wish to change the colors of
- (2) Choose *Sngl Edit/Template Edit/Assign Color To Selected...* to display the dialog
- (3) Check the "Enable" button for each of the color types you wish to change (text labels, unedited display information, and edited display information)
- (4) Select the new colors you wish to use for the controls

(5) Press the OK button

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties Patch Editor Tab.

Note: not available in MQ Jr.

See also: Patch Editor

Delete Selected Controls (Patch Editor)

Purpose:

This function allows you to remove controls which you do not require. This feature will most frequently be used when you are creating a specialized floating editor. In this case, it is likely you will not want all of the controls found within the editor but only a subset. In these situations, select the controls that are not required and delete them. Once you have removed the unwanted controls, you can save the modified editor as a floating editor. To regain the full editor, simply close the regular editor and reopen it.

Of course, if there are controls for parameters in an editor which you never plan on using and do not want to see them, they can be selected and deleted. In order to retain these changes, you must save the template after making the changes.

Directions:

- (1) Select the controls that are not required
- (2) Choose *Sngl Edit/Template Edit/Delete Selected Controls*
- (3) The selected controls are removed from the editor

WARNING: If your intention is to create a floating editor, be careful not to select Save Template after deleting controls since the standard editor will be overwritten. If this should happen, you can always reload the original editor template from your master disks in order to recover the full editor.

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties Patch Editor Tab.

Note: not available in MQ Jr.

See also: Patch Editor

Delete Unselected Controls (Patch Editor)

Purpose:

Delete Unselected Controls is similar to [Delete Selected Controls](#) except that the Unselected controls are removed instead. In certain instances, it may be easier to select the required controls and remove the remainder instead of the opposite. The function is used to remove controls from the editor, usually in preparation for creating a floating editor. For additional information, please see [Delete Selected Controls](#).

Directions:

- (1) Select the controls that are to remain in the editor
- (2) Choose *Sngl Edit/Template Edit/Delete UnSelected Controls*
- (3) Only the selected controls will remain in the editor

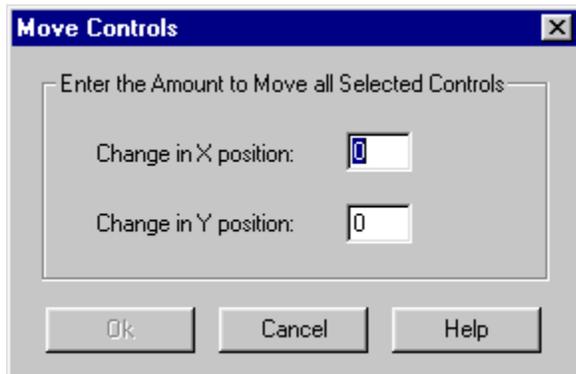
WARNING: If your intention is to create a floating editor, be careful not to select Save Template after deleting controls since the standard editor will be overwritten. If this should happen, you can always reload the original editor template from your master disks in order to recover the full editor.

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties [Patch Editor Tab](#).

Note: not available in MQ Jr.

See also: [Patch Editor](#)

Move Selected Controls (Patch Editor)



Purpose:

If you are performing a substantial amount of work on an editor including removing parts of an editor while creating a floating editor, you will probably need to move groups of controls from one location to another in order to create a well properly designed editor. This function allows you to move all of the selected controls (those circled in green) rather than having to move them one at a time.

Directions:

To move more than one control at a time:

- (1) Select the controls to move
- (2) Choose *Sngl Edit/Template Edit/Move Selected Controls...*
- (3) Enter the amount to move the controls in both the X and Y directions.
- (4) Use negative numbers to move the controls left
- (5) Press the OK button
- (6) Each of the selected controls will be moved

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties Patch Editor Tab.

Note: not available in MQ Jr.

See also: Patch Editor

Save Template (Patch Editor)

Purpose:

Save Template allows you to save a modified Patch Editor to disk as your main editor for a particular data type. You would commonly make use of this function after you have made changes to the editor. Examples of such changes are: changing the color of various controls in the editor, adding, modifying or removing decorations, adding MIDI Controller Numbers to various parameter controls, rearranging controls in an editor to better suit your working requirements, and finally, removing controls from an editor. Please remember that if you remove controls from an editor and then save the editor, these controls will be permanently lost unless you reload the original template from the master disks. Permanently removing controls is fine as long as you never intend to use them.

Directions:

To modify and save an editor:

- (1) Make all of the modifications to an editor that you wish using the various template editing functions
- (2) Choose *Sngl Edit/Template Edit/Save*
- (3) The program will overwrite your current editor with the modified editor

Warning! You should be VERY careful about the use of this function. This function automatically replaces the existing template for the given data type. If this is your intention, then continue. However, if you have removed controls from the editor, access to these parameters will be lost unless you reload the basic editors from your program's master disks.

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties Patch Editor Tab.

Note: not available in MQ Jr.

See also: Patch Editor

Save Template As... (Patch Editor)

Purpose:

Save Template As allows you to save a modified Patch Editor to disk under a different name. While this function has been provided, it is of little utility as saving the template under a different name will also require creation of a new driver or the modification of an existing driver in order to access the new editor template. Additionally, if you wish the program to use the new editor template as a floating editor, you must use *Sngl Edit/Template Edit/Save As Floating Editor...* in order for the program to be able to access it.

Directions:

To modify and save an editor:

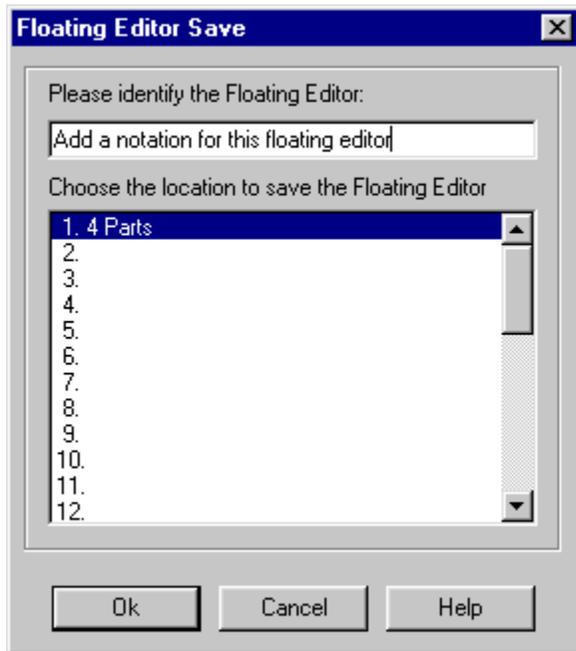
- (1) Make all of the modifications to an editor that you wish using the various template editing functions
- (2) Choose *Sngl Edit/Template Edit/Save As...*
- (3) A File Selector will appear so you can enter a new file name for the editor
- (4) Press "Save" to save the editor with new name

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties Patch Editor Tab.

Note: not available in MQ Jr.

See also: Patch Editor

Save As Floating Editor (Patch Editor)



Purpose:

Save As Floating Editor is used to save the existing editor or a new editor you have created so it can be used as a floating editor. Once saved, this new template will be made available as a floating editor by selecting Sngl Edit/Open/Floating Editor...

[For extended information on Floating Editors, please click here.](#)

Directions:

To create and save a floating editor:

- (1) Make all of the modifications to an editor that you wish using the various template editing functions
- (2) Choose *Sngl Edit/Template Edit/Save As Floating Editor...*
- (3) The dialog displayed above will appear
- (4) Choose from one of the 30 available floating editor locations to save the editor template
- (5) Enter a descriptive comment in "Please identify the floating Editor:"
- (6) If there is already a comment in the floating editor list, then you have already assigned an editor
- (7) When you save the new editor, you will lose access to the previous one
- (8) Press the OK button and you have saved a new floating editor

Note: When using this function, you will never see a file selector dialog. The program automatically takes care of naming and tracking the files so they can be loaded when requested.

Warning: These template files are automatically stored in the instrument's data directory under the MQDATA directory (eg. MQDATA\01W) and references to these files are stored in .INI files in the same directory. If you decide to delete the .INI files, you will not be able to access the floating editors even if the templates are available on the disk.

Note: this function is only available when the editor is opened with the Small Font option selected and Numeric Display not selected in the Properties [Patch Editor Tab](#).

Note: not available in MQ Jr.

See also: [Patch Editor](#)

Setup Columns (Library)

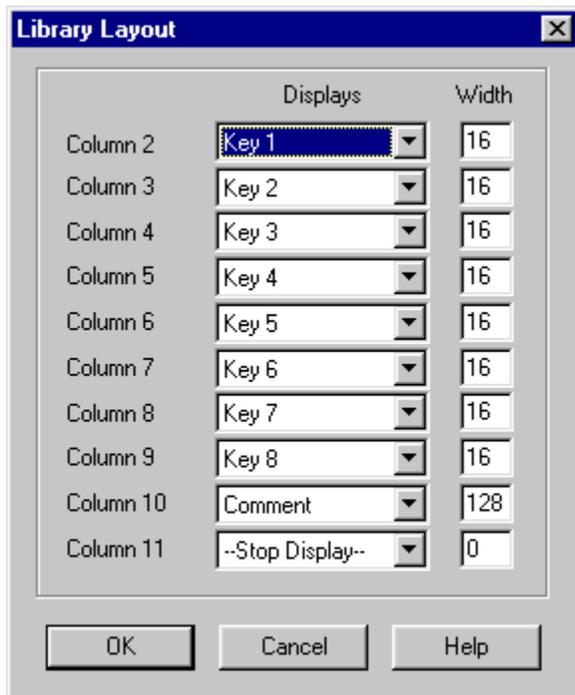
Purpose:

Setup Columns opens a dialog which allows you to set the width and order of each of the columns in the Library

Directions:

- (1) Choose *Library/View/Setup Columns...*
- (2) Make the column and column width assignments

Library Layout Dialog



The first column is always the patch name however, the other nine columns can be assigned to any other the Key 1, Key 2, Key 3, Key 4, Key 5, Key 6, Key 7, Key 8. and Comment. You can stop the column display at any point by choosing "Stop Display" and you can have two different display columns displaying the same information. Just use the drop down selectors to choose the items to display.

See Also: [Library Window](#)

Introduction

Registration

Before you do anything else, please take a moment to fill in and return your registration card. Please include on the card the serial number found on the front page of this manual. Returning this card will entitle you to the following services from Sound Quest:

- Free technical support
- Access to Sound Quest's web site for:
 - Technical support
 - New instrument support files
- Technical support via CompuServe
- Information on Midi Quest upgrades
- Information on new products from Sound Quest.

If this is an upgrade you have purchased, you will not find a serial number on the front page of the manual. Sound Quest will continue to use the serial number from the previous version of your product.

You can also register your software by:

- 1) Complete the registration form at Sound Quest's web site at <http://www.squest.com>
- 2) Send a message via Email to sales@squest.com including your name, address, serial number, and where you purchased the software and ask to be registered.
- 3) Faxing the warranty card to Sound Quest at our fax number

In this chapter, we will tell you a little bit about Midi Quest and how it works. We will cover the different formats Midi Quest stores data in and briefly cover the purpose of each type. Finally, we will provide you with information on accessing Sound Quest's web site to download new instrument files and obtain technical support.

Introduction

About Midi Quest

Midi Quest is an advanced development and organizational tool for the demanding musician. Use Midi Quest to edit, reorganize, and generate new sounds and System Exclusive information. Its comprehensive Data Base and Library facilities make storage and access to SysX information fast and easy. Midi Quest's extensive music performance features allow you to hear sounds as you develop them. You may also audition different sounds for any particular instrumental part in a musical sequence.

Midi Quest is designed to work with your instrument's System Exclusive data ("SysX"). SysX data is unique for every instrument. For instance, a DX7 does not understand the SysX from an M1, and vice versa. It used to be necessary to purchase an editor/librarian for each of your instruments since each program could only talk to one instrument. Midi Quest is different because it is modular and presently talks to over 300 instruments.

Midi Quest is comprised of multiple components. You know the main component as the program. The other components are called "drivers and templates" and have been specifically created for each instrument. There is a set of drivers and templates for each instrument.

What is a Driver?

A "driver" is a special control file which tells Midi Quest how to talk to an instrument and obtain a particular type of SysX data from it. For each type of SysX data that an instrument is capable of sending and receiving, Midi Quest has a corresponding driver which contains the necessary instructions to accomplish these tasks. For example, an M1 is capable of transmitting and receiving six different types of SysX data: Patch Bank, Patch, Combi Bank, Combi, Sequence and Global data. As a result, Midi Quest has six drivers for that instrument, one for each type of SysX data the M1 can send and receive. Each of the six drivers knows how to collect the appropriate SysX data from the instrument and return it later.

Please don't confuse Sound Quest's instrument specific drivers with the Windows MIDI drivers that come with your MIDI interface card. They are two completely different things.

What is a Template?

A "template" is a file which tells Midi Quest how to edit a particular piece of SysX data. There is a template for each type of SysX data that Midi Quest is capable of editing. Templates are linked to Midi Quest's drivers and are automatically read into the program whenever you want to edit a Patch. The template contains all of the information necessary to build a complete Patch Editor in the program. This includes codes defining each of the controls you will see in a Patch Editor. It also contains the commands that allow Midi Quest to automatically edit the instrument while you are making changes on the screen.

To extend our M1 example from above, the M1 will have three templates, one for the Patch editor, one for the Combi editor, and one for the Global editor. Since Midi Quest does not have a sequence editor, there is no template for that data.

Introduction

Working With SysX in Midi Quest

Midi Quest can store and organize your data in four different formats. Each format is designed to meet the various needs of the musician. These formats are: Data File, Group, Library, and DBase.

What is a Data File?

The "data file" is the most basic data stored by Midi Quest. A data file stores the SysX information collected by one of Midi Quest's drivers. The three other data formats are each built on the data file. The data file can be saved to disk, edited, and returned to the instrument. Examples of data files include the M1 Combi Bank, Roland D-10 Tone, or Yamaha DX7II System Setup.

A data file has two sub-types, Patches and Banks. A **"Patch"** for the purposes of Midi Quest is a single element in the instrument. This element may be a: sound, voice, multi-timbral setup, drum setup, sequence, or other data depending on your instrument's capabilities. For the purposes of this manual, each is considered to be a "Patch". So, to use Sound Quest terminology, your instrument may have: Sound Patches, Voice Patches, Multi-Timbral Patches, Drum Patches, Sequence Patches, etc. Also, remember that some instruments refer to their sounds as patches. In such cases, you have a "Patch Patch".

Note: if you are using a "Large Font" option with your graphics card, the appearance of Patch Editors in Midi/Solo Quest and the appearance of graphics in this help file may substantially improve if you select a "Small Font" option instead.

Below is an example of a Patch Editor for the Emu Proteus where the individual parameters of the SysX data can be changed. Each Patch Editor is custom designed for each type of data in the instrument:



A **"Bank"** is defined as a collection of two or more Patches which can be transferred to and from an instrument. As a result, a Bank's size is determined by the instrument's hardware design. For example, a Yamaha DX7 has 32 Voice Patches so it has a Bank size of 32. The following is an example of a Bank Editor for the DX7:



In a few instances, it is necessary to refer specifically to the type of Patch which is a tonal source for your instrument. Unfortunately, instrument manufacturers insist on calling this type of data by different names. For instance, a tonal source might be called a Sound, Voice, Patch, Tone, Preset, etc. Regardless of the name, the data still defines a tonal source. As a result of these inconsistencies and lack of industry standards, Sound Quest will refer to such a Patch as a "Sound Patch", even if it is called something else by the manufacturer.

Additionally, on most newer instruments there is a multi-timbral setup data structure where the different tonal sources are assigned to different MIDI channels. This structure, depending on the manufacturer, may be called a Performance, Multi, Combi, Global, Section, and so on. Regardless of the name, it still performs the function of creating a multi-timbral setup. In the few instances where it is necessary to distinguish between a general "Patch" and a multi-timbral setup we will refer to the multi-timbral setup as a "Multi Patch".

What is a Group?

A "Group", by default, holds a complete dump of an instrument's permanent memory area. To return to the M1 example, the instrument's permanent memory is made up of a Patch Bank, Combi Bank, Sequence, and Global data.

When a Group is loaded or a new Group created, it will hold each of these components.



Groups are extremely important because they link the individual data files of an instrument together and treat them as a whole. In a Group format, Midi Quest can provide more intelligent editing and organizing features than with individual data files.

For example, if you edit two M1 Combi Banks from two Groups and copy some of the Combis from one Bank to the other, Midi Quest will automatically copy the Combi's Patches as well. This is impossible with individual data files.

What is a DBase?



A DBase is a simple yet powerful data storage facility which allows you to take ANY combination of Sound Quest data files and store them together in one place, the DBase. The DBase can hold, for example, an M1 Patch Bank, an M1 Combi Bank, 2 DX7 Voice Banks, a D-10 system setup dump, and a Wavestation Performance. It can, in fact, hold virtually any combination of SysX data that you can think of.

The primary use for a DBase is to store snapshots of your MIDI system. A "snapshot" is a picture of the current state of one or more of your instruments and other devices in your MIDI setup. It contains all of the data necessary to restore the instrument(s) to their current state at some time in the future. After the snapshot is taken, the DBase should contain all of your instrument's banks along with any other data files which have been customized for a project. With this feature, it is possible to work on one project then take a snapshot of your instruments before going on to a different project. When you return to the first project, in the afternoon, the next day, or a year from now, you will be able to send the SysX data back to your instruments with confidence that all will be restored to the same state as when you last worked on your project.

What is a Library?



The Library is used to create a master listing of Patches from your instrument. These Patches may be Sound Patches, Multi Patches, or other types of Patches from the instrument. Each Library can hold only one type of Patch so you can have a Library of M1 Patches or a Library of DX7 Voices but you can not have a Library which holds both types. You need to create two Libraries, one for each.

If you have a large number of Patches, the best place to store and organize them is in a Library. One of the major advantages of the Library is that it can hold a virtually unlimited number of Patches. So, you can keep them all in one place. The Library becomes your one-stop source when you are looking for a Patch while working on a project. Because all of your Patches are kept in one place, it is much easier to find a Patch in a Library than it would be to hunt through a large number of Banks.

The Library's superior organizational services also enable you to search for and find various types of Patches quickly and easily. This allows you to try out various Patches for a project in about the same

amount of time it would take you to find a Patch if you kept them all in Banks.

In Review

The "Data File" is the basic block that Midi Quest is founded on. A Data File contains either a Patch or a Bank which can be edited in either a Patch Editor or Bank Editor, respectively. On any newer instrument which uses different types of data files, it is probably best not to save individual data files to disk unless you plan to keep track of which data files belong together.

The "Group" holds the contents of an instrument and should be used to save the current contents of an instrument.

The "DBase" is an extremely flexible storage facility which can hold any combination of Sound Quest data files. Its most frequent use is to hold a system

"snapshot". Use the DBase to record the contents of each of your instruments for a particular piece of music.

The "Library" is your resource center. You should have at least one Library for each of your instruments that holds your Sound Patches. Use your Sound Patch Library when you are searching for the right Patch to go with your new piece of music.

Please don't feel you are locked into this format of doing things. There are many Midi Quest users out there who have organized their data files in a completely different fashion. Also remember, it's easy to move data files between the Library, DBase, Group, Bank, and Patch so you can always reorganize later.

Introduction

Manual Organization

The manual is divided into eight basic chapters which are, the: Installation, Tutorial, Group Window, Bank Window, Patch Editor Window, DBase Window, Library Window, Driver List Window, Sound Checker, and finally, Common Elements chapters.

The **Installation Chapter** walks you through the installation of Midi Quest, various options for connecting your instruments to the computer, and finally, setting up the program so that it can talk to each of your instruments.

The **Tutorial Chapter** first takes you through the steps necessary to receive the contents of each of your instruments and save them to disk. This will ensure that Midi Quest is properly configured to talk to each of your instruments. It will also ensure that you have a backup of the data for each of your instruments. With this complete, you will not have to worry about inadvertently losing data from any of your instruments while you are learning the program. After saving the data, the Tutorial steps you through many of the most commonly used features of the program so you can begin to get a feel for how it works.

The **Group Chapter** discusses how to use the Group Window and the features within it.

The **Bank Editor Chapter** discusses how to use the Bank Editor Window and covers Bank organization features and Midi Quest's automated new Patch creation functions.

The **Patch Editor Chapter** covers the parameter editors which you will use to make new sounds, create multi-timbral configurations, and edit the various other parameters in your instrument.

The **DBase Chapter** discusses the DBase's major features.

The **Library Chapter** covers the Library's major features including how to best organize your Patches so that you can quickly and easily find the Patches you want in the future.

The **Sound Checker Chapter** details the tools within Midi Quest which are available to you for auditioning the Patches you have stored in Midi Quest or the Patch you are currently creating.

The final chapter is the **Common Elements Chapter**. This chapter covers features which apply to the whole program and not just one particular portion of it. This covers all of the global settings in Midi Quest along with the ancillary windows that do not directly relate to storing, organizing, and editing SysX data. These windows include the Driver Creator and File Conversion windows.

Manual Conventions

Throughout the manual, Sound Quest uses the following form to refer to menu selections: *menu/menu item*. As an example: *Files/Open....* This says you should go to the Files menu and choose the *Open...* menu item.

If the menu item has a sub menu, it is shown in the following form: *menu/menu item/sub item*. As an example: *Drivers/Open/Library*. This says you should go to the Drivers menu and display the Open sub menu. Choose Library within the sub-menu.

Preferences... (Options Menu)

Purpose:

Opens a tabbed dialog where program settings can be changed.

Directions:

To alter settings, click on the appropriate tab and make any desired changes. Press the OK button when complete.

Patch Edit Tab

Bank Edit Tab

Group Tab

DBase Tab

Library Tab

Driver List Tab

MIDI Tab

Tones Tab

Misc Tab

See also: Options Menu

Introduction

Getting Support

Online Help

Midi Quest comes with an extensive, context sensitive online Windows help file. This help file contains far more information about the program than the manual does. It should be the first resource you turn to for answers to your questions.

The help file contains general information on each of the windows in Midi Quest as well as specific information on each of the functions for the window. This information includes: why and when you would use the function, step-by-step instructions on how to use the function, and graphic examples.

If you would like general information on a specific window, activate that window in Midi Quest and press the Help "?" button on the icon bar. Midi Quest will open the help file and automatically go to the correct page.

If you would like specific information on a particular function in the menus or window icon bar, press SHIFT-F1 and then select the function with a mouse. Again, Midi Quest will automatically jump to the correct location in the help file so you can read about the selected function.

You can also open the help file by choosing Help/Contents. Nothing could be easier.

Via the Sound Quest Web Site

Sound Quest provides much of its technical support via its web site at <http://www.squest.com>. The web site contains the latest software support including new and updated instrument support files, FAQ (Frequently Asked Questions) pages, and program updates. If you should run into a problem, the first place to check is Sound Quest's web site as there may already be an update.

Via Email

Sound Quest provides technical support for its products via the Internet. Using the Internet, you can email any technical support questions to support@squest.com and we will email an answer back to you usually on the same or the next business day.

Via Telephone

Sound Quest provides technical support for its products via our tech support line at **(250) 478-9935**. In order to provide support we will want you to provide us with your product's serial number so please have it ready before you call.

Sound Quest's current technical support hours are 10:00am to 4:00 pm PST Monday through Thursday and 11:00am to 4:00pm PST on Friday.

If you have a question about Midi Quest which this manual does not answer, please feel free to call our tech support staff. However, please make sure that:

you have reviewed the relevant chapter(s) in the manual
you have checked the Midi Quest on-line help
the on-line help has information above and beyond the contents of the manual
you have checked Fast Tips help for the driver or editor
you have sent in your warranty card
you have your serial number and computer close by

Via Fax

Sound Quest will happily answer your tech support question via fax if you send it to **(250) 478-5838**. If you send a question via fax, please ensure that you include a return fax number along with your name and product serial number.

If you are faxing from an actual fax machine instead of a computer fax/modem card, do not lift the handset, dial the number and wait for the fax tone of Sound

Quest's fax machine. This will not work. Instead, insert the fax in the machine, dial the number and press the send button on the fax machine. Your fax machine and Sound Quest's will then connect properly.

Installation

The installation chapter takes you through the following steps:

1. Hardware requirements for running Midi Quest
2. Installing Midi Quest
3. Installing additional instruments
4. Verifying the installation
5. Running Midi Quest
6. Building a Driver List
7. Setting up the Driver List

Let's get started!

System Requirements

Midi Quest has the following system requirements:

- 286/386/486/Pentium or higher computer
- 4 Meg memory
- Microsoft Windows 3.1 or higher in standard or 386 enhanced mode
- MIDI interface with current Windows MIDI drivers
- Hard drive
- Windows compatible mouse

Installing Midi Quest / Solo Quest

Midi Quest is made up of a substantial number of different files including: program files, instrument support files (drivers and templates), and help files. As a result, the files on the disks are compressed. You must run Midi Quest's installer to properly install the program and create the necessary directories. Please follow these steps:

1. Insert the CD ROM in the CD ROM drive
2. From Program Manager, choose File/Run...
3. Enter drive:SETUP at the command line and press OK
4. A Welcome dialog is displayed, press NEXT.
5. An information dialog is displayed, enter your Name, Company and Serial #. Press NEXT.
6. Use the path dialog to select the program's installation location. Press NEXT.
7. The following dialog is used to select the type of installation. They are as follows:

7a. Typical

If this is your first installation, choose "Typical". It will take a little longer for the next window to be displayed because as the installer is calculating the file sizes for each instrument. After the next window is displayed, check the components to install, the main program files and help files are automatically selected. Click in the MIDDLE of the "Instrument Files" line and press the "Change" button to select the instruments you would like to install. To select an instrument, click in the LEFT hand side of the line for the desired instrument and a CHECK MARK will appear. Selected instruments will have a check mark beside them. Press NEXT to install the program.

Warning: with Solo Quest, you should only select one instrument. If you select

more than one, the installer will install and lock only the first selected instrument.

7b. Additional Instruments

This installation option is almost identical to the "Typical" option except that the Program files and Help files are not automatically selected for installation.

7c. New or Updated Program / Instrument files

Sound Quest provides much of its technical support via its web site at <http://www.squest.com>. The web site contains the latest software support including new and updated instrument support files, FAQ (Frequently Asked Questions) pages, and program updates. If you should run into a problem, the first place to check is Sound Quest's web site as there may already be an update.

Installing Additional Instruments

This section explains how to install additional instruments or an instrument support upgrade AFTER Midi Quest has already been installed. If you have just installed the program, please ignore this section and proceed. If you have just purchased a new instrument and want to add it to Midi Quest from your master disks, here are the steps you should follow:

1. Follow the steps above in "Installing Midi Quest / Solo Quest"
2. When you reach step 7, choose "Additional Instruments"
3. Highlight the Instruments option and press "Choose" to select the instrument(s) to install

Note: you can not add additional instruments with Solo Quest

Installation

Running Midi Quest

To run Midi Quest, double click on the Midi Quest icon in the Sound Quest Group in Windows' Program Manager. When the program is opened for the first time, you will see two dialogs, the first will request that you select the MIDI IN Ports that the program will use. Next, a dialog to select the MIDI OUT Ports is displayed. If you choose not to select the ports initially, they can be selected from within the program.

Choosing MIDI IN Ports

If you are running Midi Quest for the first time, you will be presented with the following dialog:



This is the MIDI IN Port dialog. It is used to open the MIDI IN ports that the program will use to receive MIDI data from your instruments. The dialog displays a list of the available MIDI IN ports that the program can use. You should open only those ports which are connected to instruments that you will be using the program with. To open a particular MIDI port, select it. Once you have selected each of the MIDI IN ports you will be using, press the OK button. The MIDI port selections can always be changed later from within the program.

Not all of the ports displayed will be for real MIDI IN hardware. You may also see ports which are created by programs to send specialized information to other programs such as MIDI Time Code. You should NOT select any of these MIDI IN ports. These virtual ports are often not designed or properly tested to work with SysX intensive programs such as Midi Quest. Using these ports can result in unusual error messages and program crashes which are beyond our control.

Win 3.1: If the MIDI IN Port list is empty, then most likely you have not installed any MIDI drivers in Windows. You may continue with setting up the program, however, you will not be able to talk to any of your instruments. As soon as possible, you should return to Windows' Program Manager. There you should open the Main Group and double click on the Control Panel. Once open, double Click on the Drivers icon. This will open a dialog where you can install the Windows drivers for your MIDI interface. If you do not see your MIDI interface listed and you have not received specific MIDI drivers for your MIDI interface, you should contact the MIDI interface manufacturer to obtain the proper drivers.

Win 95: If the MIDI IN Port list is empty, then most likely you have not installed any MIDI Drivers in Window. You may continue with setting up the program, however, you will not be able to talk to any of your instruments. As soon as possible, you should press the "Start" button and choose *Settings/Control Panel*. Double click on "Multimedia" and choose the "Advanced" tab. If you look under "Midi Devices and Instruments", you should see your MIDI interface drivers listed. If you don't, you will need to add the driver support for your MIDI interface. If you do not see your MIDI interface listed and you have not received specific MIDI drivers for your MIDI interface, you should contact the MIDI interface manufacturer to obtain the proper drivers.

Win 95: Another Windows 95 problem to watch for is the fact that some MIDI drivers will only work with MIDI programs that access the MIDI ports at the 16-bit level (this includes all sequencers whether they claim to be 16 bit or 32 bit programs). The 32 bit version of Midi Quest is completely 32 bit and these problem drivers will not work with it. If you see errors along the lines of "Error adding MIDI buffer" or "Error removing MIDI buffer" or the program can't open the MIDI port (and there are no other MIDI programs running). This is likely the problem. Updating your driver may work. The alternatives are to go with another MIDI interface that works with 32 bit programs or use the 16-bit version of the software.

Choosing MIDI OUT Ports

After selecting the MIDI IN ports to use, if you are running Midi Quest for the first time, you will be presented with the following dialog:



This is the MIDI OUT Port dialog. It is used to choose the MIDI OUT ports that the program will use to transmit MIDI data to your instruments. The dialog displays a list of the available MIDI OUT ports that the program can use. You should open only those ports which are connected to instruments you will be using with the program. To open a particular port, select it. Once you have selected each of the MIDI OUT ports you will be using, press the OK button. These port selections can always be changed later from within the program.

In general, you should not select MIDI OUT Ports which are only connected to an internal sound card (eg. the FM portion of a Sound Blaster), unless of course, Midi Quest has an editor for it.

If no MIDI OUT Ports are listed, most likely you have not installed any Windows MIDI drivers for your interface. Please follow the steps above to add MIDI driver support for your instrument.

Building a Driver List

After the MIDI IN and MIDI OUT ports are selected, the program will ask if you want to build a Driver List. Respond YES to the prompt. The program builds a Driver List based on the instruments you chose during program installation. After the list is built, the program will ask whether you wish to save the list. Reply YES. Midi Quest will now display your instruments in the Driver List. Remember that you can always add additional instrument support later.

There are no Instruments in the Driver List

If the Driver List Window is empty or incomplete, you have either: (1) not installed all your instruments during the installation procedure; or (2) not loaded all instrument drivers into Midi Quest. To determine whether you have installed your instruments, exit the program (Files/Quit) and view the contents of the INSTR directory of your disk. The directory should contain a list of installed instruments.

If there are no instruments or the list is incomplete, quit Midi Quest and rerun the installation procedure. Once the installation has finished:

- 1) Rerun the program
- 2) Click on the Driver List Window (to ensure it is active)
- 3) Choose Drivers/Build Driver List
- 4) When prompted to save the new driver list, select Yes.

There should now be a complete list of drivers in your Driver List Window.

Installation

Moving Midi Quest on your Hard Drive

If you wish to move the location of the program on your hard drive, you should remove the current version of the software and install it in the desired location. Simply moving the program on your hard drive will cause operating problems.

Downloading New Instruments from the Internet

Like the previous section, if you have just purchased the program you can ignore this section. In the future, when you want to add new instrument support to Midi Quest by downloading the files from Sound Quest's BBS, you should follow these steps.

If you are familiar with Midi Quest v4.0 or earlier, you will notice that the program's installer and file organization has changed substantially. To accommodate these changes and provide for the correct installation of new instruments in the future, it was necessary for Sound Quest to change the way new instruments are downloaded and installed into the program.

You must follow these steps when downloading new instrument files or you will likely experience difficulty either installing or using the new instrument files you have downloaded. Following are instructions for downloading and installing new instrument support files from Sound Quest's web site.

- 1) Connect to the internet
- 2) Connect to Sound Quest's web site at <http://www.squest.com>
- 3) Find the Downloads section for your computer
- 4) Select the instrument files you wish to download
- 5) Once downloaded, use PKZip or WinZip to expand each file
- 6) For each instrument, the Zip file will contain one ".Z" file
- 7) Download MQINSBMP.Z
- 8) Place these files on a floppy drive or a new directory on the hard drive
- 9) Run the installer program and follow the steps up to #7.
- 10) Choose option "C" in 7, install "New or Updated Program / Instrument files"
- 11) A Directory selector will appear, select the directory where the new files are located
- 12) Accept the directory and it will install the new files
- 13) Run the program and use *Drivers/Add Instrument* to add the new instrument(s)

Note: Some instrument files may also be available for download from Compuserve and in the future, the Internet. In these cases, you would download the files and follow steps 5 - 13.

Installation

Connecting Your Instruments

To receive data from your instrument, MIDI IN and MIDI OUT ports must be connected as follows:

Computer MIDI IN <-----> Instrument #1 MIDI OUT
Computer MIDI OUT <-----> Instrument #1 MIDI IN

(The computer's MIDI IN port must be connected to the instrument's MIDI OUT port. The computer's MIDI OUT port must be connected to the instrument's MIDI IN port.)

Those who plan to use Midi Quest with more than one instrument, but have no patch bay, must reconnect their MIDI cables each time data is requested from a different instrument. Those with a MIDI patch bay can easily set up Midi Quest to automatically connect the computer to the correct instrument.

Using a Patch Bay

The tutorial assumes you are not using a patch bay. If you are, we recommend that you connect the computer directly to an instrument until you are comfortable with Midi Quest.

Once you have your instruments communicating properly, you can connect your patch bay and set up Midi Quest to control it. In the Driver List window, set the PBay drop down menu to the MIDI channel that your patch bay "listens to" for patch change commands. This is the MIDI channel you send patch change commands from your sequencer to your patch bay in order for your patch bay to switch patches. After you have set the channel, you will see a "Cntrl#" appear in

each instrument's Instrument Bar. For each Instrument Bar, activate the Bar and press the Settings button. In the Settings dialog set the Cntrl# to the patch number that connects the MIDI OUT of the computer to the MIDI IN instrument and the MIDI OUT of the instrument to the MIDI IN or the computer. If you feel you need more information setting up your patch bay for use with Midi Quest, please see the online help.

JLCooper Patch Bay Warning: Due to software bugs within the MSB v1.0 patch bay, we do not recommend its use with Midi Quest. The Rev 2.0 patch bay does work properly.

Roland A-880 Patch Bay Warning: EARLY versions of the Roland A-880 patch bay have a bug which causes the A-880 to stop responding to patch change commands after SysX data has been transmitted through it. The current version of the A-880 works properly and ROM upgrades are available.

Installation

Setting Up Your Instruments

If you have correctly followed the steps in the Installation chapter and have connected the MIDI cables correctly, you are now ready to set up the control parameters in Midi Quest so that it can communicate with each of your instruments.



The program will be open with one window in the upper left corner of the screen. This window is called the "Driver List". The Driver List is used to obtain data from an instrument. The window has an icon bar along with a hierarchical list of

instruments. Each instrument has two components. The first component is the Instrument Bar. There is an Instrument Bar for each instrument. Each Instrument Bar contains an icon, the instrument's name, and a set of parameters: the Comm Channel, the MIDI channel, and the MIDI IN and MIDI OUT ports that the instrument is connected to. If you have set the program up to use a patch bay, the Cntrl # for the patch bay is also displayed.

Each of these parameters must be set correctly for your instrument or the program will not work properly! Work through the following setup routine for each Instrument Bar so the program can communicate properly with your instruments.

Preparing Your Instrument

To set up the first instrument in your list, click on the Instrument Bar for that instrument to highlight it.

Using the Fast Tips

The Fast Tips are a very important part of Midi Quest. These Help files provide specific information on the necessary steps required to setup your instrument for use with Midi Quest.



First, the Fast Tips will provide instrument specific information on how to enable System Exclusive reception on your instrument. Unfortunately, many instruments are actually shipped with their System Exclusive disabled. There is no method available for Midi Quest to automatically enable this setting for you, so you must do it for yourself. The Fast Tips will explain how.

Fast Tips will also explain how to find and change the channel that the instrument uses to transmit and receive SysX data. In Midi Quest, this is called the Comm Channel. Follow the instructions and make a note of the Comm channel.

The Fast Tips will also explain how to determine which MIDI channel the instrument is using as its basic channel if there is one. You should use this information to determine the channels being used by the instrument. Make a note of this channel as well.

Finally, the Fast Tips will provide other information, warnings, and limitations regarding your instrument. In some cases, the ROMs of particular instruments have bugs. In most cases, the Fast Tips will warn you of this. In such cases, you should get new ROMs from the instrument's manufacturer.

To open Fast Tips, click on the bar to activate the instrument then press the F. Tips button to display the

Fast Tips help for the instrument. A window similar to the following will appear:

Changing Instrument Settings

Once you have enabled SysX (on the instrument if necessary) and determined which channel the instrument is transmitting SysX on, you can proceed with setting up the instrument for use with Midi Quest.

Select the Instrument Bar for the instrument you wish to set up. Next, choose Drivers/Update Settings/Open Settings.... This will display the Settings dialog. This dialog is used to change each of the instrument's basic settings.



Each instrument in the Driver List has up to six basic settings which need to be set for proper program operation. These basic settings are: Comm(unication) Channel, MIDI Channel, Port In, Port Out, Patch # and Cntrl #.

Set the Communication Channel

The "Comm(unication) Ch(annel)" is the channel your instrument uses to communicate SysX data. By this time, you should have determined your instrument's Comm Channel with the help of the Fast Tips. Use the drop down menu beside Comm Ch to set the Comm Channel. There are a few older instruments which do not use this channel, in which case you can ignore this step.

Set the MIDI Channel

The "MIDI Ch(annel)" is NOT used for transferring SysX data between the instrument and computer. It is used for auditioning and developing sounds. In most instances, this parameter should be set to the instrument's basic MIDI channel. By this time, you should have determined your instrument's MIDI channel with the help of the Fast Tips. Use the drop down menu beside Midi Ch to set the Midi Channel.

For Roland's multi-part instruments, the MIDI channel used should be the same as for part #1 of the instrument.

Set the MIDI IN and MIDI OUT Ports

The MIDI In and MIDI Out drop down menus are used to select the MIDI In and Out ports that the instrument is connected to. The drop down menu displays all of the MIDI ports which are available to Midi Quest.

If you do not see the port to which the instrument is connected listed, close the Settings dialog and open the appropriate Midi In Ports or Midi Out Ports dialog from the Midi menu. Select the Port you require and close the port dialog. Return to the Settings dialog and choose the port.

Note: If the port still does not appear, it is likely that the port is in use by another program. You will need to shut down the other program and then reopen the port from within Midi Quest. Some drivers for MIDI interfaces, especially those from sound card manufacturers are not very sophisticated. These Midi interface drivers can only be opened by one application at a time. As a result, if the MIDI port is

being used by another application, you need to close the MIDI port there before opening it in Midi Quest.

Set the Cntrl #

The Cntrl # need only be set if you are using Midi Quest with a patch bay. In this case, the Cntrl # should be the same as the patch number in the patch bay that connects the MIDI OUT of the instrument to the

MIDI IN of the computer and the MIDI OUT of the computer to the MIDI IN of the instrument. Use the drop down menu beside Cntrl # to set the **Cntrl #** to the correct Patch number.

For Each Instrument

After you have entered the correct settings for each of the parameters in the Settings Dialog, press OK to accept the settings and use them for the instrument. After completing one instrument, select the next instrument and repeat the steps from Fast Tips onward to properly configure it. Repeat the steps until each Instrument Bar is properly configured to communicate with the instrument.

When you are finished configuring each instrument, choose *File/Save* to save the changes you have made.

You are now ready to proceed with the tutorial!

Tutorial

The tutorial will take you through a number of the more common procedures in Midi Quest. The intent is to get you comfortable with how the program operates along with some of the features you will use most often. After this, you can begin to explore the program more fully. The following is a basic outline of what we will be doing:

1. Load your instrument into a Group
2. Understanding the Midi I/O Window
3. Errors and Communications problems
4. Saving your first instrument Group
5. Saving each instrument Group
6. Loading a Group from disk
7. Editing a Bank in a Group
8. Auditioning Patches in a Bank
9. Editing a Patch from a Bank
10. Creating a new Library
11. Storing a complete Bank in a Library
12. Adding keywords in a Library
13. Searching a Library
14. Auditioning a Library
15. Creating a DBase
16. Creating a Custom Set

Load Your Instrument Into A Group

We will begin by loading the contents of one of your instruments into the program. This will not only give you your first view of how data is stored in Midi Quest, it will also allow you to verify that the Driver List has been properly configured to speak to your instruments.

For the first test, you should begin by choosing the Instrument Bar for a full synthesizer that Midi Quest has editors for. Try not to select a drum machine, effect box, patch bay, or other MIDI device. If you have a number of different brands of instruments to choose from, try choosing an instrument not made by Roland (if you are using any of Roland's D-series instruments). If you have only Roland products, choose the newest instrument you own. We are not anti-Roland here, however, Roland D-series products have substantially more bugs in their instrument ROMs than any other manufacturer and we would prefer that you get Midi Quest up and running as smoothly as possible.

Now that you have selected an instrument, double click on its icon in the Instrument Bar or click on the 'Edit' button and the Midi I/O Window will appear.

Understanding the Midi I/O Window

The Midi I/O Window will appear in the program whenever SysX data is requested from an instrument. This window shows which type of data is being requested, along with the Comm Channel, and MIDI IN and OUT ports that are being used for the data transfer.

If you have set all of the parameters in the Instrument Bar correctly using the Settings dialog and you have properly set up the instrument to receive and transmit SysX data, you will see the bar at the bottom of the window fill as data is received from the instrument. Congratulations, you have communication and can go directly to Saving Your First Instrument Group.

Errors and Communications Problems

If there is a problem communicating with the instrument, the Midi I/O Window will sit and flash the "Waiting" message. If you wait for more than 30 seconds, you can assume there is a problem somewhere in the system which needs to be corrected.

I'm not getting any SysX Data

Following are some of the more commonly encountered problems when first trying to receive data from an instrument. Work through each of the listed problems and follow the recommended suggestions. Retry the dump after each change you make,

- 1) If you have a patch bay, it may not be configured properly. For the moment, remove your patch bay from the loop and connect your instrument directly to the MIDI interface. Once you have the program talking with the instrument you can reattach the patch bay and configure it.
- 2) The MIDI cables between the computer and instrument are not connected properly. Ensure that the MIDI OUT cable from the instrument is connected to the MIDI IN of the computer and the MIDI OUT of the computer is connected to the MIDI IN of the instrument. Your reaction may be "of course they're connected" but over the years we've talked to many MIDI savvy people who have forgotten to connect the output of their rack instrument (or module) back to the computer.
- 3) The Comm Channel set in the Instrument Bar is not the channel that the instrument is currently listening to for SysX commands. Verify that the instrument's Comm Channel is the same as that set in the software. Read the instrument's Fast Tips to find your instrument's Comm Channel.
- 4) The instrument is currently ignoring SysX dump requests because SysX recognition has been disabled on the instrument. Read the instrument's Fast Tips to determine how to enable SysX reception, if necessary.
- 5) You are sending the dump request to the wrong MIDI port. Verify that the MIDI Out port selected on the Instrument Bar is the same port that the instrument is actually connected to.
- 6) Try the request again and while the Midi I/O window is waiting, go to the instrument and manually dump the data to the computer yourself. If SysX data is received by the program, you are still having a problem sending the dump request to the instrument. If the computer displays a "SysX Missing Bytes!" error message then the program did not receive exactly the SysX data it was expecting. The problem, however, is basically the same. You need to determine why the instrument is not receiving or acknowledging the dump request message from the computer. Recheck the Fast Tips..
- 7) If the program does not appear to receive any SysX data in step 6, cancel the Midi I/O window and go to the Utilities menu and choose Midi Monitor Window. Once the window is open, choose Monitor/Switch to Text Mode. Return to your keyboard and begin playing notes. You should see numbers displayed in the window which correspond to each note you play. If you see numbers, proceed to the next step, otherwise you have some sort of hardware/software problem and the program is not receiving any MIDI data. You should shut down Windows and power down your computer. Turn the computer back on, run windows, then run Midi Quest. Make sure that there are no other MIDI applications running. Next make sure the necessary MIDI port is open in the program by choosing Midi/Midi In Ports. Retry step 7 from the beginning. If the program still fails to receive data, try a different MIDI cable. If you continue to have problems, try working with a different instrument. If that fails as well, it is time to contact Sound Quest for tech support by phone, fax, or BBS.

I always get "SysX Missing Bytes" Errors

If you are receiving data from your instrument but are frequently receiving "SysX Missing Bytes" error messages the problem can be due to one of the following:

1) You are using a patch bay that is echoing the SysX dump request that is intended for the instrument back to the program. Midi Quest's receives its own dump request and interprets this as an error. The Midi I/O window will usually show that it has received a small number of bytes, usually less than 15. To fix this problem, you will need to alter the programming of the patch bay so that it does not route SysX coming from the computer back to the computer. Try bypassing the Patch Bay and connect the instrument directly to the computer's MIDI interface.

2) You have a MIDI instrument which is capable of echoing data that arrives at its MIDI IN port out its MIDI OUT port. An example of this is the Oberheim Matrix 1000. This problem is similar to that listed in problem 1 only this time it is the instrument which is echoing the dump request back to the program. To correct the problem, you will need to change the instrument so that it no longer echos MIDI events arriving at the MIDI IN port and out the MIDI OUT port.

3) If you own a Patch Bay, try connecting the instrument directly to the computer and bypass the Patch Bay. We have seen a substantial number of cases where a Patch Bay will loose part of the dump while it is being transferred. If this fails to fix the problem then continue.

4) Try the suggestions in the online help. Choose Help/Contents and pick MIDI Communication errors to obtain additional help.

The program crashes only on large SysX dumps

If you are successfully receiving smaller SysX dumps from an instrument but not large Bank dumps then the actual problem is an incompatibility between your MIDI hardware, MIDI driver software, and your computer hardware. You should make sure that you have the latest Windows MIDI drivers for your interface and if the driver allows you to set the size of the SysX buffer, increase this buffer size substantially, by as much as ten times. Additionally, there are a few other tricks you can try to get the program working.

1) Open the Midi/Midi In Ports dialog and change the SysX "Buffer Size" from 2000 bytes to 5000 bytes. Retry the dump.

2) If step 1 is not successful, return to the Midi/Midi In Ports dialog and substantially increase the "Buffer Size" to 50000. You should also reduce the number of buffers to 1 instead of 15. Retry the dump.

3) If step 2 fails, it is time to contact Sound Quest's tech support. From time to time manufacturers will change the format of their SysX dumps. It may be necessary to obtain new drivers for the instrument.

Tutorial

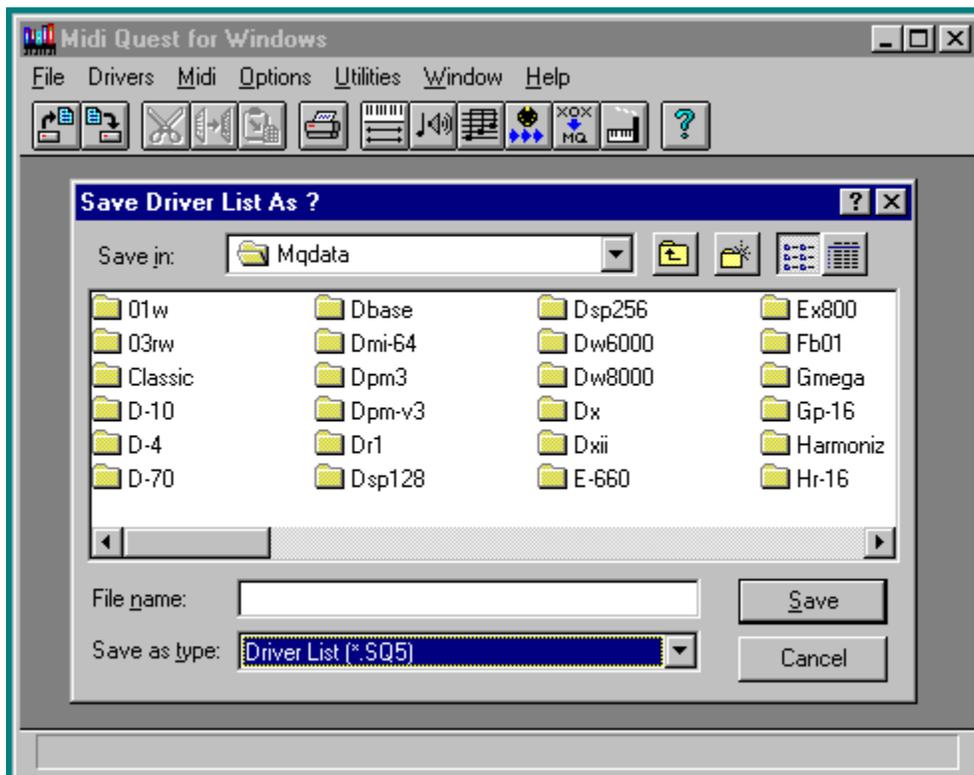
Saving Your First Instrument Group



If you have reached this portion of the tutorial, we will assume that you have successfully received a Group from your instrument. Congratulations.

You should have a window similar to the one displayed to the right.

To save this Group, you can either choose File/Save As... or click on the Save icon in the mini icon bar. A Save As dialog box is displayed so you can enter a name for the file. You might want to call the file ORIGINAL so that you will know that this file contains your original instrument settings. Notice that your File Selector has automatically opened in the instrument's data directory. If you put all of your files for this instrument in this directory, you will find it much easier to find your files in the future. After entering the file name, press the OK button to save the file.



You have successfully saved the contents of one of your instruments. You will now be able to restore your data at any time in the future. This is important should you make a mistake while learning Midi Quest and inadvertently lose part or all of the contents of the instrument.

You can now close this Group by double-clicking on the close icon in the upper left hand corner of the window.

Saving Each Instrument Group

Now that you have saved the Group for one instrument, you should work through each of the instruments

and save their Groups as well. The reason? You always want to have a backup of your instrument's contents should something unexpectedly happen to the instrument.

So please, return to the beginning of this chapter, choose your next instrument and save its Group. Once you have saved the contents of all of your instruments, we can proceed.

Loading a Group from Disk



Now that you have saved the internals of each of your instruments, let's go back to the Group loaded from the first instrument.

You have two options here, your first option is to open the file in the traditional way by choosing *File/Open...* A File Selector dialog is displayed. At this point, you should double-click on the instrument's data subdirectory to open it. In the subdirectory you will find the file you saved with an SQG extension. Double-click on the file name to load it.

The second option has the advantage of automatically placing you in the correct file directory when the File Selector is opened. First, select the Instrument Bar in the Driver List Window for the Group you wish to load. Now, go to the menus and choose *Drivers/Open/Group...* This will open a file selector and automatically place you in the instrument's data directory. The file you saved will have the SQG extension. Double-click on it to load the Group.

Editing a Bank in a Group



Now that you have loaded your Group, we will edit one of the Banks in the Group. For this demonstration, it would be best to choose the Bank containing the instrument's Sound Patches. This is often called the Patch Bank. Click on the Patch Bank line in the Group and a Bank Editor opens which contains the contents of the Group's Patch Bank.

Auditioning Patches in a Bank

Midi Quest lets you quickly listen to Patches in a Bank or Library, we call this auditioning. When you click on a Patch in a Bank or Library, the Patch is immediately sent to the instrument. Midi Quest then automatically plays a chord so you can hear the selected sound. Go ahead, try it.

If you would like to play a different chord, you can open open the Tones Window by choosing *Utilities/Tones*. From this window, you can define the number of notes to play in the chord, the chord's duration, the pitches, and the velocity of each pitch.

Swap a Patch in the Same Bank

One of the more common activities you will be involved in, while editing Banks, is organizing your Patches. This is accomplished by either pasting or swapping Patches within the same Bank or with another Bank.

In this example, we will swap two patches in the Bank you have opened to edit:

Click the mouse on the second Patch in the Bank and release the button
Click the mouse on the second Patch again and hold the button down to begin the drag
Drag the mouse so that it points to the Patch you wish to swap with

Release the mouse and the two Patches are swapped

Creating New Patches with Mix All

Mix All is one of five different Randomize functions in the Bank. Each of these Randomize functions processes Patches in a different way to quickly produce a Bank of new Patches. No editing knowledge is required. This is a fast way to create or try out new sounds in a piece of music. To create a Bank of new Patches using Mix All:

Hold down the CNTRL key and select two or more patches to mix
Choose the Mix All button or RMB Bank Edit/Randomize/Mix All
The cursor becomes a 'Wait' icon
When complete, a new Bank is displayed, like the one below



You can now try out the different patches you have created. Clicking once will send the Patch to the instrument and play a chord on the instrument so you can hear the new Patch. Note: to access menus in the Group Bank use the right mouse button.

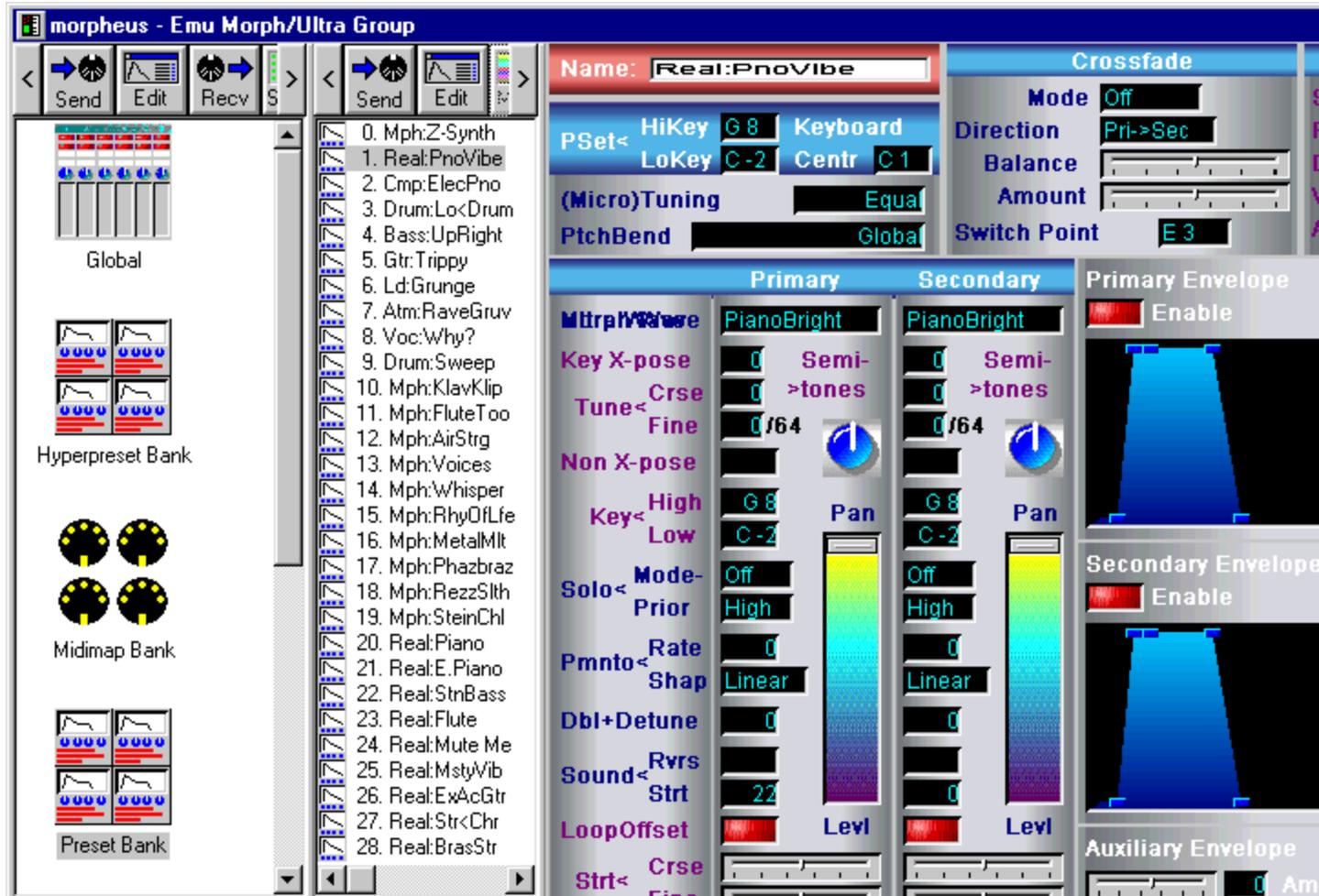
Swap Patches with Patches in Another Bank

Now that there are two Banks, we can swap Patches between them. To swap two patches between two Banks:

Select two Patches in the original Bank (hold down the CNTRL key)
Click the right mouse button and choose Bank Edit/Copy
Select two Patches to swap in the Mix All Bank
Choose Bank Edit/Swap with the right mouse button or regular menus

Tutorial

Editing a Patch from a Bank



Now, select a Patch from the original Bank that you would like to edit. Press the Edit button to open a Patch Editor for the Patch. The window shows an Emu Morpheus Editor. Each Editor will look different depending on the instrument and type of Patch you are working with.

Patch parameters can be edited in any one of the following three ways:

- Click on the parameter to edit and drag the mouse left/right
 - Click on the parameter to edit. Type in the desired value with your computer keyboard
 - Click on the parameter to edit and use the `<` and `>` keys to inc/dec
- Try changing a number of parameters in the editor.

When you are finished editing, click on the Edit button in the Bank Editor if you would like to close the Patch Editor Window. You can now make the changes to the Group permanent by pressing the Save button or choosing *File/Save Group*.

Tutorial

Using the Library

Creating a new Library

It's now time to start creating a Library for your instrument's Patches. Go to the Driver List Window and double-click on the text portion of the Instrument Bar. This will expand the bar so that each of the individual instrument drivers are displayed. Choose the individual driver that represents a sound on your instrument. This may be called a Patch, Tone, Sound, Voice, etc, depending on the instrument. We'll assume it's called a Patch. Now, press the New Lib button at the top of the Driver List Window to create a new, empty Library.

Placing an entire Bank in a Library

Return to the Group and select the Patch Bank; this is the Bank that holds the instrument's sounds. Click on the Edit button to display the bank if necessary. Now, click on any Patch in the Bank. Click again on the same Patch, hold the mouse button down, and drag the mouse towards the Library. You will see the mouse change to a Drag display. Drag the mouse over the Library Window and release the button. A message will be displayed asking whether you wish to move all of the Patches in the Bank or just the currently selected ones. Choose to move all of the Patches. Midi Quest will move all of the Patches from the Bank into the Library.

Adding Keywords in a Library



One of the major advantages of organizing your instrument's Patches in a Library is that each Patch can have descriptive keywords assigned to it. In the future you will be able to search for only those Patches which have matching keywords. For example, if you code all of your bass Patches with the keyword BASS, later when you are looking for a BASS sound, you can search for and display only your BASS patches. While it may take some time to initially set up each of your Patches, in the long run, the time will have been well spent. Now that you have a Library with some Patches, select the first Patch and press the Patch button in the Library's icon bar to display the Patch Update dialog to the right. The Patch Update dialog allows you to change the name of the Patch, assign a comment and add keywords to the Patch. To add a keyword:

Click the left mouse button in the raised area beside Keys "1."

A pop-up menu appears with a list of the keyword categories

Select a category and choose a keyword from within the category

When you release the mouse, the selected keyword is added to the keys. Even if the Patch you have selected is not a bass sound, pretend it is for this portion of the tutorial. Add the Bass keyword by clicking in the raised area beside "1." to display the pop-up menu. Move down the menu to "String-Pluck", highlight "Bass" and release the mouse button. Bass will be added as the keyword for this sound. Press the OK button to accept the changes. Choose two or three Patches and repeat this process so that you have several Patches coded with the keyword "Bass". Note: Up to eight keywords can be assigned to a Patch. Along with an indication of the type of Patch, keywords can be used to describe the type of sound such as "Slap" bass, or "Fender" bass. We strongly suggest that you must keep the descriptions of your sounds simple and consistent or you will not be able to effectively search for sounds in the future.

Searching a Library



The Choose dialog is used to select Patches from the Library based on the criteria you select. For the tutorial, we will show you how you can find Patches. The other most frequent use for the Choose dialog is to find duplicate sounds so you can expunge them. Imagine in the future, you have built up a Library of 5000 sounds and you would like to find all of the "bass" sounds in your Library. Follow these steps:

- Press the Choose button in the Library Window
- Check "Keys that match the Keywords"
- In Keyword Select, click in the first raised box
- A popup menu listing of keywords is displayed
- Move down to "String-Pluck", and highlight "Bass"
- Release the mouse button to select "Bass"
- Press the OK button in the Choose dialog

After the program has finished processing, only Patches with the keyword "Bass" are displayed in the Library.

Auditioning a Library

You can now audition each of the Bass Patches in your Library.

Similar to the Bank Editor, clicking on a Patch in the Library automatically sends that Patch to the instrument and plays a chord.

If you would like to play a different chord, you can open the Tones Window by choosing Utilities/Tones. From this window, you can define the number of notes to play in the chord, the chord's duration, the pitches, and the velocity of each pitch.

To redisplay all of the Patches in your Library, choose Library/View/Show All.

Saving the Library to Disk

To save your Library, choose File/Save As.... A File Selector dialog is displayed for you to enter a name for the new Library.

Tutorial

Creating a "Snapshot" DBase

As discussed in the Introduction, the DBase is most frequently used to create a "snapshot" of your MIDI system. In the future, this will allow you to restore your system to today's set up. To save your system "snapshot" follow these steps:

- Activate the Driver List Window
- Choose Drivers/Select/Group Banks and Patches
- This will select each Group driver for each of your MIDI instruments in order to create and store a MIDI system configuration
- Press the Driver List DBase button
- Midi Quest will proceed to load the SysX data for each selected driver into a new DBase Window when Midi Quest has finished receiving the SysX data
- choose *File/Save* to save your DBase

Providing that everything proceeded without a problem, you have now saved a system "snapshot". To send this snapshot from disk back to your instruments, simply choose Midi/DBase/Group to Instrument.

Note: Creating a snapshot DBase as described above will only work properly if you have enough MIDI ports so that each of your instruments can connect both its MIDI IN and MIDI OUT ports to a MIDI interface. If you have more than one instrument, you can accomplish this by using a multi-port MIDI interface or a patch bay.

Creating a Custom Set

If you find that you are frequently creating DBases using a particular combination of drivers, you can save this list of drivers as a Custom Set. Any time you need to create a new DBase with the drivers, select the Custom Set and it will automatically select the drivers for you. All you need to do is press the DBase button.

- Select the drivers for the data to load into a DBase
- Choose Drivers/Select/Make Custom Set and enter a name for the Set
- To use a Custom Set, choose Drivers/Select/Custom Set...
- Press the DBase button to create the DBase

Congratulations, you have reached the end of the Tutorial. It's time to explore.

Swap Hidden/Displayed (Library)

Purpose:

This function swaps the hidden and displayed patches in a Library. This function allows you to view the files hidden by a Choose operation and still keep the files separate. Suppose you performed a duplicates search on the library, the duplicate files would be displayed and all of the unique files would be hidden from view. To view the unique files you could use this function. The unique files would become visible and the duplicate files would be hidden. You could then have a look at the remaining Library before deciding whether to delete the duplicates or not.

Directions:

To swap the hidden patches with those currently displayed:

(1) Choose *Library/View/Swap Hidden / Displayed*.

See also: [Library Window](#)

Import Headerless Data (Driver List)

Purpose:

Open Import Headerless Data attempts to import SysX data currently stored in a disk file directly into a Patch or Bank Editor. To accomplish this, you must select the driver in the program which represents the type of data stored in the file. For example the Kawai K1 Patch Bank driver. Once selected, choose *Drivers/Open/Import (Headerless data)* to select the file to import and convert.

This function is useful only in very specific conditions. Essentially, you need to know that the contents of the file is the SysX data in the form it is send in from the instrument and that the header portion of the SysX dump is not saved as part of the file. This function is extremely "blind", it will accept whatever data file you give it and do the processing. If you do not provide a valide file you will gain first hand experience with the addage "garbage in, garbage out". That is exactly what you will get.

Now that you have been warned, here are some examples of places where this function will work. This function will be able to import many UniSyn and XOr patch and bank files as these are stored as described, data without any SysX header information. You should be able to import data saved for instruments such as the DX7, M1, K1, K4, and so on. The best way to test this function is to know the contents of the file and import it. It will be immediately obvious by looking at the contents of the imported Patch or Bank whether the function was successful or not.

Note: this function will not work for all files. For the importation to work, the file must contain only the type of data you are trying to load. For example, if you are trying to import a Kawai K1 Patch Bank but the file contains data other than a Patch Bank, the data is not imported correctly. In these situations, you will need to use the File Conversion Window to write a macro to perform the conversion.

Directions:

To read a file and open a Patch or Bank editor for it:

- (1) Select the driver in the program which corresponds to the data in the file
- (2) Choose *Drivers/Open/Import (Headerless Data/UniSyn/XOr)* and use the file selector to choose the file to convert
- (3) An editor window will open containing the imported data

See Also: [File Conversion Window](#)

See Also: [Driver List Window](#)

Add to Cakewalk Toolbar... (Utilities Menu)

Purpose:

This option is used to attach the program to Cakewalk's Toolbar so that the software may be quickly opened from Cakewalk. This function will not be enabled if the program does not find a registered version of Cakewalk in the computer.

Directions:

(1) Choose *Utilities/Add to Cakewalk Toolbar...*

See also: [The Utilities Menu](#)

Uninstall Preparation... (Utilities Menu)

Purpose:

This function presents a dialog which is used to remove elements of the program which the included uninstaller is or may not be able to. There are two types of information which the Uninstaller Preparation dialog can remove.

First are additional instrument files which were added after the initial program installation. If you have added instrument files after the initial program installation, the Uninstaller program is unaware that these have been added and will not delete them. Checking "Instrument Files" will remove the selected files.

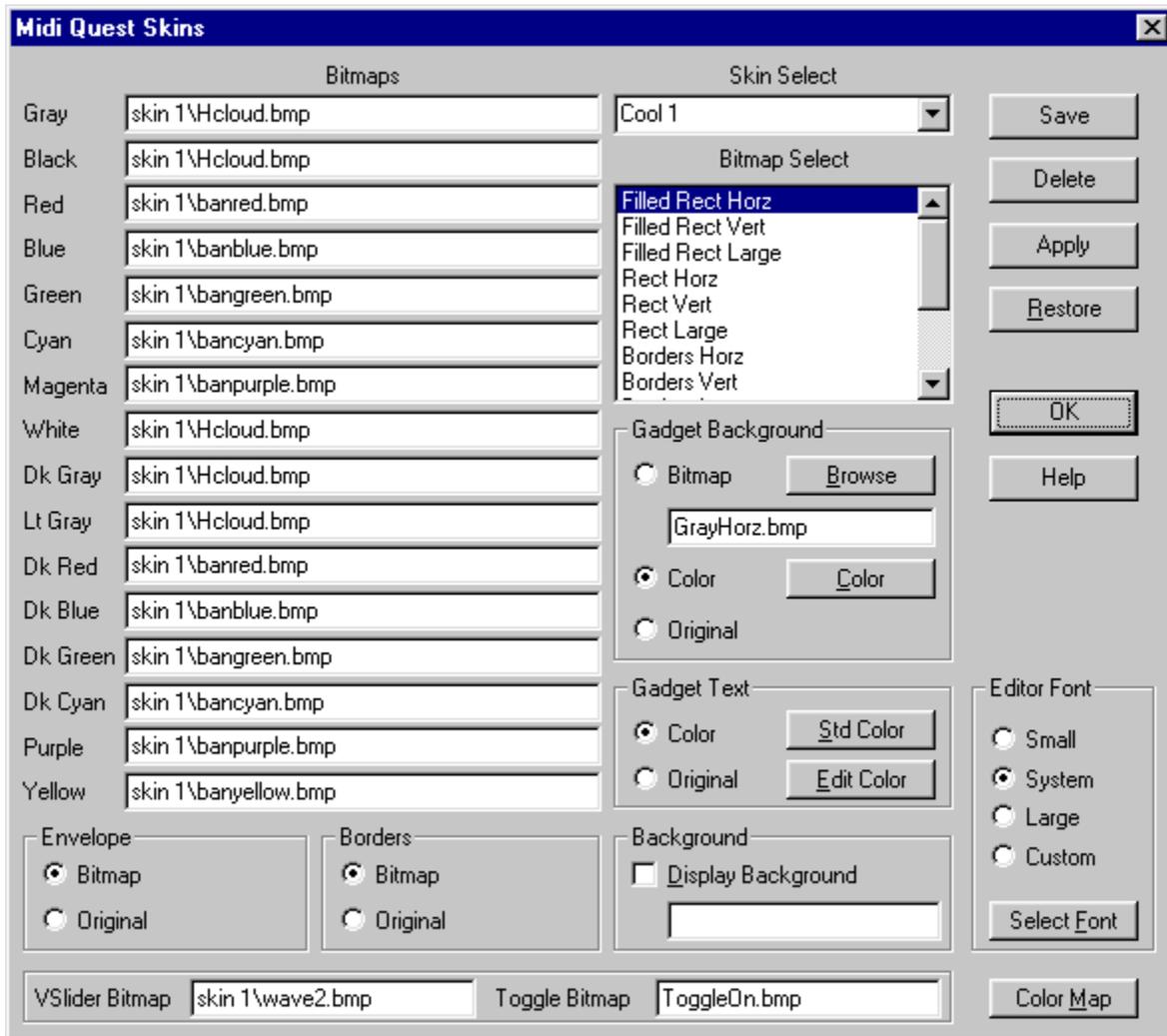
The second set of information the Uninstaller can not remove are all of the Registry assignments for individual data file types. These registry entries allow Explorer to show program data files with the correct icons and to open files into the program by double clicking on them. This option will only be active for 32 bit applications.

Directions:

- (1) Choose *Utilities/Uninstall Preparation...*
- (2) Check the data to delete from the hard drive.

See also: [The Utilities Menu](#)

Skins (Common Elements for Patch Editor)



Purpose:

Skins gives you a substantial amount of control over how the editors in Midi Quest are displayed without having to work in Tech Quest, Midi Quest's template editor.

So how do the skins work? Essentially, this dialog allows you to substitute your own bitmap graphics for existing graphic objects in the patch editors. You can also specify your own font and color map.

Please remember there can only be one Skin active at a time and it is applied to all open editors. If you want to give specific characteristics to a specific editor, you will need to use Tech Quest to accomplish your task.

Skin Select

Skin Select is a drop-down list box which displays a list of all of the available skins. To select a skin, simply pick it from the list. This will automatically fill in all of the parameters in the dialog to match the

settings of the newly selected skin.

To set the newly selected skin as the default skin to be used by the patch editors, click the Apply button. Any patch editors that are currently open are automatically reconfigured to use the new settings.

Bitmaps and Bitmap Select

The settings that will have the greatest effect over the display of the editors are the bitmap assignments. In total, you can assign 208 different bitmaps however, a small number of these will have a substantial effect on the display of the editors while many will have almost no effect.

So how does bitmap substitution work? In Tech Quest there are 3 different graphic structures which you can add to patch editors to enhance their look. In order of frequency of use they are: Borders, filled rectangles, and unfilled rectangles. Border are unfilled rectangles which are attached to a specific parameter and move when that parameter is moved. Both forms of rectangles are placed in an absolute position in the editor.

Here, we have divided these structures into horizontal, vertical and large options. This allows you to define a different bitmap for long, thin vertical structures vs long thin horizontal ones.

In addition to the frames, you can also specify bitmaps that are used for envelope backgrounds, horizontal levers, vertical levels, and knobs. This adds up to 13 categories.

Midi Quest allows 16 different colors to be defined and used so you can have a black border, a blue border, a red border and so on. So 13 categories with 16 different colors means 208 bitmaps in total.

Lets do a quick example:

1. This example will work with many of the Multi/Combi/Performance style editors in Midi Quest. If you want to see one that is guaranteed to work, open (install if necessary) the 01/W Combi editor.
2. Open a new editor from the Driver List Window
3. Open Explorer and find a .bmp file that you would like to have used as the background for each of the 8 vertical parts in the editor.
4. In the Midi Quest Skins dialog, under "Bitmap Select" click on "Borders Vert"
5. In explorer, click on the .bmp file and drag it so that the mouse is over the text are beside "Black" in the "Bitmaps" column.
6. Release the mouse and you should see the path and file name of your file appear in the text area. If it does, great, you're done. If it doesn't, you'll need to retry.
7. Press the "Apply" button and have a look at the Combi editor. you should see that your bitmap has replaced the existing one.
8. There you go, its that easy!

Using the example, you can now substitute any bitmap you wish for a specific combination of graphic structure (eg Borders Vert) and color (eg. Black). You can either type in the file name yourself or drag and drop the file from Explorer. To save the new skin you have created, type in a name for the skin under "Skin Select" and press the "Save" button.

Now, you may say, 208 graphics is a lot of graphics, and it is. But as previously mentioned, many of these combinations are never used so you can leave them blank. Sound Quest made the decision to offer the maximum amount of flexibility when designing the dialog.

While getting started, keep in mind that the following bitmaps will have the most effect on the visual display of most editors. Once you get the hang of working with these, you can expand to the less commonly used ones where necessary:

Bitmap Select: BordersVert Bitmap Color: Black
Bitmap Select: Borders Large Bitmap Color: Black
Bitmap Select: Filled Rect Horz Bitmap Color: all colors, notably: Red, Blue, Green, Cyan and Magenta

Again, you must remember that the skins are intended to allow you to change the overall feel and appearance of the editors. If you want absolute control, you will need to use Tech Quest to place bitmaps in the particular locations you would like them.

Gadget Background

For editable parameters that use the standard display configuration in Tech Quest (regular color = black, edited color = red), it is possible to define the background of the parameter.

If you select "Bitmap" then you should enter the name of the bitmap you wish to use as the background for the parameter gadget. You can either type the file name in directly or use the "Browse" button to make the selection.

If you select "Color", press the "Color" button to use the color dialog to select a color to use as a background.

If you select "Original" then the original v7 display style is used

You will find that the default skin that ships with Midi Quest v8 specifies a background color of black. This setting will affect the display of number, text, and envelope parameter gadgets.

Gadget Text

For editable parameters that use the standard display configuration in Tech Quest (regular color = black, edited color = red), it is possible to define the standard and edited colors.

If you select "Color", press the "Std Color" and "Edit Color" buttons to select the colors which will be used to display parameters in their edited and unedited states.

If you select "Original" then the original v7 display style is used (black on white / red on white)

You will find the default skin that ships with Midi Quest v8 specifies a standard color of cyan and an edited color of yellow.

Background

The entire background of an editor can be specified in one of two ways.

You can check "Display Background" and provide the name of the .bmp to use as the background.

Alternately, you can place a .bmp in the directory containing the instrument support files. For example, the 01W support files are found in the Instr\01W directory. In this case, the file name but be the same as the editor file name but with the .bmp extension. For example, the Combi editor has a file name of "Combi.sqt". To supply your own bitmap, you should use "Combi.bmp" as the name of the bitmap.

In either case, the bitmap is used to fill the entire window background. if the bitmap is smaller than the editor, the bitmap will be replicated a sufficient number of times so as to fill the window display.

Borders

The borders setting selects whether the original border graphics are used or whether bitmap substitution is used, when specified.

"Original" specifies the use of the standard frame

"Bitmap" specifies that bitmaps be substituted if they are defined in the bitmaps section of the dialog

Envelope

The envelopes in patch editors can be drawn in one of two ways. The "Original" style is to draw the envelope as a simple line with grab boxes. When "Bitmap" is selected and a bitmap is specified in the Envelope section of the Bitmaps section of the dialog, then the envelope is drawn in a filled form. This is the default for v8.0.

VSlider Bitmap

This parameter is used to specify the bitmap that is used to fill the vertical bitmaps. If an alternate bitmap is not specified, Midi Quest uses the default "VSlider.bmp" file.

Toggle Bitmap

This parameter is used to specify the bitmap that is most frequently used to displayed toggle parameters. If an alternate bitmap is not specified, Midi Quest uses the default "ToggleOn.bmp" file

Editor Font

This section of the dialog is used select the font to be used by the skin.

Small is defined as an 8 pixel fixed width "Terminal" font. This was the original font used by the program is still required when creating floating editors.

System uses the defined system font and is the default option.

Large is defined as a 16 pixel fixed with "Terminal" font.

Custom allows a custom font to be selected. When this option is chosen, you MUST click on the "Select Font" button and choose the font you wish to use with skin.

Color Map

This button opens a color definition dialog which allows you to define each of the 16 colors for the editors. This dialog has its own page. [Click here](#) to reach it.

Save

The current skin definition is saved under the name entered in "Skin Select". If a skin of that name already exists, you are prompted before it is overwritten.

Delete

The skin selected in "Skin Select" is deleted.

Apply

The current skin definition is saved as the active skin to be used by all editors. Any currently open editor is automatically redrawn using the new skin. The only exception is the defined font. A change in font only comes into effect as new editors are opened.

Restore

When a skin is loaded for editing, a backup is automatically created. To return to the original skin, just click the Restore button.

OK

Press the OK button to close the dialog

How Midi Quest Finds Skin Bitmaps

For any location in the Midi Quest Skins dialog, you have two choices for how you can enter file names. You can use a complete file path designation such as "c:\MyBitmapFiles\Sound Quest\Bitmap.bmp". In this case, the program looks to this specific location to find the specified bitmap. This method is fine if you are planning on just using the skins yourself.

If you are planning on distributing your skins, we strongly recommend that you use relative file locations. In this case, your bitmaps should be placed in the MQBmp directory or in any directory you care to create below MQBmp. In this case, you just enter the bitmap's file name if the file is in MQBmp or the directories and file name if the file is in a subdirectory. In this case, Midi Quest will automatically add the necessary path information to find and load the file.

For example, if Midi Quest was installed in "c:\Program Files\Sound Quest\Midi Quest" and you entered the file name "Neon\Blue.bmp", Midi Quest would look for your file in:

```
c:\Program Files\Sound Quest\Midi Quest\MQBmp\Neon\Blue.bmp"
```

This method is useful for distribution because, regardless of where Midi Quest is installed, as long as the files are placed in a subdirectory of MQBmp, the program will be able to find them.

Storage of Skins Information

All Skins settings are stored in the main program directory with the file name "MQ SKINS.ini". If you wish to edit this file in its raw form, you can open it into any text editor and make whatever changes you wish.

So what does all of this work result in? Well, you can have editors that look just like the editors in Midi Quest v7.0:

01w - Korg 01/W Group

Send Edit R

Combi Bank

Patch Bank

Drums

Global

0. Ephemerals
1. 16' Piano
2. Orchbrass
3. Woodwind
4. RosewoodGt
5. VS Bells 1
6. XFade Bass
7. TheStrings
8. Residrops
9. Total Kit
10. GhostRyder
11. DigiPiano1
12. OrchTrpts
13. Alto Sax
14. Alan's Run
15. Marimba
16. B.Bass
17. ChamberEns
18. Tidal Wave
19. Jet Stream
20. OxygenMask
21. Perc.Org 1
22. Brass Band
23. Bottles
24. ZingString
25. SolarBells
26. RezzzzBass
27. Analog Pad
28. SynPiano
29. Dance Kit
30. Fresh Air
31. DoubleStop
32. FrHrn&Tuba
33. Sweet Oboe
34. Harmonics1
35. SteelDrums
36. Pick Bass
37. Choir L+R

Pitch Eg Int

Osc 1 Pitch Eg 0

Osc 2 Pitch Eg 0

Cutoff MG

Osc1 Osc2 Key Wform

Freq 59

Delay 0

Inten 0

Joy Stick

PBend +2

Sweep 0

VDFMg 2

Osc-1 Pan

Osc-2 Pan

Oscillator 1

	Att	Dec	Slo	Rel
F Eg KT	0	0	0	0
F Eg Vel	0	+	0	0
A Eg KT	0	0	0	0
A Eg Vel	+	0	0	0

EG Polarity Swi

OSC 1 VDA 15 75 24 99 82 59 18

Level

- AMP KBD Trk
- AMP Vel Sen
- EG Tim Trak
- EG Tim Vel
- KBD Trk Ctr

OSC 2 VDA 15 62 24 99 67 59 14

Level

- AMP KBD Trk
- AMP Vel Sen
- EG Tim Trak
- EG Tim Vel
- KBD Trk Ctr

OSC 1 VDF 0 +99 77 +77 71 +90 99 0

Level

- EG Tim Trak
- EG Tim Vel

Or you can add bitmap backgrounds for an editor which looks like this:

01w - Korg 01/W Group

Send Edit R

Combi Bank

Patch Bank

Drums

Global

0. Ephemerals
 1. 16' Piano
 2. Orchbrass
 3. Woodwind
 4. RosewoodGt
 5. VS Bells 1
 6. XFade Bass
 7. TheStrings
 8. Residrops
 9. Total Kit
 10. GhostRyder
 11. DigiPiano1
 12. OrchTrpts
 13. Alto Sax
 14. Alan's Run
 15. Marimba
 16. B.Bass
 17. ChamberEns
 18. Tidal Wave
 19. Jet Stream
 20. OxygenMask
 21. Perc.Org 1
 22. Brass Band
 23. Bottles
 24. ZingString
 25. SolarBells
 26. RezzzzBass
 27. Analog Pad
 28. SynPiano
 29. Dance Kit
 30. Fresh Air
 31. DoubleStop
 32. FrHrn&Tuba
 33. Sweet Oboe
 34. Harmonics1
 35. SteelDrums
 36. Pick Bass
 37. Choir L+R

Pitch Eg Int

Osc 1 Pitch Eg 0

Osc 2 Pitch Eg 0

Cutoff MG

Osc1 Osc2 Key Wform Pitch

Freq Delay Inten

Triang 59 0 0

VDF Cut VDF MG VDA Amp

Joy Stick

PBend Sweep VDFMg

Osc-1 Pan Osc-2 Pan

Oscillator 1

Att Dec Slo Rel

F Eg KT F Eg Vel A Eg KT A Eg Vel

EG Polarity Swi

OSC 1 VDA 15 75 24 99 82 59 18

99 75 50 25 0

Level AMP KBD Trk AMP Vel Sen EG Tim Trak EG Tim Vel KBD Trk Ctr

OSC 2 VDA 15 62 24 99 67 59 14

99 75 50 25 0

Level AMP KBD Trk AMP Vel Sen EG Tim Trak EG Tim Vel KBD Trk Ctr

OSC 1 VDF 0 +99 77 +77 71 +90 99 0

99 50

Level EG Tim Trak EG Tim Vel

or you can remap the colors to a set that you prefer:

01w - Korg 01/W Group

Send Edit R

Combi Bank

Patch Bank

Drums

Global

0. Ephemerals
1. 16' Piano
2. Orchbrass
3. Woodwind
4. RosewoodGt
5. VS Bells 1
6. XFade Bass
7. TheStrings
8. Residrops
9. Total Kit
10. GhostRyder
11. DigiPiano1
12. OrchTrpts
13. Alto Sax
14. Alan's Run
15. Marimba
16. B.Bass
17. ChamberEns
18. Tidal Wave
19. Jet Stream
20. OxygenMask
21. Perc.Org 1
22. Brass Band
23. Bottles
24. ZingString
25. SolarBells
26. RezzzzBass
27. Analog Pad
28. SynPiano
29. Dance Kit
30. Fresh Air
31. DoubleStop
32. FrHrn&Tuba
33. Sweet Oboe
34. Harmonics1
35. SteelDrums
36. Pick Bass
37. Choir L+R

Pitch Eg Int

Osc 1 Pitch Eg 0

Osc 2 Pitch Eg 0

Cutoff MG

Osc1 Osc2 Key Wform Pitch

Freq Delay Inten

Triang 59 0 0

VDF Cut VDF MG VDA Amp

Joy Stick

PBend Sweep VDFMg

Osc-1 Pan Osc-2 Pan

+2 0 2

Oscillator 1

Att Dec Slo Rel

F Eg KT F Eg Vel A Eg KT A Eg Vel

0 0 0 0 0 + 0 0 0 0

EG Polarity Swi

OSC 1 VDA 15 75 24 99 82 59 18

99 75 50 25 0

Level AMP KBD Trk AMP Vel Sen EG Tim Trak EG Tim Vel KBD Trk Ctr

OSC 2 VDA 15 62 24 99 67 59 14

99 75 50 25 0

Level AMP KBD Trk AMP Vel Sen EG Tim Trak EG Tim Vel KBD Trk Ctr

OSC 1 VDF 0 +99 77 +77 71 +90 99 0

99 50

Level EG Tim Trak EG Tim Vel

or select something completely different

01w - Korg 01/W Group

Send Edit R

Combi Bank

Patch Bank

Drums

Global

0. Ephemerals
 1. 16' Piano
 2. **Orchbrass**
 3. Woodwind
 4. RosewoodGt
 5. VS Bells 1
 6. XFade Bass
 7. TheStrings
 8. Residrops
 9. Total Kit
 10. GhostRyder
 11. DigiPiano1
 12. OrchTrpts
 13. Alto Sax
 14. Alan's Run
 15. Marimba
 16. B.Bass
 17. ChamberEns
 18. TidalWave
 19. Jet Stream
 20. OxygenMask
 21. Perc.Org 1
 22. Brass Band
 23. Bottles
 24. ZingString
 25. SolarBells
 26. RezzzzBass
 27. Analog Pad
 28. SynPiano
 29. Dance Kit
 30. Fresh Air
 31. DoubleStop
 32. FrHrn&Tuba
 33. Sweet Oboe
 34. Harmonics1
 35. SteelDrums
 36. Pick Bass
 37. Choir L+R
 38. Raw Deal

Pitch Eg Int

Osc 1
 Pitch Eg 0

Osc 2
 Pitch Eg 0

Cutoff MG

Osc1 Osc2 Key Wform Pitch

Freq 59 VDF Cut

Delay 0 VDF MG

Inten 0 VDA Amp

Joy Stick

PBend +2

Sweep 0

VDFMg 2

Osc-1 Pan Osc-2 Pan

Oscillator 1

	Att	Dec	Slo	Rel
F Eg KT	0	0	0	0
F Eg Vel	0	+	0	0
A Eg KT	0	0	0	0
A Eg Vel	+	0	0	0

EG Polarity Sw

OSC 1 VDA 15 75 24 99 82 59 18

99
75
50
25
0

Level
AMP KBD Trk
AMP Vel Sen
EG Tim Trak
EG Tim Vel
KBD Trk Ctr

OSC 2 VDA 15 62 24 99 67 59 14

99
75
50
25
0

Level
AMP KBD Trk
AMP Vel Sen
EG Tim Trak
EG Tim Vel
KBD Trk Ctr

OSC 1 VDF 0 +99 77 +77 71 +90 99 0

99
50

Level
EG Tim Trak
EG Tim Vel

Using the skins you can replace all of the bitmaps used by Midi Quest to create your own custom and unique editors. If you have any creations you would like to share with the world, please contact Sound Quest. We intend to build a collection of layouts which can be downloaded from our web site.

Move Keywords (Library)

Purpose:

Move Keywords does precisely that, it allows you to move key words from one key column to another.

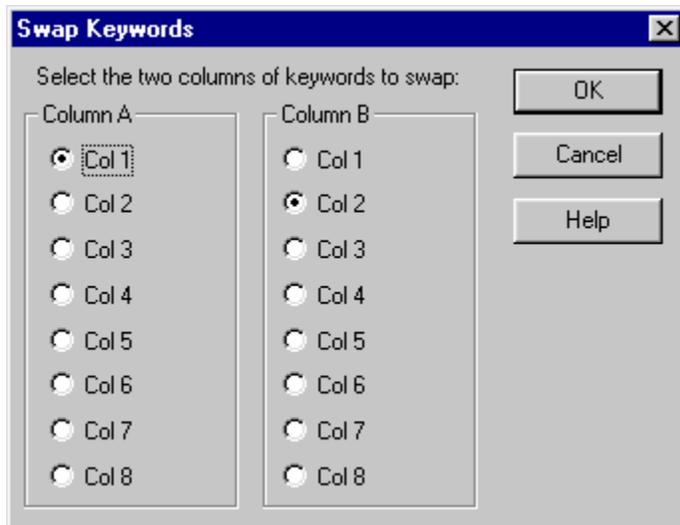
This feature allows you to quickly organize your keys. You can select the patches containing the keywords whose locations you would like to change. Run this function and use the displayed dialog to select the source and destination keywords columns. Pressing OK will actually swap the keyword locations so that no data is lost.

Directions:

To move keywords:

- (1) Select the patches whose keys you would like to move
- (2) Choose *Library/Keys/Move Keywords*
- (3) Select the source and destination columns
- (3) Press the OK button to proceed with the swap

Swap Keywords Dialog



This is the dialog that is displayed when *Library/Keys/Move Keywords* is chosen. The dialog allows you to select a source and destination key column. When OK is pressed the contents of the two columns is actually swapped so that no data is lost.

See also: [Library Window](#)

Merge Keywords (Library)

Purpose:

"Merge Keywords" collects all of the key words assigned to all of the currently selected patches. It then assigns all of the keywords to each of the these patches.

This function makes it easy to assign a set of keywords to an entire group of patches.

Directions:

To merge keywords:

- (1) Select all of the patches containing the keywords you wish to assign to each of the patches.
- (2) Choose Library/Keys/Merge Keywords
- (3) Midi Quest will merge and assign the keywords as described above

See also: [Library Window](#)

Midi Parser (SysX View)

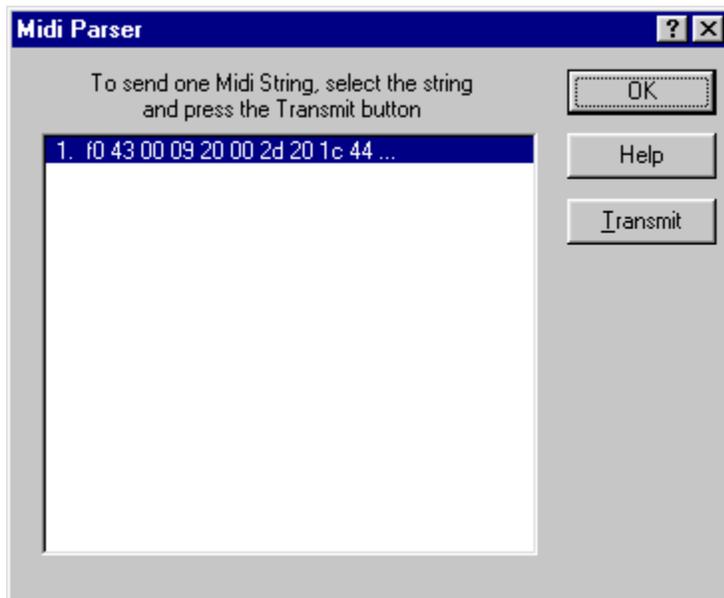
Purpose:

This selection opens the Midi Parser dialog window. This window breaks down the data in the SysX View Window into its individual SysX components. It is then possible to send each SysX string separately.

Directions:

(1) Select *SysX View/Midi Parser*

Midi Parser Dialog



This dialog is of little general purpose. It was written to provide easy access to SY77/TG77 data banks in non-Midi Quest format. Seeing as this data uses varying sizes of patches, it is difficult to access the individual patches other than from this dialog.

See also: [SysXView Window](#)

Advance Voice Size (SysX View)

Purpose:

This function automatically advances the data in the SysX View window by the number of bytes defined in the Voice Bytes parameter of a bank. This function should make jumping from one patch to another in a bank easier.

Directions:

(1) Select *SysX View/Advance Voice Size*

See also: [SysXView Window](#)

Backup Voice Size (SysX View)

Purpose:

This function automatically backup the data in the SysX View window by the number of bytes defined in the Voice Bytes parameter of a bank. This function should make jumping from one patch to another in a bank easier.

Directions:

(1) Select *SysX View/Backup Voice Size*

See also: [SysXView Window](#)

Update Settings (SysX View)

Purpose:

Opens the Settings Dialog to change the basic settings of the data

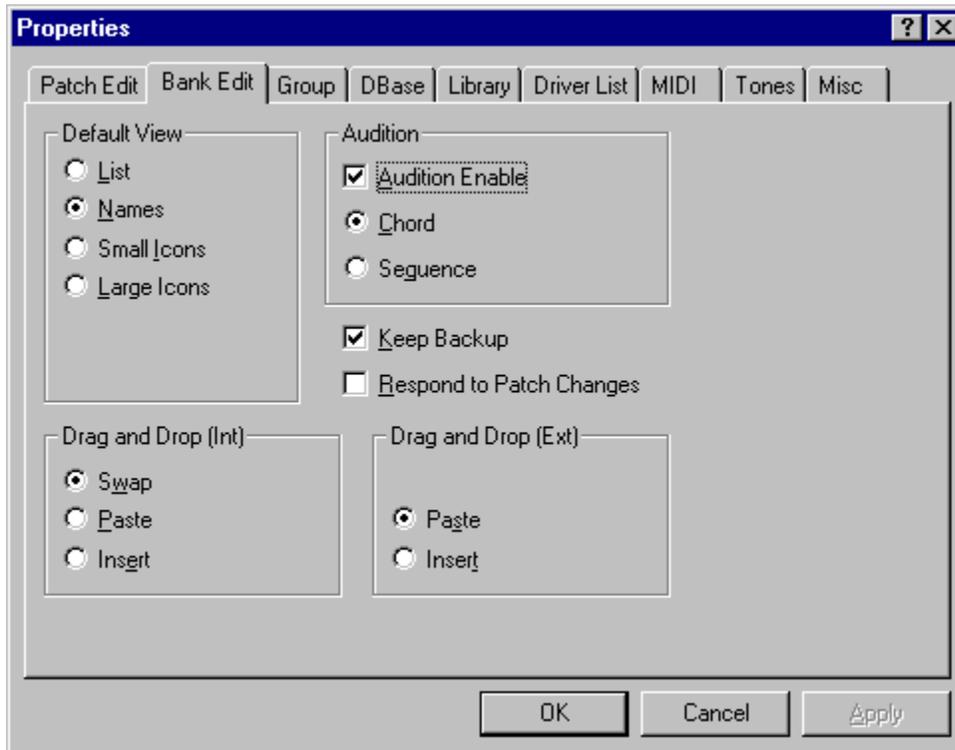
Directions:

- (1) Choose *SysX View/Update Settings...*
- (2) Make any desired changes
- (3) Press OK

See also: [Settings Dialog](#)

See also: [SysXView Window](#)

Bank Edit Preferences (Options Menu)



Default View

Default View selects the initial view style for all Bank editors. There are four view styles with which a Bank can be viewed and edited. This option selects which style in which Bank Editors are initially opened.

Default: Names

Audition

Audition controls auditioning from Bank Windows. When this item is checked, selecting a Patch in a Bank will automatically audition that Patch to the instrument. This provides the quickest and easiest way to try out various Patches determine which one is best suited to the current piece of music. To audition with a chord, select "Chord" or to audition with a sequence, choose "Sequence" The chord or sequence is defined under the Tones Tab.

Default: On

Keep Backup

Keep Backup, when checked, causes the program to automatically create a backup of each subsequently opened Bank Editor. With a backup, any Patch in a Bank can be restored to its pre-edited state. Note: making a backup uses an additional chunk of memory. This was more of a concern years ago when computers had much less memory than they do today. It would be very unusual to need to disable the

memory backup facility.

Default: On

Respond To Patch Changes

When *Respond to Patch Changes* is checked, a patch change command can be used to audition patches in a bank. Any time a patch change is received by the software it will check each bank editor. If a bank editor is auditioning on that channel then the software will use the patch change number to select and audition the patch. This feature allows for the remote selection and auditioning of patches.

For example, a patch change command of MIDI Channel 2, patch 20 is received. The program finds that the M1 is auditioning on MIDI Channel 2. The program automatically auditions patch 20.

Default: Off

Drag and Drop (Int)

The action of dragging and dropping a patch within the same bank (internal) can be handled in one of three ways: swapping, pasting, or inserting.

Swap: the patches in the source and destination patch positions are swapped

Paste: the source patch overwrites the destination patch

Insert: the source patch is inserted at the bank position where the mouse button was released. The insertion is done in such a way that no patches are lost in the bank.

Default: Swap

Drag and Drop (Ext)

The action of dragging and dropping a patch from one bank to another (external) can be handled in one of two ways: pasting or inserting

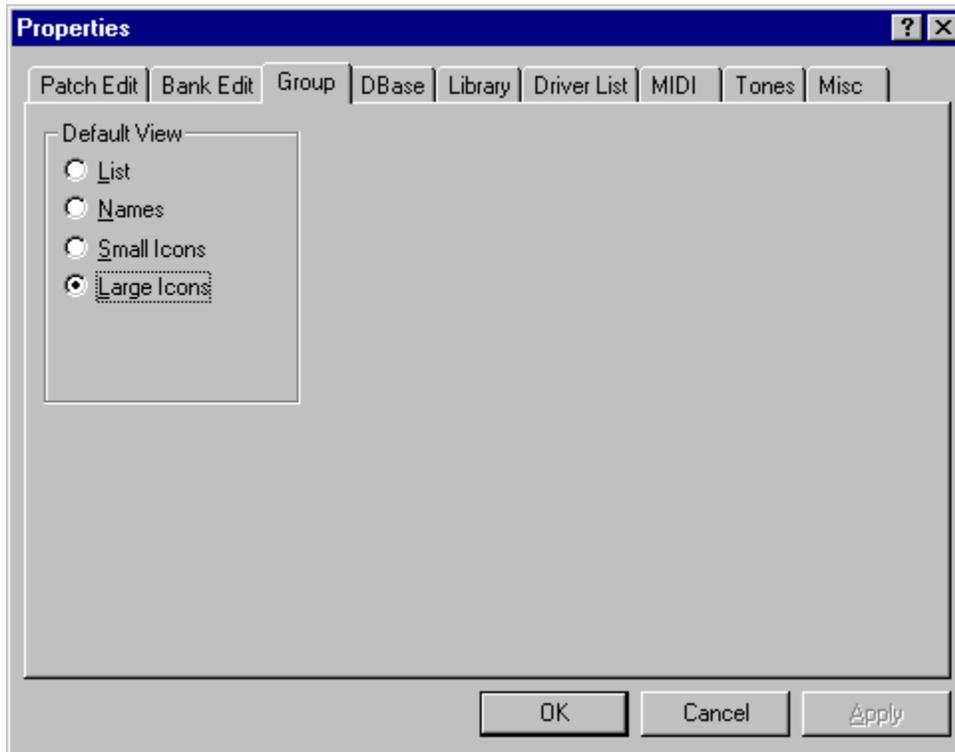
Paste: the source patch overwrites the destination patch

Insert: the source patch is inserted at the bank position where the mouse button was released. All patches in the destination bank from the drop position to the end of the bank move down one position. The last patch in the bank is lost.

See Also: [Bank Editor Window](#)

See also: [Options Menu](#)

Group Preferences (Options Menu)



Default View

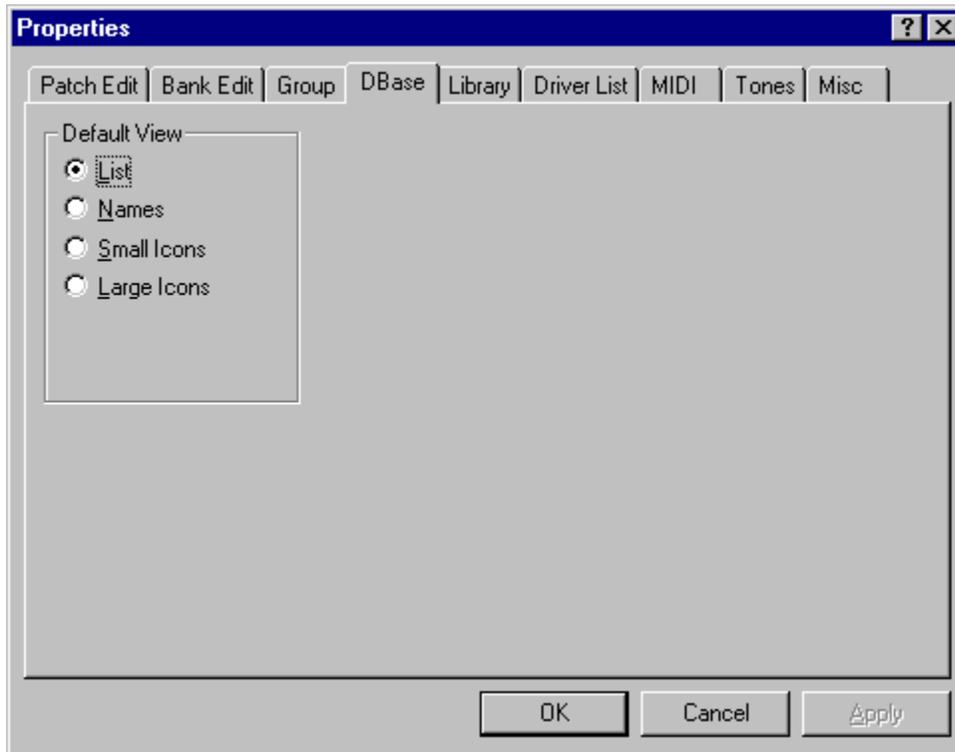
Default View selects the initial view style for all Groups. There are four view styles with which a Group can be viewed and edited. This option selects which style in which Groups are initially opened.

Default: Large Icons

See Also: [Group Window](#)

See also: [Options Menu](#)

DBase Preferences (Options Menu)



Default View

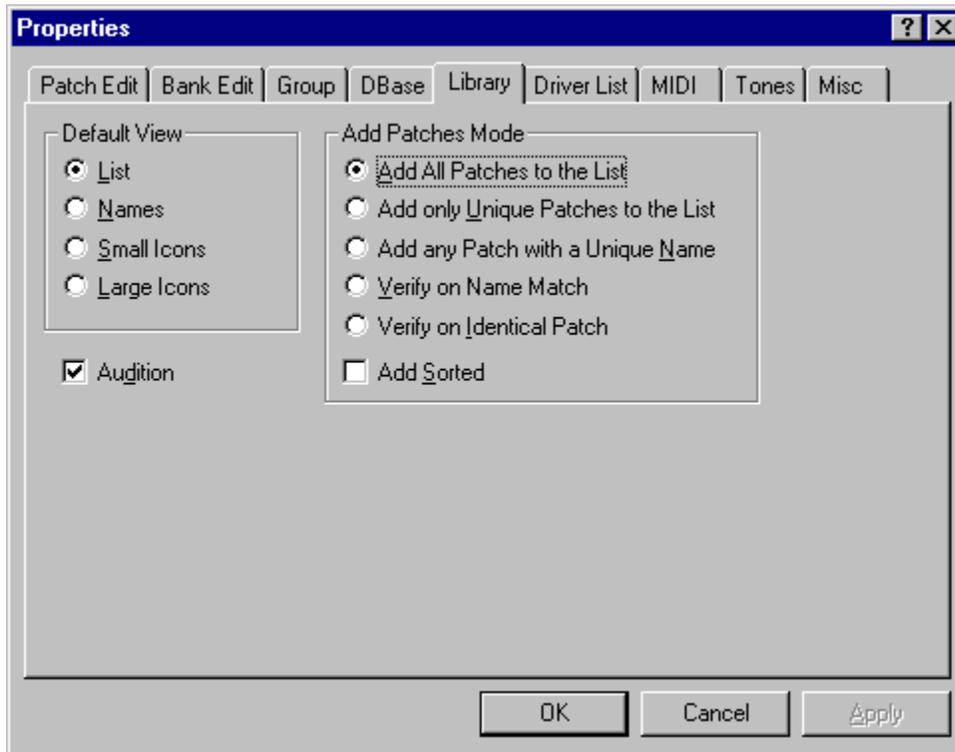
Default View selects the initial view style for all DBases. There are four view styles with which a DBase can be viewed and edited. This option selects which style in which DBases are initially opened.

Default: List

See Also: [DBase Window](#)

See also: [Options Menu](#)

Library Preferences (Options Menu)



Default View

Default View selects the initial view style for all Libraries. There are four view styles with which a Library can be viewed and edited. This option selects which style in which Libraries are initially opened.

Default: List

Audition

Audition controls auditioning from Library Windows. When this item is checked, selecting a Patch in a Library will automatically audition that Patch to the instrument. This provides the quickest and easiest way to try out various Patches determine which one is best suited to the current piece of music. The auditioning chord is defined under the Tones Tab.

Default: On

Add Patches Mode

When patches are added to a Library there are five different ways in which ways in which they can be tested before they are added to the Library

Add All Patches to the List: All patches that are added to the Library will be accepted even if there are duplicates in the Library. This is the fastest way to add patches to a Library as there are no tests

performed when patches are added.

Add Only Unique Patches to the List: only patches which are currently not in the Library are added. Any patch that is already in the Library is ignored. This is a slower but more efficient method of adding patches to a Library. Before each patch is added, the software performs a byte level comparison with each patch already in the Library to ensure that the patch does not already exist.

Add Any Patch with a Unique Name: only patches with unique names are added to the Library. This test allows patches to be added to a Library much more quickly than in the above option but patches that have the same name but possibly different sounds will automatically be rejected.

Verify on Name Match: all patches with unique names are automatically added however, when the patch being added to the Library already exists by name, you are given the option of adding it or not.

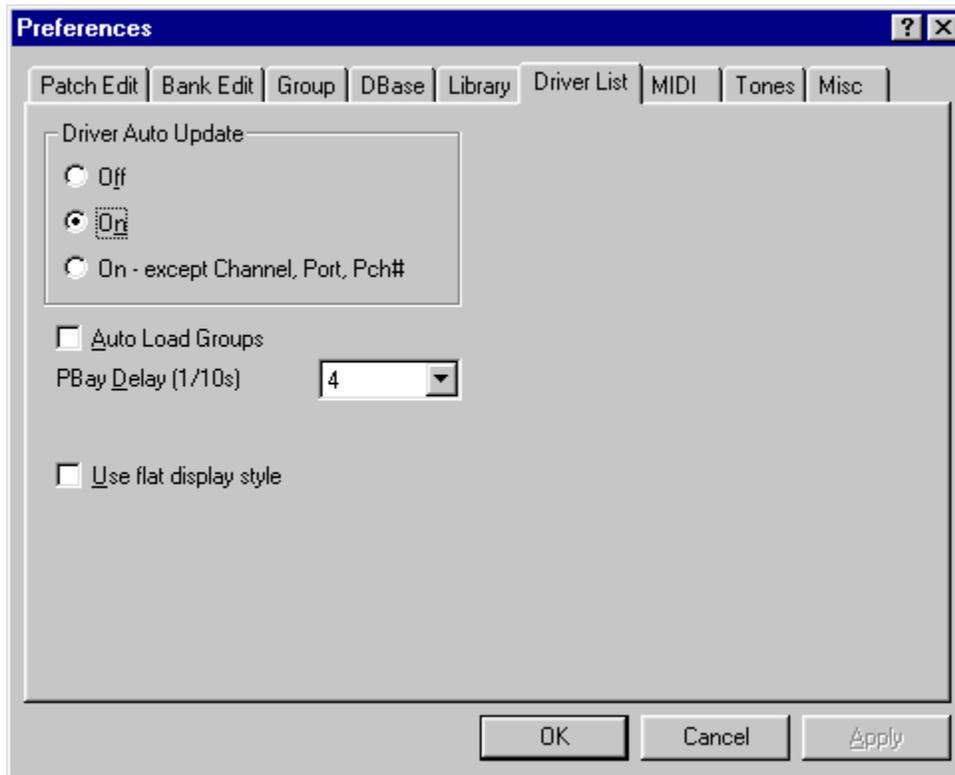
Verify on Identical Patch: all unique patches are automatically added however, when a patch already exists in a Library, you are given the option of adding it or not.

Add Sorted: If this option is checked then when patches are added to the Library, the program attempts to maintain the alphabetical ordering of the patches. If this option is unchecked then patches are appended to the list. Note: using this function only makes sense if create and maintain and alphabetical listing.

See Also: [DBase Window](#)

See also: [Options Menu](#)

Driver List Preferences (Options Menu)



Driver Auto Update

Driver Auto Update controls whether a data file's (Patch or Bank) driver is automatically updated when a new or updated driver is installed in the Driver List. This updating includes, the Comm Channel, Midi Channel, Port, and Pch#. If any of these parameters are changed, any files on disk should be automatically updated when they are loaded. This makes it easy to change your instrument's basic setup without having to change all of the associated data files.

Under most conditions this option should be set to 'On'. If you own two instruments of the same type (eg. 2 TX81Zs), set this option to *On - except Channel, Port, Pch#*. In this mode, any necessary changes to the basic driver will be made but each data file will retain its other settings so that the data will be sent to the correct instrument.

Default: On

Auto Load Groups

When checked, Auto Load Groups will automatically load a Group for each instrument in the Driver List each time the program is started. Using this feature, you can quickly collect the current state of each of your instruments and begin editing.

Default: Off

PBay Delay

When Patch Bay control is enabled in the Driver List Window, this parameter controls the delay between sending the command to configure the patch bay and the the actual dump request to be sent to the instrument. This parameter allows you to optimize the software to match the response time of your patch bay. If your patch bay configures itself quickly after receiving Patch Change commands, this number could be reduced. If the response time of your patch bay is slow, you may need to increase the delay between sending the patch change to the patch bay and sending the dump request to the patch bay so that it can be routed to the instrument.

Default: 4/10s of a second

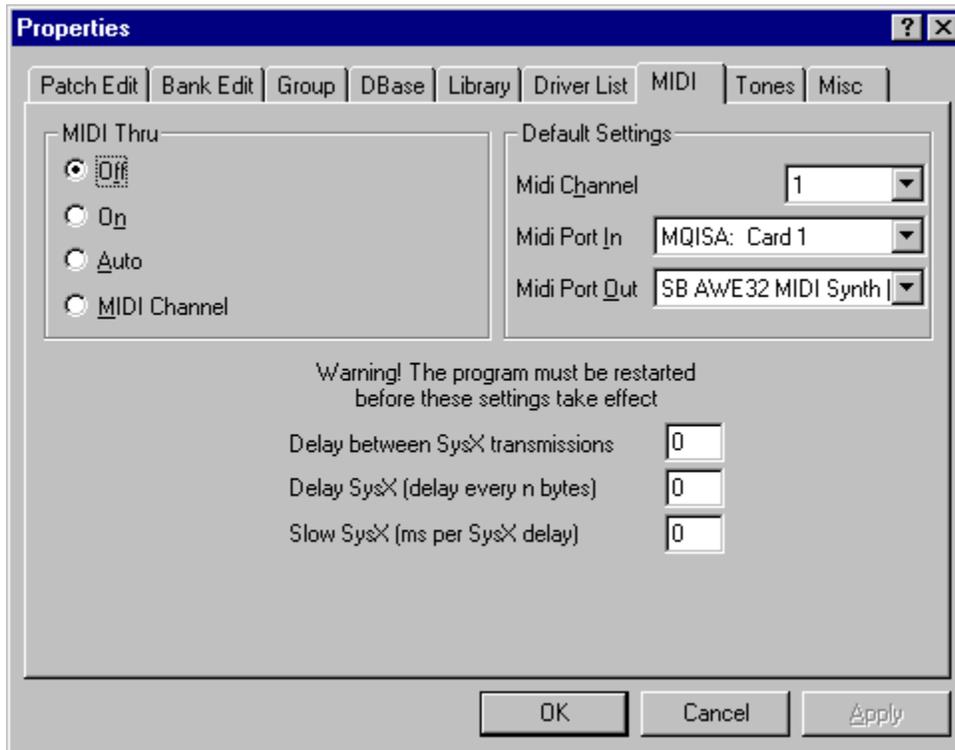
Use Flat Display Style

When checked, the driver list is displayed using standard Windows colors instead of the 3d graphic background.

See also: [Driver List Window](#)

See also: [Options Menu](#)

MIDI Preferences (Options Menu)



MIDI Thru

MIDI Thru is a four phase switch. When active, it takes MIDI events at a MIDI IN Port and sends them out one of the MIDI OUT Ports.

The four options are:

- Off - No MIDI THRU.
- Echo On - MIDI THRU using the received MIDI channel and port.
- Drv Ch - Map MIDI THRU to the current MIDI driver channel and port.
- MIDI Ch - Map MIDI THRU to the default MIDI channel and MIDI OUT Port.

Default: Off

Default Settings

The default settings are the settings that the program uses when the information is not contained within the window itself. The MIDI channel is used when notes are played from a window with no assigned MIDI channel.

MIDI Channel: the MIDI channel used by a window to output notes on when no MIDI channel is assigned to that window

MIDI Port In: the MIDI port used by a window to received MIDI data on when no MIDI port is assigned to that window

MIDI Port Out: the MIDI port used when MIDI data is sent from a window with no assigned ports, for example, MIDIX SysX transmissions.

SysX Transmit Control

Under normal conditions, it should not be necessary to make use of these parameters. However, if you are using a very fast computer with older instruments, it may be necessary to make adjustments in order to properly transfer SysX data.

Why is this the case? The support files for a particular instrument are normally created and tested around the time of that instrument's release. As computer speeds increase, they are capable of sending individual MIDI bytes closer together and in some cases and then able to send data faster than an older instrument is capable of accepting it. These parameter allow you to optimize the program in a global fashion to bypass potential errors.

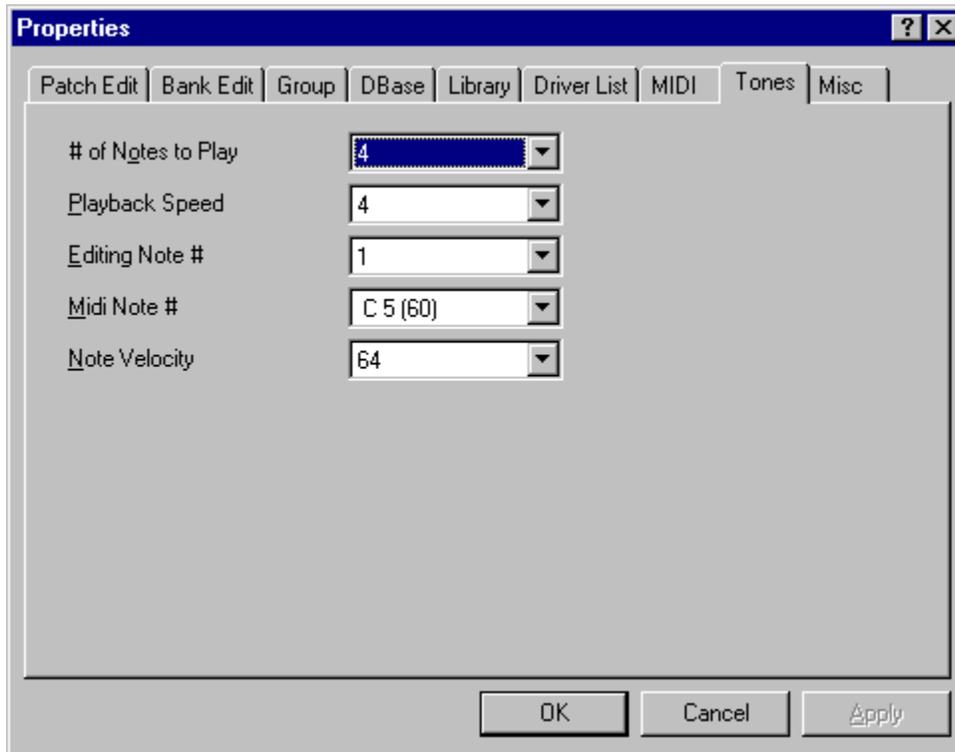
Delay between SysX Transmissions: Midi Quest always delays 80 milliseconds between each SysX transmission and if, during development, we find a longer delay is necessary, we incorporate that delay. In the case where it appears a longer delay is required between each SysX transmission. That delay can be added here. The delay amount is in milliseconds.

Delay SysX: In the case where an instrument can't receive a transmitted SysX message because it is being sent too quickly, this parameter allows the transmission rate to be slowed down. The value "n" entered here indicates how frequently the program should insert a 1 millisecond delay. Entering a value of "1" would cause the program to send 1 byte then delay a millisecond, send 1 byte and delay a millisecond and so on. A value of "10" would result in the program sending 10 bytes, delaying a millisecond, 10 byte sent, a millisecond delay and so on until the message is complete. This is the equivalent of the "DM n" macro command.

Slow SysX: If it is necessary to have a greater than 1 millisecond delay in the function above, a delay of any number of milliseconds can be selected by this parameter.

See also: [Options Menu](#)

Tones Preferences (Options Menu)



The Tones Tab is used to define up to 16 notes that can be played as either a chord or a sequence. The chord/sequence notes defined in the Tones Tab can be triggered from any window.

This feature is ideal for testing new Patches under development in a Patch Editor Window or auditioning from the Bank Editor or Library Windows. All of the information defined in the Tones dialog is saved in the program's configuration file.

Note: if you wish to audition using a sophisticated sequence, use the Sequencer Window to load and play a MIDI sequence created with a sequencer program.

Using the Tones Window

The Tones Window contains five parameters which are used to define the chord and sequence:

of Notes to Play defines the number of notes that are played when a chord and sequence is triggered. The value can be set anywhere between 1 and 16, with a default of 4.

Playback Speed defines the duration of each note (1 = fastest, 16 = slowest).

Editing Note selects the note to display for editing in the MIDI Note # and Note Velocity parameters.

MIDI Note #: sets the pitch of the note selected by the "Editing Note" parameter.

Note Velocity sets the velocity of the note selected by the "Editing Note" parameter.

Triggering a Chord or Sequence

A chord or sequence can be triggered as follows:

- Computer keyboard F2 to play the first defined note
- Computer keyboard F3 to play the second defined note
- Computer keyboard F4 to play the third defined note
- Computer keyboard F5 to play the fourth defined note
- Computer keyboard F6 to play the fifth defined note
- Computer keyboard F7 to play the sixth defined note
- Computer keyboard F8 to play the seventh defined note
- Computer keyboard F9 to play the eighth defined note
- Computer keyboard F10 to play a chord from any window.
- Computer keyboard F11 to play sequence from any window.
- Computer keyboard F12 to replay the last played note
- Bank Edit/Play Tones Chord item.
- Bank Edit/Play Tones Sequence item.
- All of the auditioning options can be found in the program's Midi menu

The MIDI Channel of the Tones

When a chord or sequence is triggered, the MIDI channel used is determined as follows:

Group / Bank / Editor / Library Window - MIDI Ch/Port set in the Settings Dialog

Data Base - MIDI Ch/Port of currently selected data file (or default).

Driver List - MIDI Ch/Port of the first selected driver (or default).

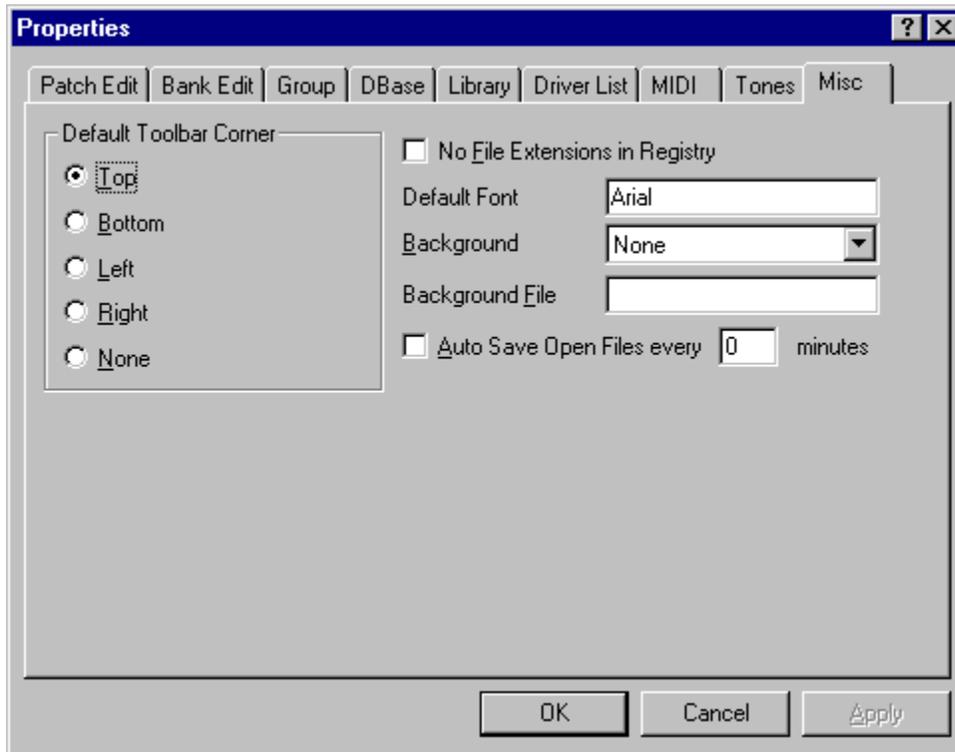
Other Windows - MIDI Ch/Port set in the MIDI Tab

(or default) - if no data is selected, parameters are taken from the Default Window.

See also: Sound Checker

See also: Options Menu

Miscellaneous Preferences (Options Menu)



Default Toolbar Corner

Toolbar Position is a submenu of the Options menu used to select where the toolbar is positioned when the program starts as well as open a new toolbar. The first four options determine the position of the toolbar at startup, the top, bottom, left, or right hand corner of the window. To stop the toolbar from opening, select the No Bar option.

The final option, Open Toolbar will open a new toolbar in the specified location. It is possible to have more than one copy of the toolbar open at one time. To remove a toolbar, grab the toolbar at the top left corner and drag and drop it in to the main window of the program. The toolbar will now have a close button which can be pressed.

Default: Top

No File Extensions In Registry

Normally, Midi Quest will automatically add links in the registry for every file extension that the program uses. This can include many extensions as each type of data for an instrument has its own unique extension along with the standard extensions for libraries, databases, and groups.

After this option is enabled, the program will no longer add references these references. The "Utilities/Uninstall Preparation" will allow you to remove most of the program's registry references.

Default: Off

Default Font

Warning! Take great care when changing this parameter!

It is possible to define the default font that Midi Quest uses while drawing in various windows. The Default font is arial and should be left as such unless you really know what you are doing. If you would like to use a different font, enter its name in the Default Font area.

Default: Arial

Background

Choose the background to use in the main Midi Quest window. This option allows for the selection from no background image, one of a selection of default backgrounds, or the selection of a BMP file to load from disk.

Background File

If the "Disk File (enter name)" option is selected for the Background option then the full path name of the BMP file to use (including the extension) must be entered here.

Auto Save Open Files

If this option is checked then every n minutes, Midi Quest will automatically save all open files to disk. In order for each the file to be saved, it must have a valid path and file name so new files with the "New" file name designation are not saved.

See also: [Options Menu](#)

DBase Views (DBase)

32 bit only

A DBase has four different views: Columns, Names, Small Icons, Large Icons. These are identical to the view options you have in Explorer.

To select the default view, the view that each DBase is opened with, see the DBase Tab in the Properties dialog.

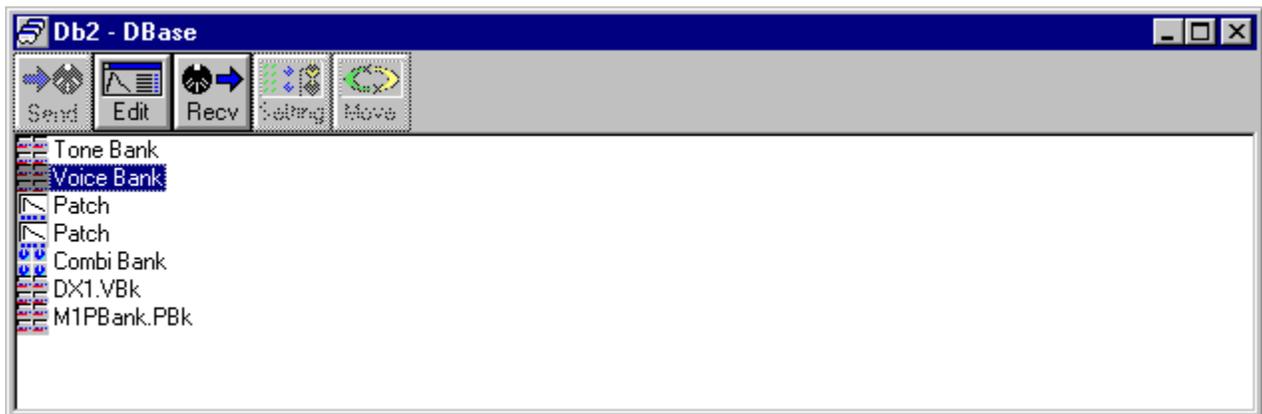
To change the view of the DBase once the window is open, choose *DBase/View Mode* and select the desired view.

Columns (Default)



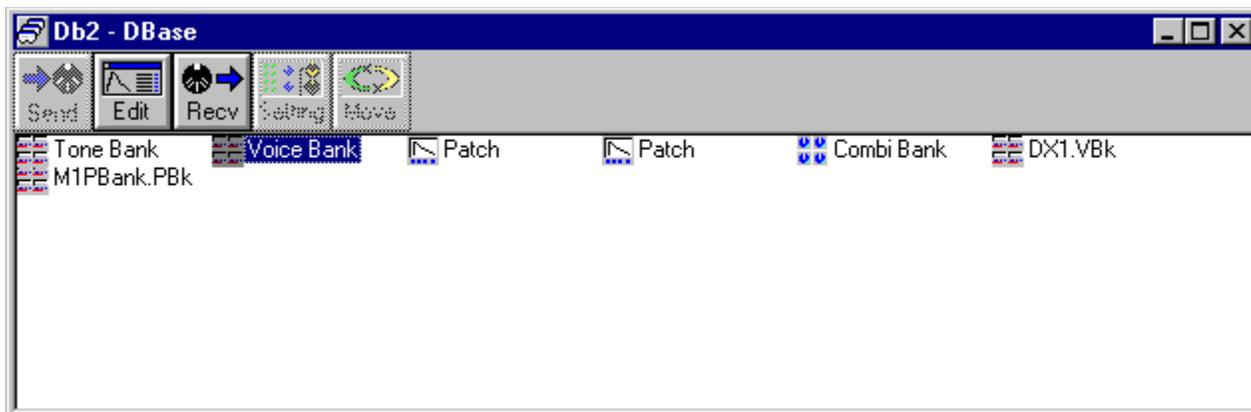
Shows the data files by name including columns for the listing of Data Type, Comm Channel, MIDI OUT port, Date, and File Size..

Names



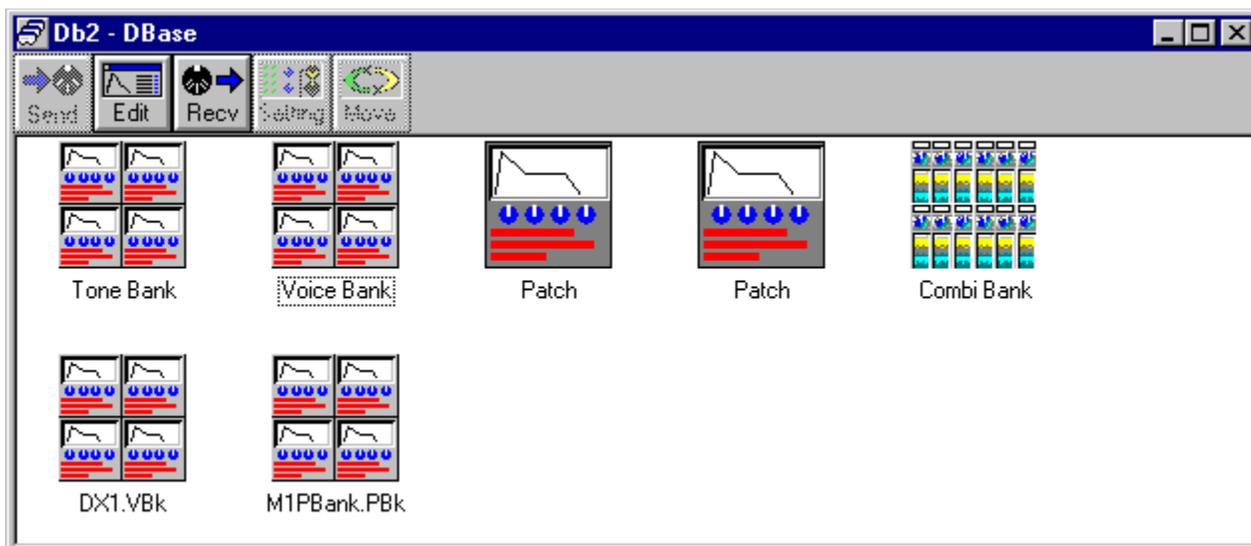
Lists names only in vertical columns with a small identifying icon

Small Icons



Lists names only in horizontal columns with a small identifying icon

Large Icons



Lists name only in horizontal columns with large identifying icon

Group Views (Group)

32 bit only

A Group has four different views: Columns, Names, Small Icons, Large Icons. These are identical to the view options you have in Explorer.

To select the default view, the view that each Group is opened with, see the Group Tab in the Properties dialog.

To change the view of the Group once the window is open, choose *Group/View Mode* and select the desired view.

Columns



Shows the data files by name including columns for the listing of Data Type, Comm Channel, MIDI OUT port, Date, and File Size..

Names



Lists names only in vertical columns with a small identifying icon

Small Icons



Lists names only in horizontal columns with a small identifying icon

Large Icons (Default)



Lists name only in horizontal columns with large identifying icon

Undo (Bank)

[Click here for Patch Undo](#)

Purpose:

Each time the bank is edited, be it using cut, copy, paste, patch edit, and so on, the program automatically makes a backup of the current state of the bank so that you can undo the change. There are an infinite number of undo levels (limited by computer memory). To undo the change, simply select the *Bank/Undo* option. It is possible to undo all the way back to the original unedited bank.

Directions:

To undo the last change:

- (1) Choose *Bank Edit/Undo*
- (2) Verify the undo operation

See Also: [Bank Editor Window](#)

Auto Name (Bank)



Purpose:

Auto Name displays a dialog to enter the basic name for each name in the bank.

You may wish to rename Patches this way when:

- (1) Only the name of the Patch needs to be changed, its faster this way
- (2) There is no Patch Editor for the data, so this is the only way to change the name
- (3) The name is not a component of the Patch Editor so again, this is the only way to change the name

The last option applies only to some older instruments where Patch names were not included as a component of the Patch. In such cases, the program will create space so that names can be assigned and stored. Remember that these names cannot be sent back to the instrument. The instrument does not know how to store them.

Directions:

To auto name each Patch in a Bank:

- (1) Select one or more Patches you wish to name.
- (2) Choose the Name button or *Bank Edit/Edit/Auto Name...*
- (3) Use the displayed Patch Name Dialog to enter a new name
- (4) Press the OK button
- (5) If more than one patch was selected, the dialog will appear for each patch until each has been renamed. If audition is enabled, the patch will be auditioned before the name appears for editing.

See Also: [Bank Editor Window](#)

Set Patch Editor Colors (Options/Skins Menu)

Purpose:

Allows the 16 color color scheme used by the patch editors to be modified. When finished, the color in all editors is updated.

This dialog is

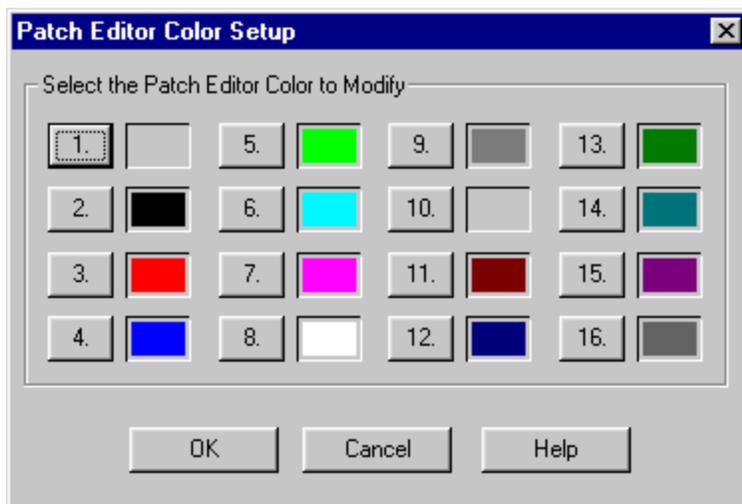
Directions:

To change the colors used by the patch editors:

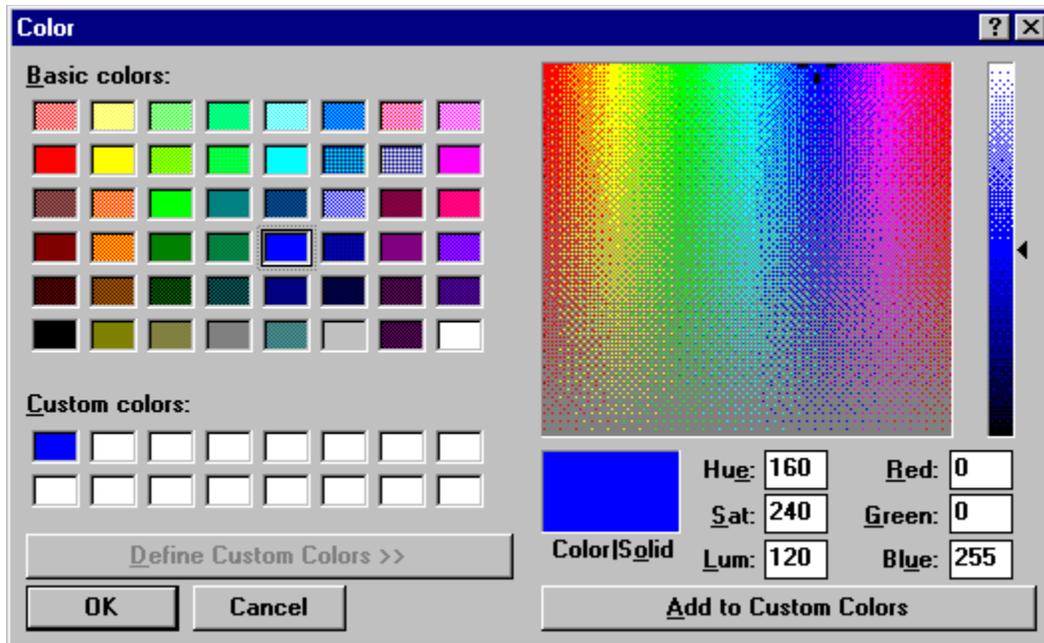
- (1) Click the Color Map button in the Midi Quest Skins dialog.
- (2) Use the Patch Editor Color Setup dialog to select the colors to use in the editors
- (3) Press OK to accept the changes

Set Patch Colors

The Color Setup dialog



The Color Assign dialog



Set Patch Editor Colors... opens a dialog which controls the basic 16 color display used by a particular skin. With this dialog it is possible to reassign any color used by a patch editor. If you wish all text that is displayed in blue to be displayed as dark green, it is as simple as opening this dialog, and replacing blue with dark green. This provides an easy way to change all of your editors without specifically modifying any editor.

Directions:

- (1) Choose *Options/Edit Patch Skins...* and press Color Map in the Skins dialog
- (2) The Color Setup dialog (as pictured above) will appear
- (2) In the dialog are each of the currently assigned 16 colors.
- (3) To change an assigned color, click on its button
- (4) The Color Assign dialog (as pictured above) will appear for the selected color
- (5) Use this dialog to select a new color and press OK
- (6) When finished assigning colors in the Color Setup dialog press OK

Note: if a color is selected in the Color Assign dialog which is not part of the current palette, the closest available color will appear in the editor.

See Also: [Patch Editor Window](#)

See also: [Options Menu](#)

Bank Views

Purpose:

32-bit only

A Bank has four different views: Columns, Names, Small Icons, Large Icons. These are identical to the view options you have in Explorer.

To select the default view, the view that each Bank is opened with, see the Bank Tab in the Properties dialog.

To change the view of the Bank once the window is open, choose *Bank Edit/View Mode* and select the desired view.

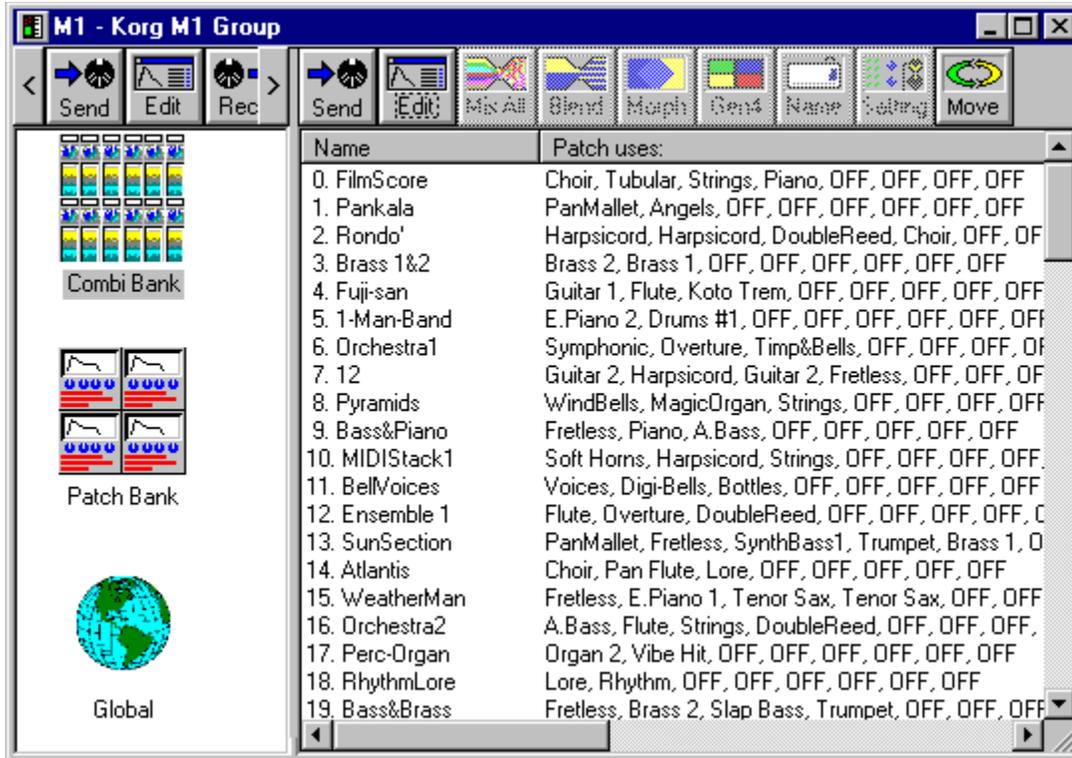
Columns



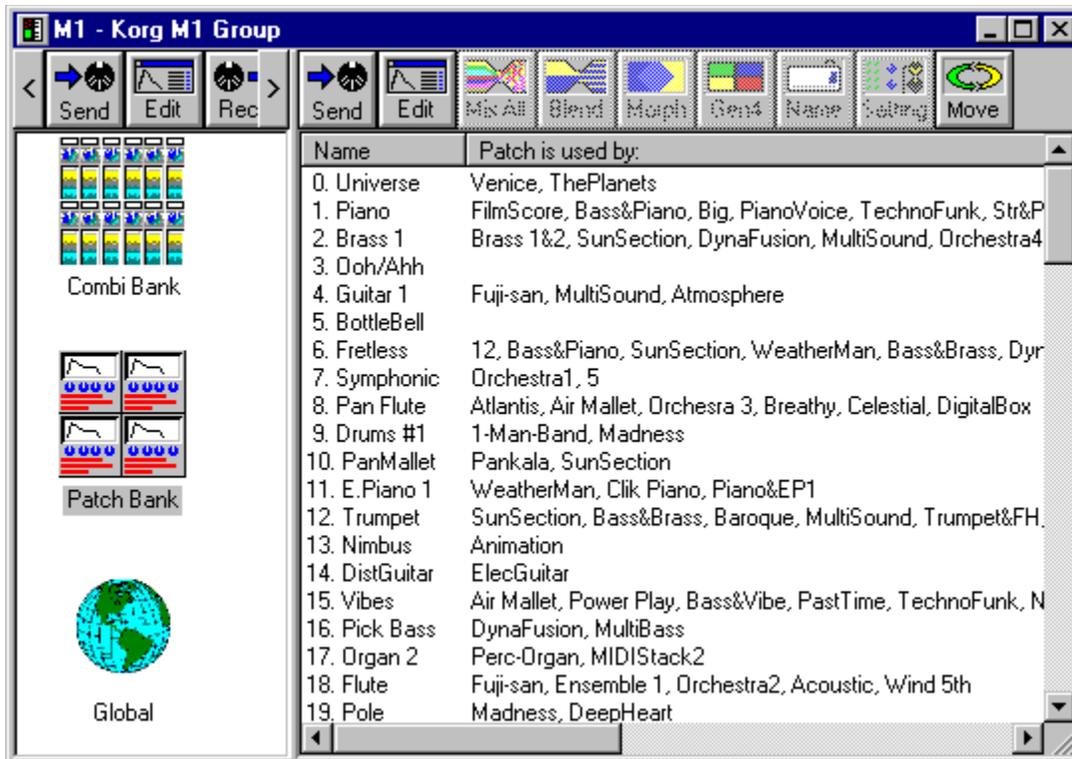
Shows the patches in a list by name.

32 bit only: Below are two examples of banks in list mode as they appear in a Group.

When in a Group, if the Bank is a Parent Bank (the Patches in the bank reference other patches) then Midi Quest will automatically find which child patches are referenced by the patch and will list them. In the M1 example below, you can see that the Combination "Filmscore" uses the patches: Choir, Tubular, Strings, and Piano



When in a Group, if the Bank is a Child Bank (the Patches in the bank are used by other patches) then Midi Quest will automatically find which parent patches are using the displayed patches. In the M1 example below, you can see that the "Piano" patch is used by the Combinations: FilmScore, Bass&Piano, Big, PianoVoice, TechnoFunk, and Str&Piano.



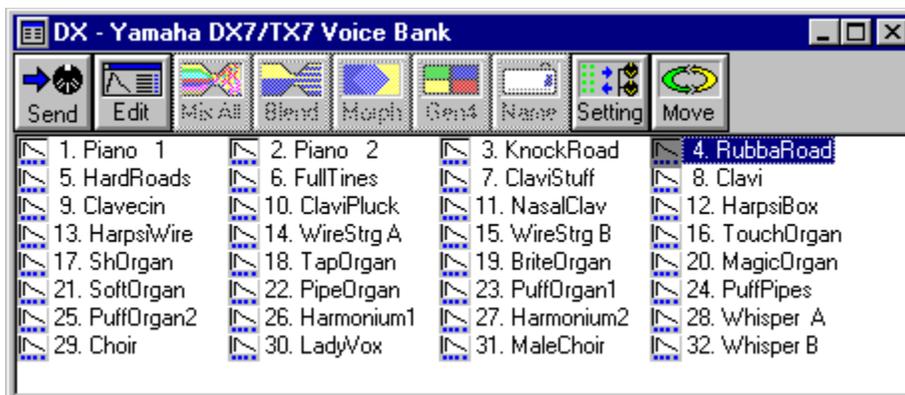
The special lists of patches being used are built by background threads so you do not need to wait while the program builds them. There is one limitation however, you must wait until processing is completed before you can close the window or switch from editing one data type to another. If you do not want to wait for this processing, you should use one of the other display modes.

Names (Default)



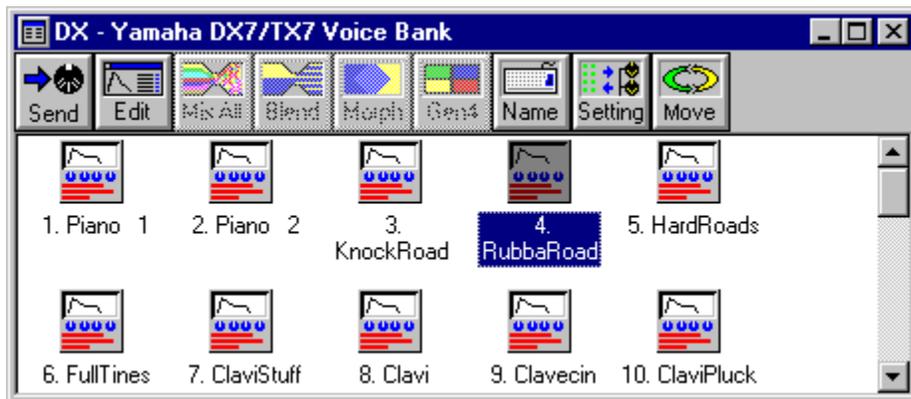
Lists names only in vertical columns with a small identifying icon

Small Icons



Lists names only in horizontal columns with a small identifying icon

Large Icons



Lists name only in horizontal columns with large identifying icon

See Also: [Bank Editor Window](#)

Undo (Patch Editor)

Purpose:

Each time the patch is edited, be it editing a parameter or using cut, copy, paste, patch edit, and so on, the program automatically makes a backup of the current state of the patch so that you can undo the change. There are an infinite number of undo levels (limited by computer memory). To undo the change, simply select the *Sngl Edit/Undo* option. It is possible to undo all the way back to the original unedited patch.

Directions:

To undo the last change:

- (1) Choose *Sngl Edit/Undo*
- (2) Verify the undo operation by responding 'Yes'

See Also: [Patch Editor](#)

Jump To (Patch Editor)

Purpose:

For each editor, you can define up to 10 different jump points. This allows you to quickly get to various points in larger editors.

Directions:

To jump to a particular position in the editor:

(1) Choose *Sngl Edit/Jump To* and select the point you would like to jump to

To set a jump point:

(1) Preposition the editor (relative to the upper left hand corner) to the location you would like to be able to get to

(2) Choose *Sngl Edit/Jump To/Set Jump Point* and select the point to use

(3) A dialog will appear, enter the name you wish to use for the jump point

(4) Press the OK button

(5) Check *Sngl Edit/Jump To* and you will see your jump position in the list

See Also: [Patch Editor](#)

Auto Update (Patch Editor)

Purpose:

Allows control of whether the patch is automatically sent to the instrument when its window is activated or not. By default it is. This parameter is closely tied to the "Update Patch on Activate" parameter in the Properties [Patch Edit Tab](#). If "Update Patch On Activate" is set On or Off then all Patch Editor windows use these settings and the Auto Update function is disabled. However, if "Update Patch On Activate" is set to "Custom" then Auto Update will be enabled (default is on). At this point, you are free to set any type of patch (eg Wavestation Wavesequences) so that it does not automatically update on window activation. To do this, uncheck Auto Update.

Why would you want to do this? Some patches are extremely large and require a notable amount of time to download. If you find that this time lag is interfering with your work, you can disable the auto updating. What effect will this have? Well, you need to watch out for two things. First, when you open the patch it will not be immediately sent to the instrument. You'll need to do this yourself using [Midi/Current Data To Instrument](#). More importantly, if you are editing two patches of the same type, this could lead to confusion because as you switch back and forth the instrument will not be automatically updated. You run the risk of thinking that you have a result you are happy with but find that the settings that created the patch are split between the two patches you are editing. It is best to turn this function off only if you plan on editing only one patch of that type at a time.

Directions:

If Auto Update is enabled:

(1) Choose "Auto Update" to toggle its setting.

See Also: [Patch Editor](#)

Relationships (Library)

Purpose:

Relationships opens a dialog which allows you to set up two types of relationships that apply to all Libraries.

The first relationship is short forms. Any time a patch is added to a library, Midi Quest will automatically search its keyword list. If there are any exact matches, that keyword is automatically added as a keyword for the patch. For example, adding a patch called "Piano 1" will automatically add the keyword "Piano".

However, patch names frequently contain short forms to indicate a type. For example, a patch might be named "RhodesPno". In this case you would like to have the keywords "Rhodes" and "Piano" assigned to the patch. Library Relationships dialog described below allows you to do this.

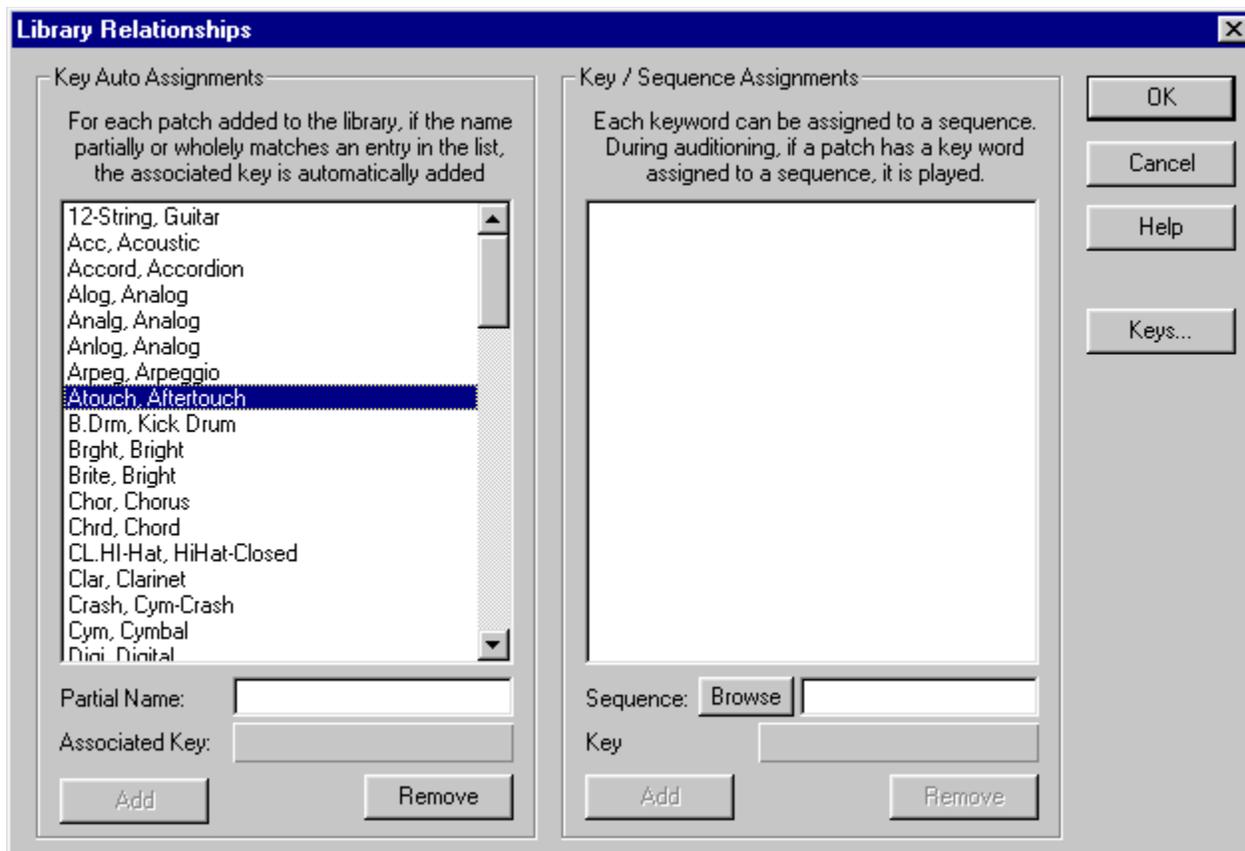
The other relationship is for auditioning.

Directions:

To set up relationships:

- (1) Choose *Library/Keys/Relationships*
- (2) Use the displayed dialog to define relationships for your libraries

The Library Relationships Dialog



Key Auto Assignments

The left hand side of this dialog is dedicated to creating relationships between short forms and key words. For example, you might use the short form "Pno" to indicate that the patch is a piano sound. When you add the patch to a library, you can ensure that the key word "Piano" is automatically assigned to any patch containing the short form "Pno".

To do this, enter the short form in the "Partial Name" entry area. Now, left click in the boxed area beside "Associated key" to display your list of keywords. Select the key word you want associated with this short form. Once you have made the association, press "Add" button to add it to the list.

To remove an association from the list, select the short form and its associated keyword in the list and press the "Remove" key. This operation can not be undone.

You will find that Midi Quest ships with a large number of pre-defined associations. These associations were gleaned from reviewing a number of our larger patch libraries.

Key / Sequence Assignments

While auditioning patches, you may have a MIDI file which is ideal for auditioning basses, other for pianos, another for trumpets, and so on. Using this dialog, you can create a relationship link between a particular keyword and a particular MIDI file. If a patch is auditioned that contains a keyword that is associated with a particular MIDI file then instead of playing a chord, Midi Quest automatically loads and plays the MIDI file instead.

To define the relationship between a key word and a MIDI file, enter the path to the MIDI file in the area to

the right of "Sequence". Alternately, click the "Browse" button to display a file selector. Use this selector to find and select the MIDI file. Next, click in the boxed area to the right of "Key". Select the key to associate with the MIDI file. Once selected, press the Add button to create and store the association.

Once you have associated a key word with a sequence, return to the library and click on a patch containing the key word to hear the results.

To remove an association from the list, select the key word/MIDI file from the list and press the "Remove" key. This operation can not be undone.

See also: [Apply Default Keys](#)

See also: [Library Window](#)

Apply Default Keys (Library)

Purpose:

"Apply Default Keys" performs the same two tests that are automatically performed when patches are added to a library. This allows you to automatically update existing libraries with keywords.

The first test adds keywords that are found as exact matches within the patch names. The second test searches the patch name for short forms and if there is a match, the associated key is added to the patch.

Directions:

To add default keys to patches in a library:

- (1) Select the patches in the library that you wish to have tested
- (2) Choose *Library/Keys/Apply Default Keys*
- (3) Midi Quest will add key words where appropriate

See also: [Relationships](#)

See also: [Apply Default Keys](#)

See also: [Library Window](#)

Compare (Patch Editor)

Purpose:

Compare allows you to contrast the current version of the patch with the patch as it sounded when it was first opened in the editor.

Directions:

To compare the patch now with the patch as it originally sounded,

- (1) Choose *Sngl Edit/Compare*
- (2) A dialog is displayed indicating that the original patch has been sent to the instrument
- (3) Audition the original patch
- (4) When you are finished with the original patch, press the OK button in the dialog
- (5) The edited patch is sent to the instrument and you can continue editing

See Also: [Patch Editor](#)

Select Parameters (Patch Editor)

Purpose:

"Select Parameters" allows you to select a predetermined group of parameters.

This function works in conjunction with [Create Parameter List](#). Create Parameter List will save the currently selected parameters (those highlighted in green) as a list with a descriptive name you assign.

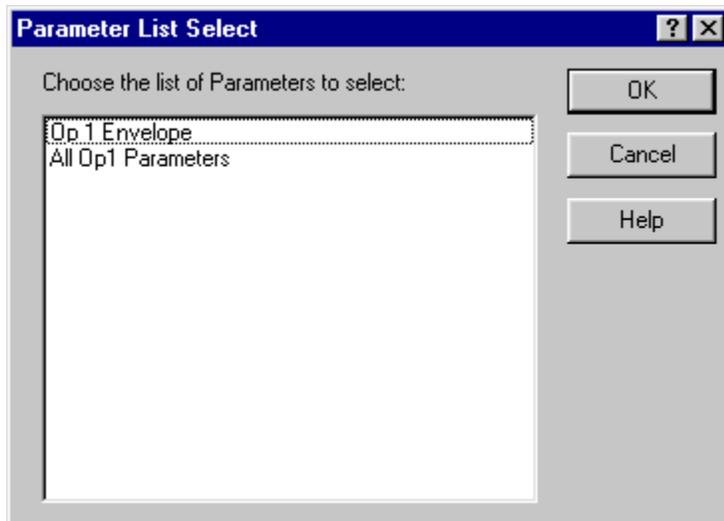
When you wish to have this particular configuration of parameters selected again, simply choose *Sngl Edit/Select Parameters* and select the desired configuration from the displayed list.

Directions:

To select a predefined list of parameters:

- (1) Choose *Sngl Edit/Select Parameters*
- (2) Use the displayed dialog box to select the desired parameter list
- (3) Press the OK button
- (4) The parameters in the selected list will be highlighted in the editor

Parameter List Select Dialog



This dialog is displayed when "Select Parameters" is chosen from the menus. The dialog presents a list of available selection presets for the editor. After selecting the desired preset list, press the OK button and the appropriate parameters will be selected in the patch editor.

See Also: [Selecting Parameters](#)
See Also: [Create Parameter List](#)
See Also: [Patch Editor](#)

Create Parameter List (Patch Editor)

Purpose:

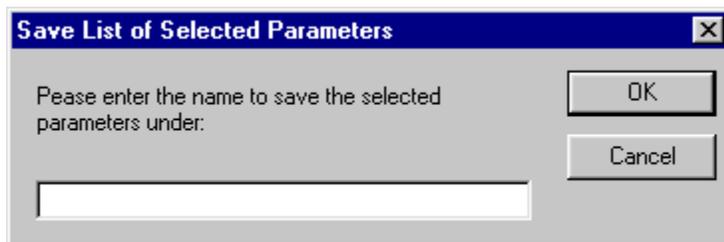
"Create Parameter List" allows you to store the currently selected parameters as a list so that the same selection may easily be made in the future.

Remember, you are not saving the value of the parameter, you are saving the fact that the parameter is selected. This is very useful if you are frequently copying all of the parameters from one oscillator to another. Instead of having to find and select all of the parameters in OSC1, copy them, then find and select all of the parameters in OSC2 and paste them, all you need to do is set up an OSC1 and OSC2 selection list once and then select the OSC1 list, choose copy, select the OSC2 list, and choose paste. Its a real time saver.

Directions:

To store a list of selected parameter:

- (1) Select the parameters you wish to store as a list
- (2) Choose *Sngl Edit/Create Parameter List*
- (3) The following dialog will appear:



- (3) Use this dialog to enter a descriptive name for the selected parameters
- (4) Press the OK button
- (5) The list of selected parameters is saved so that it can be used by Select Parameters

See Also: [Selecting Parameters](#)

See Also: [Select Parameters](#)

See Also: [Patch Editor](#)

New SysX Message (DBase)

Purpose:

"New SysX Message" opens the New SysX dialog so you can enter a custom SysX message. After the message is entered in the dialog, the newly created data file is stored in the DBase.

For more information on the new SysX Message dialog, [click here](#).

Directions:

To create a new SysX message for the DBase:

- (1) Choose *DBase/Open/New SysX Message...*
- (2) Use the displayed dialog

See also: [New SysX Message](#)

See also: [DBase Window](#)

New SysX Message (File/New menu)

Purpose:

This option allows you to create a custom SysX message simply by typing it into a dialog box. This option is ideal for creating simple SysX messages that can be used to place an instrument in a particular configuration. For example, you could create a GM reset message to send to a Roland GM instrument as part of its configuration process.

Directions

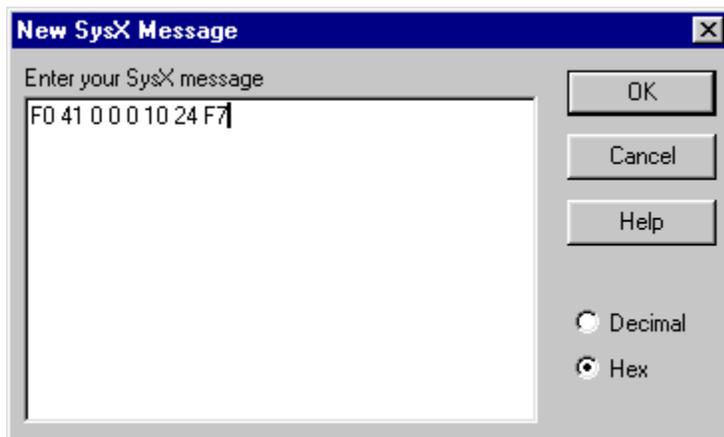
To create a SysX message:

1. Choose *File/New/New Sysx Message...*
2. Enter the SysX message in the dialog
3. Press the OK button to create the SysX message
4. A new SysX View window is created with the message
5. From there it can be manipulated or stored

See Also: [File Menu](#)

See Also: [DBase/New SysX Message](#)

New SysX Message Dialog



This dialog is opened from *File/New/New SysX Message...* or by selecting *DBase/Open/New SysX Message...* from the DBase window. This dialog lets you enter a SysX message of any length and store it for later use with Midi Quest.

Use the text entry area to enter the SysX data. If the **Hex** button is pressed, all of the entered values are assumed to be hexadecimal values. The example above shows a valid SysX message in hex.

If the **Decimal** button is pressed, all entered values are assumed to be in decimal *except* for those that begin with "0x". The equivalent of the message above as entered in decimal would be: 240 65 0 0 0 16 36 247. Alternate, some of the values could be written in hex, as in the following example: 0xF0 65 0 0 0

0x10 36 0xF7.

Once you have entered the SysX message you wish to create, press the OK button. If the dialog was opened from the *File/New/New SysX Message* menu item, the SysX message is opened in a SysX View window. If the dialog was opened from the DBase window using *DBase/Open/New SysX View*, the SysX data will appear as a new entry in the DBase Window.

Note: this dialog should only be used by those who understand SysX data. Midi Quest does not verify the entered data in any way so it is quite possible to enter values which do not create a valid MIDI stream.

Midi Quest (Cubase) VST Plugin

Using the Midi Quest VST Plugin

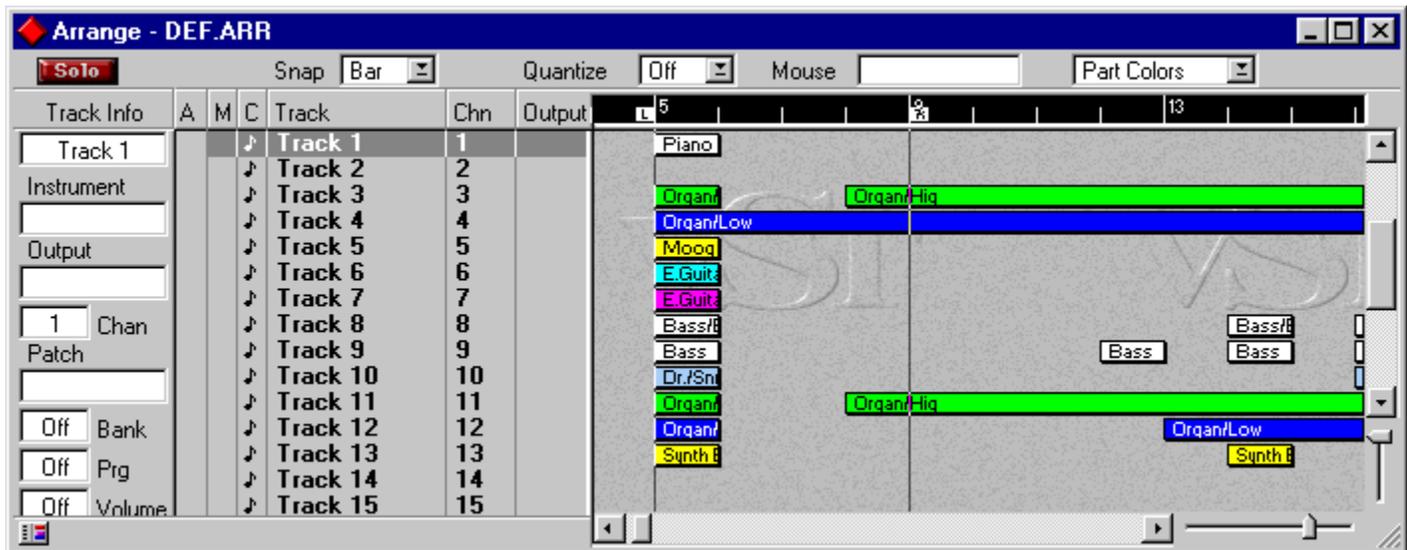
The VST plugin version of Midi Quest is designed so that patches can be created that plugin into any software package that acts as a VST 2.0 host and fully implements the VST 2.0 spec, this includes Cubase v3.7.

Installing the VST Plugin

If the Cubase VST application is already installed when you install Midi Quest then the Midi Quest VST plugin will automatically be installed. If Cubase is not installed or you have an older version (less than v3.7) then you will need to reinstall Midi Quest after installing Cubase in order for the plugin to be correctly configured.

Creating Your First Midi Quest VST Plugin

First, open a Cubase sequence as in the following example:



Next, choose *Audio|VST Instruments* to display the following window:



Left click on "No VST Instr" and you will see the following menu:



Select the "Midi Quest VST" option. If the entry is not there, you will need to either reinstall Midi Quest or copy the "Midi Quest VST.dll" file from your Midi Quest directory to the Cubase\svstplugins directory. After selecting the option, the VST instruments window will appear as follows:



Press the Power button to enable Midi Quest and you're ready to get started.

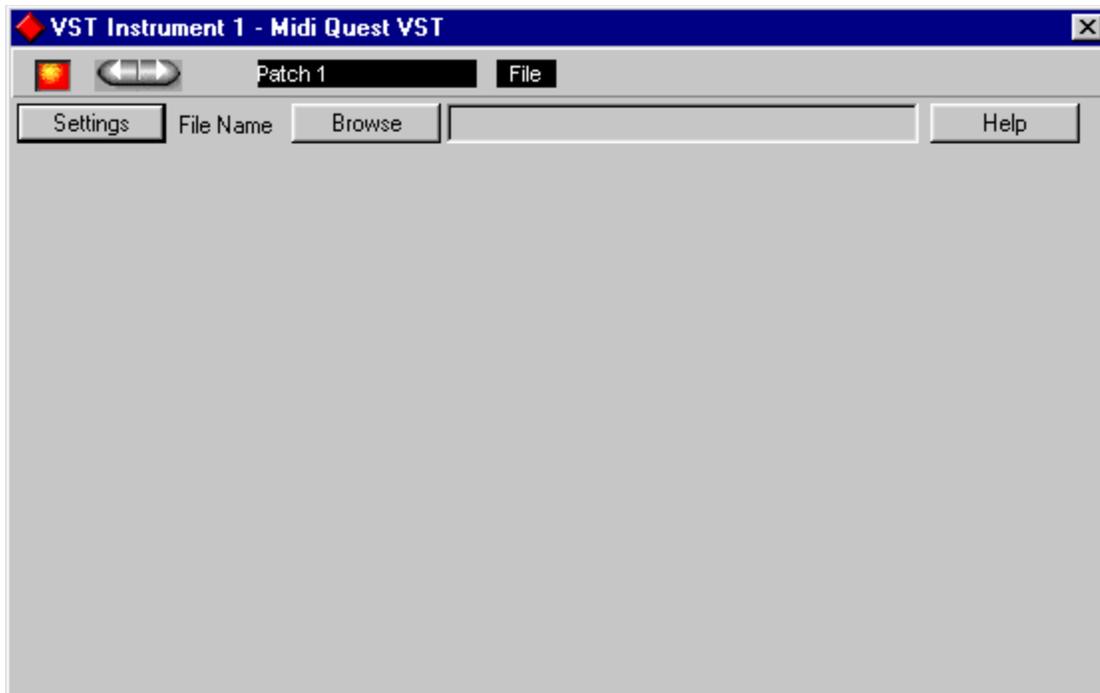
The first thing you need to do is to assign Midi Quest to a track. This is the track where you will insert patch change commands so that Midi Quest can be automated. To do this, go to the Arrange window, select the track to assign to the Midi Quest plugin and click on the Output selector of the track. A popup list of all available MIDI ports and synth plugins is displayed as follows:



Cubase seems Midi Quest as a synth even though Midi Quest doesn't generate any audio.

The Midi Quest VST Window

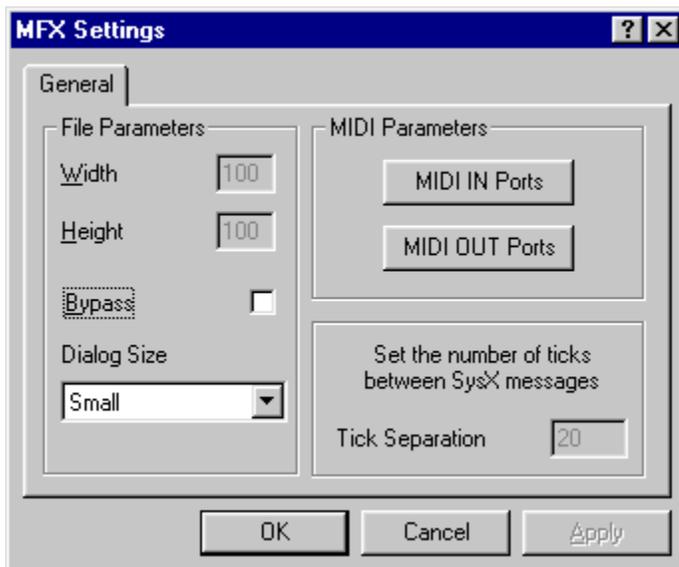
Select "Midi Quest VST" to send the output of the track to the Midi Quest VST plugin. With the output of the track correctly set, if you press the Edit button in the VST Instruments window you will see the following window:



This is the window where you will perform all of your Midi Quest activities.

Selecting MIDI Ports

The first thing you will need to do is select the MIDI ports that Midi Quest will use. While Midi Quest allows you to select a different set of MIDI interfaces for MFX, in most cases you will probably use the same configuration. To select MIDI Ports, press the **Settings** button and you will see the following dialog:



To select MIDI OUT ports, click on the **MIDI OUT Ports** button. Use the displayed Midi Out Ports dialog to select the ports you wish to use.

To select MIDI IN ports, click on the **MIDI IN Ports** button. Use the displayed Midi In Ports dialog to select

the ports you wish to use. Remember that virtually all activity in MFX will be in the form of sending MIDI data, not receiving it so you will probably not need to open any MIDI IN ports.

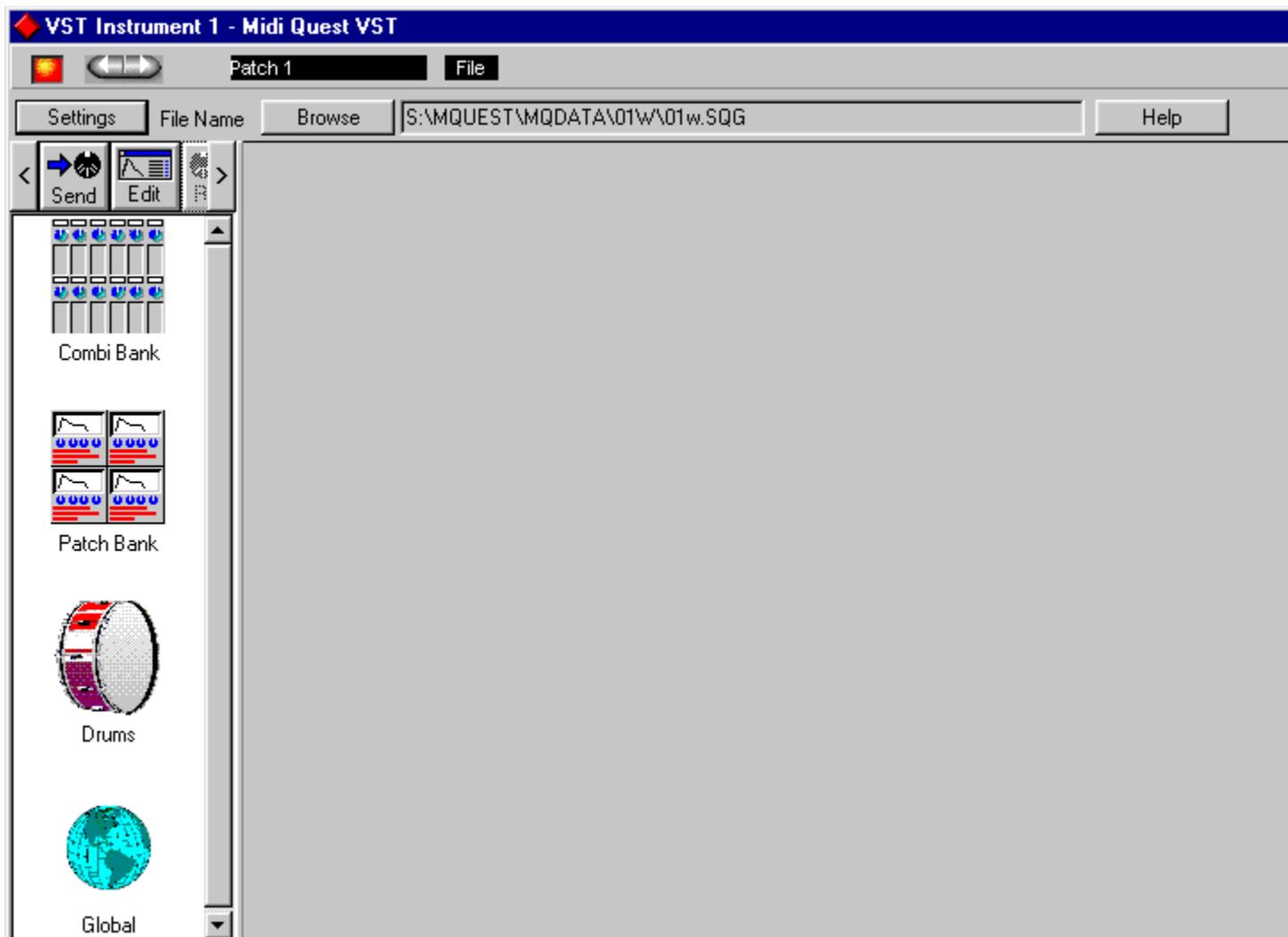
Important: It is also important to note that both Midi Quest and Cubase will need to have these ports open simultaneously so you need to use a MIDI interface with "multi-client" drivers (or constantly enable and disable MIDI drivers in Cubase and Midi Quest VST). Multi-client drivers allow more than one MIDI application to access the interface at the same time. Companies which offer multi-client drivers include MIDI Man, Opcode, MOTU, and Turtle Beach. You will find that the driver which come with your standard sound card, including those made by Creative Labs, are **not** multi-client.

After selecting your MIDI Ports, press the OK button to close the settings dialog.

Selecting a Midi Quest File

The next thing you will want to do is select one of the Midi Quest files that is used by the current song. To do this, press the **Browse** button and a file selector is displayed. Use this selector to choose a Midi Quest file to load. You can choose any Group (.sqg), Library (.sql), DBase (.sqd), Bank, or Patch file. After selecting the file, press the OK button to load it.

If you selected a Group file, you will see something like this:



You will notice that this window look exactly like Group in Midi Quest, and it is. You have all of the same functionality and can edit banks and patches from within the Group. Just remember the virtually all of a particular window's functionality can be accessed by right clicking the mouse button in that window.

That's it. You're now set up to work with Midi Quest VST in Cubase just as if you were using Midi Quest as a separate program.

For information on all of the options found in the Settings dialog, please click [here](#).

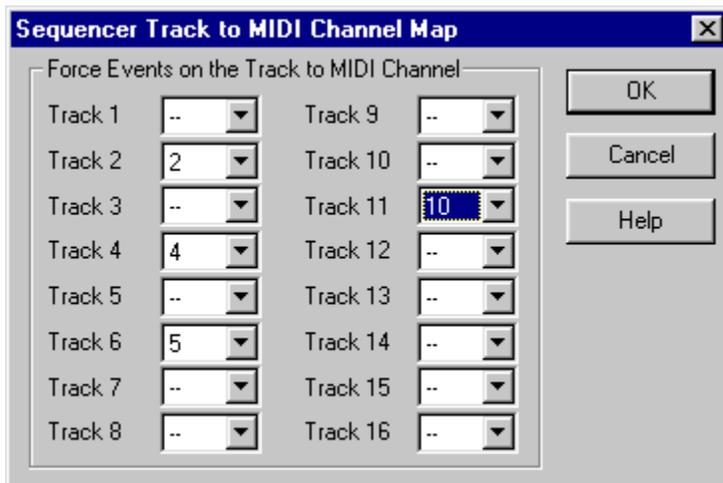
VST Patches

Each VST instrument is actually capable of holding up to 128 "patches" where each patch can be a group, library, dbase, bank, or patch.

See also: [Midi Quest MFX](#)

See also: [MFX./VST Settings](#)

Sequencer Channel Map (Sound Checker)



Purpose:

This dialog is used to force all MIDI events on a particular track to a specific MIDI channel.

For each track you have the option of leaving the track as it is using "--" or select a particular MIDI channel for the track.

Directions:

To assign a MIDI channel:

- (1) Click on "Channels" in the Sequencer Window
- (2) For each track, select a MIDI channel or "--" for no assignment
- (3) Click OK to accept the settings

See Also: [Sequencer Window](#)

See Also: [Sound Checker](#)

Cakewalk Name Support

Using Midi Quest it is easy to export entire banks of names directly into Cakewalk. Using this feature, you can save a lot of time as it is no longer necessary to manually type in all of the names from a custom bank of patches. All you need to do is save a Midi Quest bank in .INS format and import this file into Cakewalk.

Option 1

You have two options. The first option is saving the Bank of names directly into Cakewalk's MASTER.INS file. In this case, once the names are entered, you must create the instrument yourself and link the names to the instrument. At the start of this procedure, Cakewalk must be shut down. The instructions are as follows:

- a) load the bank from the disk or instrument whose names you wish to save to Cakewalk
- b) the bank must be loaded individually, it can not be part of a Group or DBase
- c) choose *Files/Save As...* to save to disk
- d) from the "File types" option in the save dialog, choose the .INS option
- e) using the file selector, move into Cakewalk's directory
- f) double click on the MASTER.INS file
 - g) the program will **ADD** the Bank's Patch Names into the file. It will **NOT** overwrite any other information.
 - h) you now need to quit Cakewalk and rerun it then create an Instrument using the saved bank
 - i) It is important to note, that after adding the bank of names, you must create the instrument yourself and associate the names with the instrument yourself.
- j) To create an instrument and assign the bank of names in Cakewalk v9.0 - Run Cakewalk
- k) Choose *Options/Instrument* from the Cakewalk menus
- l) Click on **Define...** in the "Assign Instruments" dialog
- m) In the "Define Instruments and Names" dialog right click on "Instruments" in the left column
- n) Choose "Add Instrument" from the popup menus
- o) Name the new instrument whatever you wave
- p) Double click on the new instrument to expand it
- q) In the right column, expand the "Patch Names" list
- r) Find the new bank you just saved to the MASTER.INS file
 - the name is the Midi Quest file name with the data type appended
 - eg. MyM1Bank.PBk - Korg M1 Patch Bank
- s) Drag this bank and drop it on "Patch Names for Bank" of the newly created instrument
- t) Replace the default GM bank
- u) Click on Close
- v) In the "Assign Instruments" dialog, assign the instrument to a port and MIDI channel

This option will work with any version of Cakewalk. From v3 through v9.

Option 2

This alternative will work in Cakewalk v9 and there is no need to shut down Cakewalk to perform the function. In this case, you create a separate .INS file on disk, this file is then manually or automatically integrated into the MASTER.INS file. The instructions are as follows:

- a) load the bank from the disk or instrument whose names you wish to save to Cakewalk
- b) the bank must be loaded individually, it can not be part of a Group or DBase
- c) choose *Files/Save As...* to save to disk
- d) from the "File types" option in the save dialog, choose the .INS option
- e) give the file a name and save it anywhere (except the MASTER.INS file)
 - c) once the file is saved, for older versions of Cakewalk you must use a text editor to integrate the file into the MASTER.INS file (Cakewalk must not be running)
 - d) there should be instructions in the Cakewalk manual for accomplishing this
 - e) If you have a copy of Cakewalk v9.0 Cakewalk can automatically create an instrument for you. In Cakewalk, follow these steps:
 - i) Choose *Options/Instruments*
 - ii) Click the **Define** button in the **Assign Instruments** dialog
 - iii) Click the **Import...** button in the **Define Instruments and Names** dialog
 - iv) Select the .ins file you just created in Midi Quest and load the instrument

That'S it. You're done and you've saved a load of time.

Reload (Patch Editor)

Purpose:

This function allows you to reload a patch directly into an editor.

This function is particularly useful under several conditions. You may be working on a patch and have made several edits on the synthesizer instead of the editor. You can use this function to load the patch from the instrument directly into the editor.

In situations where the instrument internally changes a number of parameters as a result of changing a different parameter you can reload the data to ensure that the editor and synthesizer are in sync. An example of this situation is when a new effect is selected in the M1 Patch.

Directions:

To load a patch from an instrument directly into the current editor:

(1) Choose *Sngl Edit/Reload*

See Also: [Patch Editor](#)

Midi Quest MFX/VST Settings

The Midi Quest MFX and VST Settings dialog provides all of the settings available for the plugin versions of Midi Quest. The majority of these settings determine how the plugin can be automated by the host program. In addition, this dialog offers a range of display control parameters.

For those of you using Cubase VST, you will find that many of the parameters are disabled. This is normal. The Cakewalk MFX MIDI processing environment offers many more MIDI control options than does VST.

Midi Quest Automation

The most important feature of Midi Quest MFX and Midi Quest VST is that they offer automation capabilities. This gives you a way to automatically send Midi Quest SysX data to your instruments when you wish.

The benefits of this are obvious: you no longer need to switch back and forth between two programs in order to configure your synthesizers; you can tweak your synthesizer settings in your sequencer while your sequencer is running; and you can automatically send SysX data to your instruments and the appropriate place in a song. You can configure Midi Quest and your sequencer once to send data when you need to and then forget about it.

If you have not read the setup instructions for [Midi Quest MFX](#) or [Midi Quest VST](#), you should do so now.

Custom Cakewalk Info

When you are making edits, auditioning patches, or sending files in the Midi Quest MFX editor windows, it is important to remember that all of this MIDI data is being sent out the ports that you have opened using the **MIDI IN Ports** and **MIDI OUT Ports** buttons in the General tab of the MFX Settings dialog.

However, when SysX data is sent automatically using the options discussed below, it is sent using the MIDI ports **opened by Cakewalk**. The SysX data is actually given to Cakewalk and Cakewalk is responsible for transmitting it at the correct time during sequence playback. For this method to work properly you must ensure that Midi Quest MIDI OUT ports are mapped to Cakewalk MIDI OUT ports. To do this, see **Midi Quest Ports -> Cakewalk Ports** below.

Custom Cubase Info

Unlike Cakewalk, the Midi Quest VST plugin is always responsible for sending SysX data because the VST 2.0 spec does not currently support SysX data. As a result of this, Midi Quest can not be tick accurate in sending SysX data, it is only accurate to the size of your sample block. The smaller the sample block size you use, the more accurate the SysX automation will be.

Lets look at each of the available tabs in the Settings Dialog...

General Tab



The General Tab is always available regardless of the type of Midi Quest editor displayed. This dialog allows you to select MIDI ports, map MIDI ports for Cakewalk, control the window display, and specify the tick separation of SysX messages.

Both MFX and VST use a fixed window size to display plugins. The following set of parameter are used to specify the size of the window to use with the window. Changes to these parameters do not take effect until the window has been closed and reopened.

Width specifies the width of the editing window

Height specifies the height of the editing window

Bypass, when checked, specifies that there is to be no automation during playback. The plugin behaves as if it is not there and does not respond to any input from the sequencer.

Dialog Size specifies the default dialog size to use. *Standard* is configured for 1024x768 pixel displays. *Small* is configured for 800x600 pixel or smaller displays. *Medium* is configured for 600x800 pixel displays. *Xtra Large* is configured for displays that are larger than 1024x768 pixels. The setting is parameter is retained from session to session and it is used with each newly created plugin until it is changed to a different value.

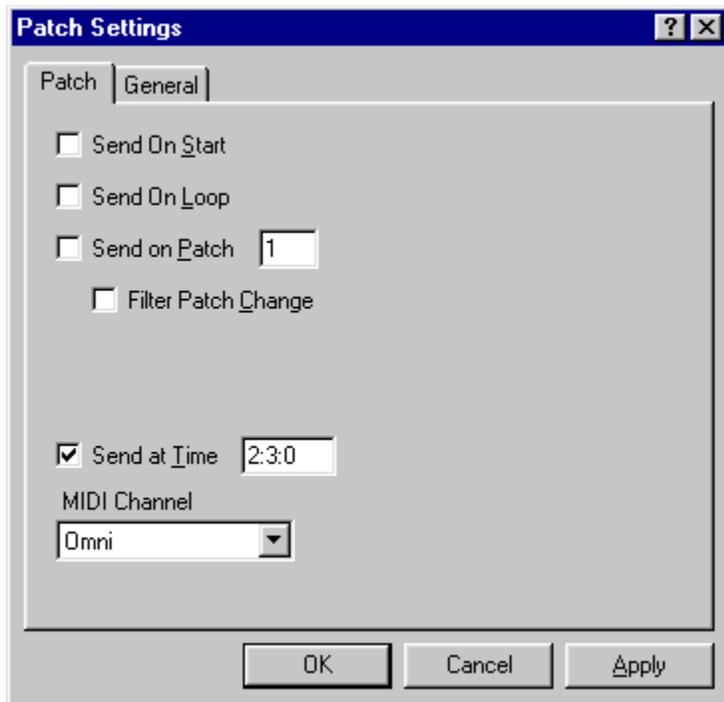
MIDI IN Ports opens a dialog to select the MIDI IN ports to open for MFX or VST. This setting is global, not file specific. For more information on the Midi In Ports dialog, [click here](#).

MIDI OUT Ports opens a dialog to select the MIDI OUT ports to open for MFX or VST. This setting is global, not file specific. For more information on the Midi Out Ports dialog, [click here](#).

Tick Separation specifies the number of additional ticks to delay between back-to-back SysX transmissions

MQ Port -> Cakewalk Port allows you to map Midi Quest port assignments to Cakewalk port assignments so that SysX data stored by Midi Quest can be sent to the correct MIDI port in Cakewalk. **For Midi Quest MFX to work properly, it is important that this map be configured correctly!** For more information in configuring the map, please see the [Configuring Midi Quest MFX for Cakewalk Output](#) section in [Midi Quest MFX](#). A final note, this is a global setting and applies to all Midi Quest MFX plugins, not just the current one. (This configuration is not necessary for VST.)

Patch Tab



The Patch Tab is used with Patch editors. It is used to control when the patch is automatically sent to the instrument. The options are:

Send On Start: if checked, the patch is sent when the sequence is started

Send On Loop: if checked, the patch is sent when the sequence loops

Send on Patch: if checked, the patch is sent when a patch change command is received and patch number matches the number entered in the dialog. The patch number range is 0 through 127.

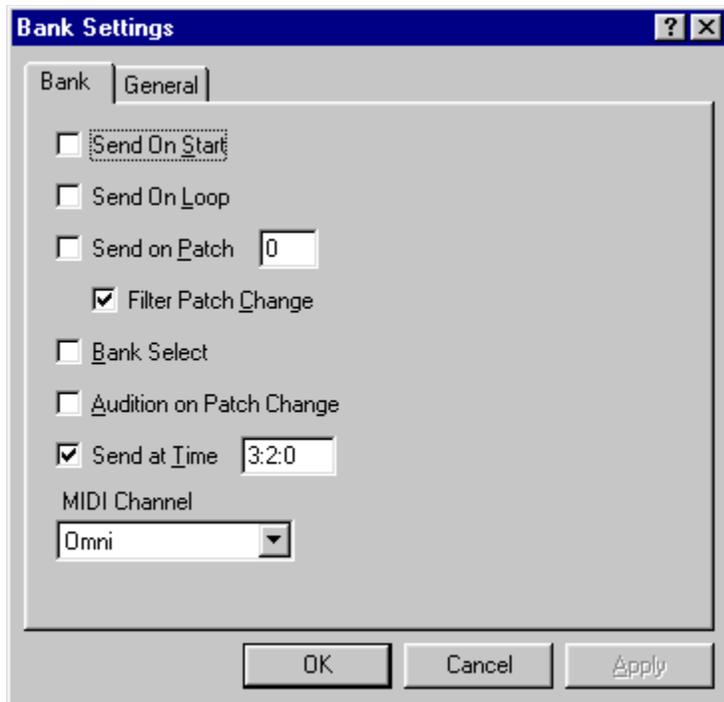
Filter Patch Change: if checked, the patch change that triggers sending the patch is not sent out the MIDI port otherwise it is. (eg. If you enter "15" as the patch number, then when patch 15 is received, the patch data is sent and the patch number is not. All other patch changes are sent to the instrument)

Send at Time: if checked, the patch is sent at the entered time in Measures:Beats:Ticks format

MIDI Channel: sets the MIDI channel on which Patch Change commands will be recognized. Omni recognizes all MIDI channels.

Note: the following Patch Editor functions are not available in the MFX and VST plugin versions of Midi Quest: Move To..., Copy Selected Parameters To..., Morph, Open|Floating Editor, Open|New SysX View, and Fast Tips

Bank Tab



The Bank Tab is used with Bank editors. It is used to control when the Bank or patches in the bank are automatically sent to the instrument. The options are:

Send On Start: if checked, the bank is sent when the sequence is started

Send On Loop: if checked, the bank is sent when the sequence loops

Send on Patch: if checked, the bank is sent when a patch change command is received and patch number matches the number entered in the dialog. The patch number range is 0 through 127.

Filter Patch Change: if checked, the patch change that triggers sending the bank is not sent out the MIDI port otherwise it is. (eg. If you enter "15" as the patch number, then when patch 15 is received, the bank data is sent and the patch number is not. All other patch changes are sent to the instrument)

Bank Select: if checked the bank select number is used in the calculation of patch numbers. This allows access to patches 128 and higher when

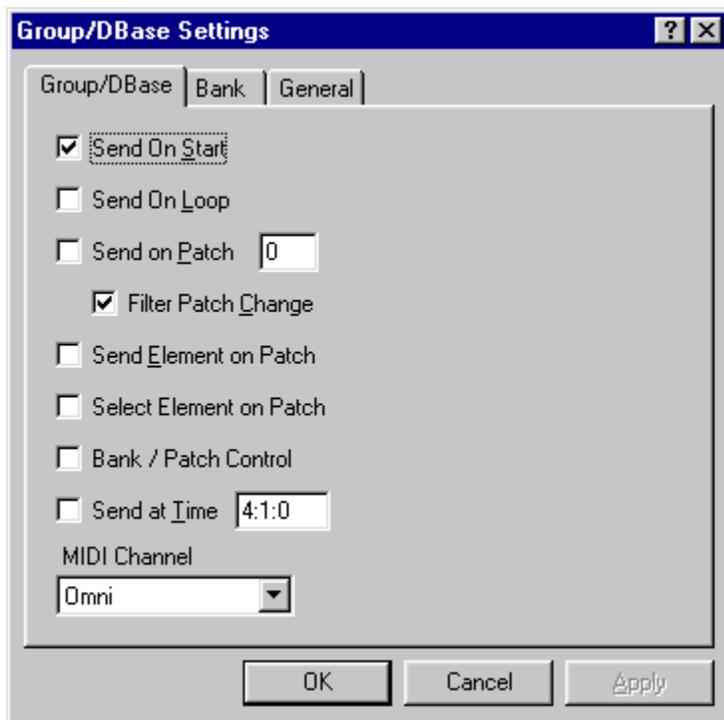
Audition on Patch Change: if checked then instead of sending the bank on receipt of a specific patch change number, a patch is auditioned. In this case, Midi Quest responds to any patch change command. Receiving a patch change of 0 sends the first patch in bank, 1 sends the second and so on. If Bank Select is unchecked then only the first 128 patches in the bank are accessible. If it is checked then a Bank Select of 1 will send patches in the range 128 to 255, Bank Select of 2 will send patches in the range 256 to 383, and so on.

Send at Time: if checked, the bank is sent at the entered time in Measures:Beats:Ticks format

MIDI Channel: sets the MIDI channel on which Patch Change commands will be recognized. Omni recognizes all MIDI channels.

Note: the following Bank Editor functions are not available in the MFX and VST plugin versions of Midi Quest: Move To..., Blend, Gen4, Morph, Mix, Mix All, Edit|Patch, and Edit|Open SysX View

Group/DBase Tab



The Group/DBase Tab is used with both Group and DBase editors. It is used to control when the Group or DBase or part there of is sent to the instrument. The options are:

Send On Start: if checked, the entire group/dbase is sent when the sequence is started

Send On Loop: if checked, the entire group/dbase is sent when the sequence loops

Send on Patch: if checked, **all** elements in the group or dbase are sent when a patch change command is received and patch number matches the number entered in the dialog. The patch number range is 0 through 127.

Filter Patch Change: if checked, the patch change that triggers sending the group/dbase is not sent out the MIDI port, otherwise it is. (eg. If you enter "15" as the patch change, then when patch 15 is received, the group/dbase is sent and the patch number is not. All other patch changes are sent to the instrument)

Send Element on Patch: if checked, the patch number of the patch change command is used to select and send the corresponding entry in the group or dbase. For example, a patch change of 0 selects and sends the first entry in the group, patch change 1 selects and sends the second entry, and so on.

Select Element On Patch: if checked, the patch number of the patch change command is used to select (not send) the corresponding entry in the group or dbase. For example, a patch change of 0 selects the first entry in the group, patch change 1 selects the second entry, and so on. This function is used to select a patch bank in the Group or DBase so that individual patches may be auditioned from it (see the example below).

Bank/Patch Control: if checked, patch changes can be used to select patches from within a bank

Send at Time: if checked, the entire group/dbase is sent at the entered time in Measures:Beats:Ticks format

MIDI Channel: sets the MIDI channel on which Patch Change commands will be recognized. Omni recognizes all MIDI channels.

Group/DBase Automation Example

The Group and DBase editors offer a wider range of automation control for those who wish to take advantage of it. In this dialog there are 3 different options which respond to patch changes. **Send on Patch** will send the entire Group or DBase. **Send Element on Patch Change** will send one element from the Group or DBase where that element is determined by the number of the patch change. **Select Element on Patch Change** uses the patch change number to simply select an element in the Group or DBase.

Select Element On Patch Change actually works in conjunction with **Bank / Patch Control**. When both of these options are set, then it is actually possible to configure the Midi Quest plugin to audition patches from within a bank.

Here is what you need to do to setup for bank auditioning from a Group or DBase:

1. In the Group/DBase tab, check both **Select Element on Patch** and **Bank / Patch Control**.
2. Under the Bank tab, check **Audition on Patch Change**
3. Under the Bank tab, check **Bank Select** if the bank has more than 128 patches (and you wish to access them)
4. Under the Bank tab, check **Filter Patch Change** if you don't want the patch change sent on to the instrument
5. You're finished the configuration so press the OK button.

Here is what you need to do in your sequencer to control auditioning of patches. For this example, we will use an 01/W Group. The group has four elements in the following order: Combi Bank, Patch Bank, Drums, and Global. You need to add these MIDI messages in the track to which the Midi Quest plugin is assigned:

1. **Send a patch change** to select the bank you wish to audition from. To audition from the Patch bank you need to use patch #1 (assuming that the first patch is patch #0). When received, this patch change will select the Patch Bank.
2. **Send a Bank Select** (controller 0) message with a value of 127. This is a special message that the Midi Quest plugin to mean - "forward all patch change messages to the selected bank". When this message is received by the plugin, the plugin knows that any future patch changes and controller messages are to be routed to the bank selected in #1 above.
3. **Send one or more patch changes**. Each patch change will send the associated patch from the patch bank.
4. If you are working with an instrument that has a bank size greater than 128 patches, you can use bank selected commands (controller 0) to select which block of 128 patches you wish to access. A Controller 0 message with a value of 0 allows the patch change command to select from patches 0 - 127. A Controller 0 message with a value of 1 allows the patch change command to select from patches 128 - 255, and so on.
5. To select a different bank in the Group or DBase you need to cancel the forwarding of messages, that

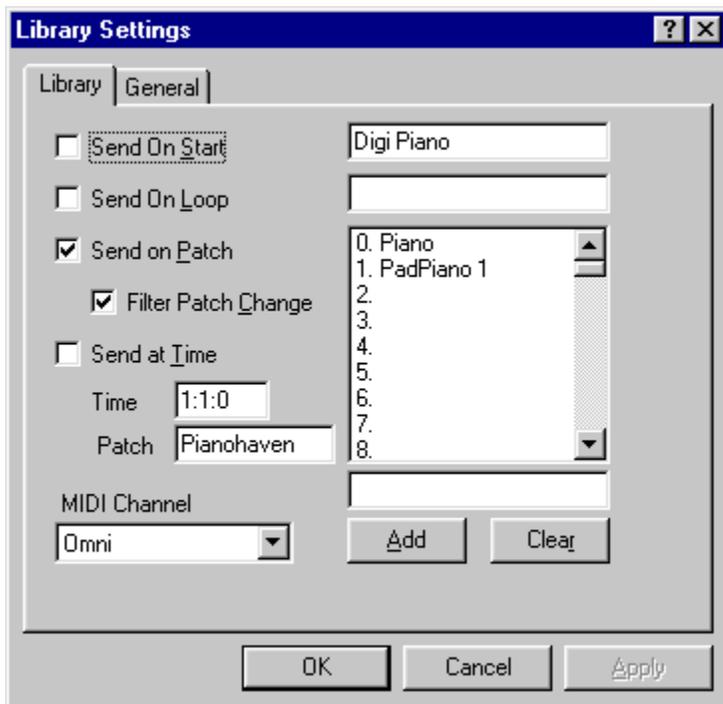
is. cancel the bank select 127 message. This is done with a Bank Select (controller 0) message with a value of 126.

Following is a compilation of the MIDI messages that must be sent to the Midi Quest plugin in order to automate it for auditioning of patches in a bank in a group or dbase.

Message	value 1	value 2	
Patch Change	1		(select the Patch Bank for an 01/W)
Controller	0	127	(route patch changes and controller messages to the bank)
Patch Change	n		(n = number of the patch in the bank to send)
Patch Change	n		(audition as many patches as you wish)
Controller	0	126	(DBase/Group now gets patch changes - select a new bank)
Patch Change	0		(select the Combi Bank for an 01/W)
Controller	0	127	(route patch changes and controllers to the Combi Bank)
Patch Change	n		(n = number of the combination to send)

Note: the following Group and DBase Editor functions are not available in the MFX and VST plugin versions of Midi Quest: Move To..., Open|Editor, and Open|SysX View.

Library Tab



The Library Tab is used with Libraries. It is used to control when the Patches from the library are automatically sent to the instrument. The options are:

Send On Start: if checked, the library is searched for a patch with the corresponding name. If one is

found, the patch is sent each time the sequence is started. The name of the patch is entered in the text area to the right (eg. Trumpet 1).

Send On Loop: if checked, the library is searched for a patch with the corresponding name when the sequence loops. If one is found, the patch is sent when the sequence loops. The name of the patch is entered in the text area to the right. (eg. Trumpet 2)

Send on Patch: if checked, the library is searched for the patch with a name corresponding to the listed names. If a match is found, that patch is automatically sent to the instrument. For example, a patch change #2 would cause the library to be searched for a patch named "Mandolin". If there is a match, that patch is sent to the instrument.

To create the list of patch names, select a patch number, enter the name of the patch in the text area below and press the "Add" button. To remove a patch from the list, select the patch name and press

Filter Patch Change: if checked, any patch change that triggers transmission of a patch is not sent out the MIDI port, otherwise it is. (eg. If you enter "15" as the patch change and there is a match in the library for the assigned patch name, then when patch 15 is received, the patch data is sent and the patch number is not.)

Send at Time: if checked, the library is searched for the corresponding patch name. If there is a match, the patch is sent to the instrument. In the above example, if "Send at Time" were checked, the patch "Oboe" would be sent at Measure 1, Beat 1, Tick 0.

MIDI Channel: sets the MIDI channel on which Patch Change commands will be recognized. Omni recognizes all MIDI channels.

Note: the following Patch Editor functions are not available in the MFX and VST plugin versions of Midi Quest: Move To..., Blend, Gen4, Morph, Mix All, Build Bank, Open|Editor, and Open/SysX View.

See also: [Midi Quest MFX](#)

See also: [Midi Quest VST](#)

Control Info (Patch Editor)

Purpose:

Displays a dialog listing the id, minimum, and maximum values of the currently selected control.

This information is not required for the operation of Midi Quest itself. However, it is useful for automated editing using Infinity, Sound Quest's object oriented MIDI control language.

Directions:

To obtain information on a particular control:

- (1) Select the control
- (2) Choose *Sngl Edit/Organize/Control Info...*
- (3) A dialog is displayed with the information

See Also: [Patch Editor](#)

