WinSysEx 2.0 - Copyright 1993 by Don Strenczewilk

General Topics

Using WinSysEx

Commands

Reference

Introduction To WinSysEx

What's It Do?

WinSysEx is a Shareware program that lets you create dialog boxes that send system exclusive messages from your computer to a MIDI device or Window's clipboard. You enter groups of system exclusive messages into WinSysEx's editor, giving each group a descriptive name. WinSysEx creates a SysEx Dialog. The **SysEx Dialogs** listbox, located in WinSysEx's main window, lists your dialogs and allows you to pick one to run or edit. Double-click on a name in the list to run the SysEx dialog. When you click the SysEx dialog's **Send** menu item, WinSysEx sends the group of system exclusive messages out the MIDI port.

WinSysEx Features

- Lets you easily define input-controls: Spin controls, Text controls, Scrollbars, Check boxes, and drop-down Combo boxes.
- Creates dialog boxes, automatically inserting your input-controls.
- Supports binary operations: AND, OR, PLUS, MINUS, SHIFT LEFT, SHIFT RIGHT.
- Lets you enter numbers in hexadecimal, decimal, or binary format.
- Intermediate results of calculations are performed with 32-bit signed accuracy.
- Supports complex checksums calculations at runtime.
- Allows Local and Global variables.
- Lets you save your dialogs to disk.

WinSysEx's Limitations

WinSysEx is not a bulk data dump manager. It has no provision to *receive* MIDI data; it only *transmits*. A patch editor and librarian created with WinSysEx would probably not be as elegant as a program dedicated to that purpose. It won't be as elegant because of the way WinSysEx automatically positions the input-controls in the SysEx dialogs. On the other hand, you can define input-controls very quickly and easily.

Data Storage

After you edit a group of SysEx dialogs, you can save them to a *.WSX file and reload the file at any time.

You can also save the SysEx dialogs to *.WSD data files. When you work with WSD files, you can save and restore the values of input-controls as you change them while the dialogs run.

System Requirements

To use WinSysEx, you need:

- Microsoft Windows version 3.1 or greater
- A MIDI interface or sound card installed in your computer
- A Windows Multimedia driver for the MIDI interface
- A synthesizer or other midi device that can be controlled with system exclusive messages

See Also:

Setting your MIDI ports
About WinSysEx files

Acknowledgments

WinSysEx is Copyright 1993 Don Strenczewilk, All Rights Reserved.

WinSetup is Copyright 1993 Money Smith Systems Inc, All Rights Reserved.

Microsoft Windows is Copyright 1993 Microsoft Corp, All Rights Reserved.

CakeWalk for Windows is Copyright 1991, Greg Hendershott, Twelve-Tone Systems.

The File Menu

The file menu has items that let you load and save files and to quit the program. When a command requires removal of the current file from memory, you are asked whether you want to save it or not.

New Create a new unnamed WSX file.

Open Load a WSX or WSD file. You can also load a file by dragging a file from Window's File Manager to WinSysEx's main window with your mouse.

Save Save the current file. If the current file is unnamed, activate the **Save as** dialog. You can also select this command by typing [Ctrl-S].

Save as Lets you save the current file under a new name or type. You can save a WSX file as a WSD file only if the WSX file compiles with no errors. Since WSD files do not contain any source code, you cannot save them as WSX files.

Exit Closes all running dialogs and shuts down WinSysEx.

See Also:
About WinSysEx Files

Dialogs Menu

The items on this menu pertain to the SysEx Dialogs displayed in the **SysEx Dialogs** list box.

Note: If the current file is a *.WSD file, the Edit, Delete, Rename, and Search items on the menu are grayed. Those commands relate only to *.WSX files.

Run Run the highlighted SysEx dialog.

Edit Edit the highlighted SysEx dialog. A short-cut for this is to right click in the list.

Delete Delete the highlighted SysEx dialog.

Rename Rename the highlighted SysEx dialog.

Send All To MIDI Out
Sends all system exclusive messages in all SysEx dialogs in the current file out the MIDI port.

Search Lets you search for a particular string of characters in the source code. After presenting a dialog that lets you enter a word to search for, it searches through the SysEx Dialogs until it finds one that contains the word you entered. If the word is found, the SysEx Dialog is selected in the list.

Cascade Cascade all running SysEx dialog windows.

Close all Close all SysEx dialog windows.

See Also:
About WinSysEx files
Running SysEx Dialogs

Control states Menu

This menu is displayed only when you're working with *.WSD files.   The items on this menu pertain to the items in the Control states listed in the **Control states** listbox.

Save as Save the settings of all input-controls as a control state.   A dialog pops up that lets you name the control state.

Restore Set all input-control values to the values in the highlighted control state.

Delete Delete the highlighted Control state.

MIDI Menu

The MIDI menu has commands that let you select MIDI input and output devices. The devices you select in the MIDI Setup Dialog and the checked/unchecked state   of the MIDIIN ON/OFF menus are maintained in WINSYSEX.INI.

Setup Opens a dialog that lets you select MIDI devices.

MidiOut ON The MPU-401 driver supplied with MS-Windows allows   only one program to use the MIDI OUT device at one time.   If you're using that driver and you want to run a program in another window that needs   to use the same MIDI OUT device as the one you selected from the MIDI|Setup dialog, uncheck this item.

When you check this item: WinSysEx opens the MIDI OUT device and keeps   it open until you uncheck this item, or close WinSysEx.

When you uncheck this item: WinSysEx opens the MIDI OUT port only when it's   ready to send system exclusive messages, and closes the port immediately after the messages are sent.

MidiIn ON When you check this menu item WinSysEx opens the MIDI input   device. If your keyboard is connected to the MIDI IN port of your PC MIDI interface card and   you want the notes you play to pass through to the MIDI OUT port, check this item.

Most Windows MIDI device drivers don't allow multiple programs to simultaneous use the   same MIDI IN device. Therefore, don't check this item if you want to run WinSysEx   simultaneously with another program that uses the same MIDI IN device as WinSysEx.

This item will be disabled if you don't have any MIDI input devices   installed in your machine.

See Also:
MIDI Setup Dialog

MIDI Setup Dialog

Activate the MIDI Setup Dialog from the **MIDI|Setup** menu item in WinSysEx's main window. It contains two list boxes:

MIDI In Select the MIDI device to use for input from your MIDI keyboard/controller. This list box won't appear if you don't have any MIDI input devices installed in your machine.

MIDI Out Select the MIDI device that you want WinSysEx to use to send sysex messages.

See Also:
The MIDI Menu

Running SysEx Dialogs

After you create SysEx Dialogs with the SysEx Editor , you can run them by double clicking on their names in the **SysEx Dialogs** list box or by selecting **Dialogs|Run** from WinSysEx's menu.

The SysEx Dialog Menu

Every SysEx Dialog has a menu bar with three items:

Edit This drop down menu has five items:

Copy text to clipboard. WinSysEx recalculates all the code in the dialog, converts the bytes to hex strings, and puts the results in MS-Window's clipboard. You can then paste into the text to another program, like Window's NOTEPAD.EXE. This is useful for debugging since you can see if your source code is doing what you expect. It's also useful for pasting into sequencers that can accept system exclusive messages messages in that format, such as CakeWalk for Windows.

Send to MIDI out. When you click on this item, WinSysEx recalculates all the code in the dialog, and sends each system exclusive message in the dialog to MIDI OUT. This can also be accomplished by clicking the right mouse button in the dialog.

Save Sysex bytes as. Select this item to save the sysex bytes to a file.

Reset default values. Reset all the input-controls in the dialog to their default values.

Exit. Select this item to close the dialog

Options This drop down menu has three items that let you set options for the SysEx dialogs. Your selections are saved in WINSYSEX.INI.

Font. Lets you select the font and size to use in the dialog. Because the size of the controls and text in the sysex dialogs are dependant on the font size, changing the font size changes the size of the entire dialog.

Highlight color. Select the color to draw input-controls whose value is not set to the default.

Send MIDI whenever input changes. Check this menu item if you want to send the SysEx messages to MIDI OUT whenever you modify an input-control. Uncheck this item if you want to send the SysEx messages manually. *See **Edit|Send to MIDI Out** above.*

Help This menu has items to display this help topic and the *About* dialog.

Navigating The Input Controls

These commands describe how to navigate the controls. Note that when you resize the dialog, a scroller appears on the right side of the dialog that you can use to scroll controls into view that are outside the viewing area.

Action	Keyboard	Mouse
Next	Down Arrow or Tab	Click on the control
Previous	Up arrow or Shift-Tab	Click on the previous control
Next Screen	PgDn	Click the Scroller on the right side of the dialog
Previous Screen	PgUp	

Spin Input-Controls

Besides simply typing the value you want in the spin control, you can use the following methods to change its value:

Action	Keyboard	Mouse
increment	+	Click on up arrow
Decrement	-	Click on down arrow
Reset default	*	N/A

Text Input-Controls

Text controls are simple edit controls, used for entering patch names and such. To edit a text control, simply type the name as you do with any other Windows edit control.

Scrollbars

Action	Keyboard	Mouse
Increment	Right arrow	Click on right arrow
Decrement	Left arrow	Click on left arrow
Reset default	*	N/A

Checkboxes

Checkboxes have two possible states: Unchecked and Checked.

Action	Keyboard	Mouse
Toggle	Spacebar	Click on it
Reset default	*	N/A
Check	+	Click on it
Uncheck	-	Click on it

Comboboxes

To make a selection from a Combobox, you must first drop its list down.

Action	Keyboard	Mouse
Drop	Enter or Right Arrow	Click on it
Raise	Enter	Click on an item
Reset default	*	N/A

Byte Conversion Table

Dec	Hex	Binary	Dec	Hex	Binary
0	00	00000000	127	7F	01111111
1	01	00000001	128	80	10000000
2	02	00000010	129	81	10000001
3	03	00000011	130	82	10000010
4	04	00000100	131	83	10000011
5	05	00000101	132	84	10000100
6	06	00000110	133	85	10000101
7	07	00000111	134	86	10000110
8	08	00001000	135	87	10000111
9	09	00001001	136	88	10001000
10	0A	00001010	137	89	10001001
11	0B	00001011	138	8A	10001010
12	0C	00001100	139	8B	10001011
13	0D	00001101	140	8C	10001100
14	0E	00001110	141	8D	10001101
15	0F	00001111	142	8E	10001110
16	10	00010000	143	8F	10001111
17	11	00010001	144	90	10010000
18	12	00010010	145	91	10010001
19	13	00010011	146	92	10010010
20	14	00010100	147	93	10010011
21	15	00010101	148	94	10010100
22	16	00010110	149	95	10010101
23	17	00010111	150	96	10010110
24	18	00011000	151	97	10010111
25	19	00011001	152	98	10011000
26	1A	00011010	153	99	10011001
27	1B	00011011	154	9A	10011010
28	1C	00011100	155	9B	10011011
29	1D	00011101	156	9C	10011100
30	1E	00011110	157	9D	10011101
31	1F	00011111	158	9E	10011110
32	20	00100000	159	9F	10011111
33	21	00100001	160	A0	10100000
34	22	00100010	161	A1	10100001
35	23	00100011	162	A2	10100010
36	24	00100100	163	A3	10100011
37	25	00100101	164	A4	10100100
38	26	00100110	165	A5	10100101
39	27	00100111	166	A6	10100110
40	28	00101000	167	A7	10100111
41	29	00101001	168	A8	10101000
42	2A	00101010	169	A9	10101001
43	2B	00101011	170	AA	10101010
44	2C	00101100	171	AB	10101011
45	2D	00101101	172	AC	10101100
46	2E	00101110	173	AD	10101101
47	2F	00101111	174	AE	10101110
48	30	00110000	175	AF	10101111
49	31	00110001	176	B0	10110000
50	32	00110010	177	B1	10110001

51	33	00110011	178	B2	10110010
52	34	00110100	179	B3	10110011
53	35	00110101	180	B4	10110100
54	36	00110110	181	B5	10110101
55	37	00110111	182	B6	10110110
56	38	00111000	183	B7	10110111
57	39	00111001	184	B8	10111000
58	3A	00111010	185	B9	10111001
59	3B	00111011	186	BA	10111010
60	3C	00111100	187	BB	10111011
61	3D	00111101	188	BC	10111100
62	3E	00111110	189	BD	10111101
63	3F	00111111	190	BE	10111110
64	40	01000000	191	BF	10111111
65	41	01000001	192	C0	11000000
66	42	01000010	193	C1	11000001
67	43	01000011	194	C2	11000010
68	44	01000100	195	C3	11000011
69	45	01000101	196	C4	11000100
70	46	01000110	197	C5	11000101
71	47	01000111	198	C6	11000110
72	48	01001000	199	C7	11000111
73	49	01001001	200	C8	11001000
74	4A	01001010	201	C9	11001001
75	4B	01001011	202	CA	11001010
76	4C	01001100	203	CB	11001011
77	4D	01001101	204	CC	11001100
78	4E	01001110	205	CD	11001101
79	4F	01001111	206	CE	11001110
80	50	01010000	207	CF	11001111
81	51	01010001	208	D0	11010000
82	52	01010010	209	D1	11010001
83	53	01010011	210	D2	11010010
84	54	01010100	211	D3	11010011
85	55	01010101	212	D4	11010100
86	56	01010110	213	D5	11010101
87	57	01010111	214	D6	11010110
88	58	01011000	215	D7	11010111
89	59	01011001	216	D8	11011000
90	5A	01011010	217	D9	11011001
91	5B	01011011	218	DA	11011010
92	5C	01011100	219	DB	11011011
93	5D	01011101	220	DC	11011100
94	5E	01011110	221	DD	11011101
95	5F	01011111	222	DE	11011110
96	60	01100000	223	DF	11011111
97	61	01100001	224	E0	11100000
98	62	01100010	225	E1	11100001
99	63	01100011	226	E2	11100010
100	64	01100100	227	E3	11100011
101	65	01100101	228	E4	11100100
102	66	01100110	229	E5	11100101
103	67	01100111	230	E6	11100110
104	68	01101000	231	E7	11100111
105	69	01101001	232	E8	11101000
106	6A	01101010	233	E9	11101001
107	6B	01101011	234	EA	11101010

108	6C	01101100
109	6D	01101101
110	6E	01101110
111	6F	01101111
112	70	01110000
113	71	01110001
114	72	01110010
115	73	01110011
116	74	01110100
117	75	01110101
118	76	01110110
119	77	01110111
120	78	01111000
121	79	01111001
122	7A	01111010
123	7B	01111011
124	7C	01111100
125	7D	01111101
126	7E	01111110
127	7F	01111111
128	80	10000000
235	EB	11101011
236	EC	11101100
237	ED	11101101
238	EE	11101110
239	EF	11101111
240	F0	11110000
241	F1	11110001
242	F2	11110010
243	F3	11110011
244	F4	11110100
245	F5	11110101
246	F6	11110110
247	F7	11110111
248	F8	11111000
249	F9	11111001
250	FA	11111010
251	FB	11111011
252	FC	11111100
253	FD	11111101
254	FE	11111110
255	FF	11111111

Files Included and Created by WinSysEx

All files included with WinSysEx are located in the directory that WinSysEx was installed. No files are written to your WINDOWS directory. To uninstall WinSysEx, simply delete all the files in its directory, and any WSX, WSD, and INI files you created with it.

Files Included With WinSysEx

winsysex.exe	The main program executable file
winsysex.hlp	WinSysEx's help file
genmid.ini	A Combobox data file containing general MIDI voice, drum, and note names.
sample.wsx	Sample source code with examples of input controls and modifiers.
***.WSX**	Source code examples
***.INI**	Combobox data files used in source code examples.
order.doc	Printable order form.
vendor.doc	Information for distributors, disk vendors, sysops, etc.
file_id.diz	BBS file description.
readme.txt	Contains information about the program version enhancements, bug fixes, and last minute information that did not make it into the documentation and help file.

Files Created By WinSysEx

winsysex.ini	Contains user definable info used by WinSysEx.
***.WSX**	Source code files
***.WSD**	Compiled Data files.

See Also:

About WinSysEx Data Files
The File Menu

About WinSysEx Data Files

With WinSysEx, you can save and load two types of files.   The two file types can be identified on disk by their three letter filename extensions: WSX or WSD.

WSX files After you create SysEx dialogs, you can save as a *.WSX file and reopen the file   at any time to edit them.

WSX files hold only source code. If your SysEx Dialogs contain combobox input-controls, the combo box data files must be present in the directory that WINSYSEX.EXE   or the WSX file resides or the WSX file will not load correctly.

WSD files You can also save SysEx dialogs as *.WSD files. These files contain compiled SysEx dialogs. In other words, they store SysEx Dialog information in a binary format.   Therefore, you can not edit the SysEx Dialogs in these files.

WSD files let you store *control states*, which let you save the state of all the input-controls   in the file after you've altered them while running the SysEx dialogs. You can use control states,   for example, to store synthesizer patch parameters. When you load a WSD file, WinSysEx's main dialog expands to display the **Control States** listbox.

If your SysEx Dialogs include Comboboxes, their data is stored directly in the WSD files.

See Also:

The File Menu
The Control states Menu

Contacting the author

Please feel free to contact me at any time via CompuServe or US Mail with any questions or suggestions you have. For the quickest response, contact me via CompuServe Mail and you can ususally expect a reply within 24 hours.

Author:	Don Strenczewilk
CompuServe ID:	72617,132
Mailing address:	72 Knapp Ave.
	Rochester, NY 14609-1127

If you write a WSX file for a particular synthesizer or other MIDI device, please send it to me and I'll forward it to anyone else who wants it. By the same token, before you write a WSX file for your synthesizer from scratch, check with me to see if I already have a WSX file that might help.

Association of
Shareware
Professionals
Member

WinSysEx is produced by a member of the Association of Shareware Professionals (ASP). ASP wants to make sure that the shareware principle works for you. If you are unable to resolve a shareware-related problem with an ASP member by contacting the member directly, ASP may be able to help. The ASP Ombudsman can help you resolve a dispute or problem with an ASP member, but does not provice technical support for members' products. Please write to the ASP Ombudsman at 545 Grover Road, Muskegon, MI 49442-9427 or send a CompuServe message via Cmail to ASP Ombudsman 70007,3536.

Definition of Shareware

Shareware distribution gives users a chance to try software before buying it. If you try a Shareware program and continue using it, you are expected to register. Individual programs differ on details -- some request registration while others require it, some specify a maximum trial period. With registration, you get anything from the simple right to continue using the software to an updated program with printed manual.

Copyright laws apply to both Shareware and commercial software, and the copyright holder retains all rights, with a few specific exceptions as stated below. Shareware authors are accomplished programmers, just like commercial authors, and the programs are of comparable quality. (Inboth cases, there are good programs and bad ones!) The main difference is in the method of distribution. The author specifically grants the right to copy and distribute the software, either to all and sundry or to a specific group. For example, some authors require written permission before a commercial disk vendor may copy their Shareware.

Shareware is a distribution method, not a type of software. You should find software that suits your needs and pocketbook, whether it's commercial or Shareware. The Shareware system makes fitting your needs easier, because you can try before you buy. And because the overhead is low, prices are low also. Shareware has the ultimate money-back guarantee -- if you don't use the product, you don't pay for it.

See Also:
Disclaimer - Agreement
Registering WinSysEx
Order Form
Registering via CompuServe SWREG
Contacting the author

Disclaimer - Agreement / WinSysEx - © Copyright 1993 by Don Strenczewilk

Users of WinSysEx must accept this disclaimer of warranty: "WinSysEx is supplied as is. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The author assumes no liability for damages, direct or consequential, which may result from the use of WinSysEx." WinSysEx is a "shareware program" and is provided at no charge to the user for evaluation. Feel free to share it with your friends, but please do not give it away altered or as part of another system. The essence of "user-supported" software is to provide personal computer users with quality software without high prices, and yet to provide incentive for programmers to continue to develop new products. If you find this program useful and find that you are using WinSysEx and continue to use WinSysEx after a reasonable trial period, you must make a registration payment of $39 to Rochester Heritage. The $39 registration fee will license one copy for use on any one computer at any one time. You must treat this software just like a book. An example is that this software may be used by any number of people and may be freely moved from one computer location to another, so long as there is no possibility of it being used at one location while it's being used at another. Just as a book cannot be read by two different persons at the same time.

Commercial users of WinSysEx must register and pay for their copies of WinSysEx within 30 days of first use or their license is withdrawn. Site-License arrangements may be made by contacting Rochester Heritage.

Anyone distributing WinSysEx for any kind of remuneration must first contact Rochester Heritage at the address below for authorization. This authorization will be automatically granted to distributors recognized by the (ASP) as adhering to its guidelines for shareware distributors, and such distributors may begin offering WinSysEx immediately (However Rochester Heritage must still be advised so that the distributor can be kept up-to-date with the latest version of WinSysEx.).

You are encouraged to pass a copy of WinSysEx along to your friends for evaluation. Please encourage them to register their copy if they find that they can use it. All registered users receive a copy of the latest version of the WinSysEx system.

See Also:
Definition of Shareware
Registering WinSysEx
Order Form
Registering via CompuServe SWREG
Contacting the author

Registering WinSysEx

You can register by filling out an order form and   sending it along with a check. Or, If you have a CompuServe account, you can register via CompuServe's SWREG service.

What You Get For Registering

- When your order arrives, I'll send your serial number via CompuServe Mail (if you have a CompuServe account) to immediately eliminate the opening and closing *About* dialogs.
- Via US Mail, a laser-printed manual and a disk with the latest version of the program.
- WSX source code for all supported synths.
- Unlimited support via CompuServe or US Mail
- I'll notify you whenever a major upgrade is released.

See Also:
Definition of Shareware
Disclaimer - Agreement
Order Form
Registering via CompuServe SWREG
Contacting the author

WinSysEx Order Form

Name: ____________________________

Address: ____________________________

____

____

CompuServe UserID: _________________

ITEM	PRICE	UNITS	AMOUNT
WinSysEx	$39.00	_____	______
If you don't live in the U.S., Canada, or Mexico, add $5.00 for mailing.			______
Sales Tax (New York State residents only) County____________ Rate_______%		Tax:	______
		TOTAL:	______

Select **File|Print** from the menu to print this order form,
and send the form and your payment to:

Rochester Heritage
72 Knapp Ave.
Rochester, NY 14609-1127

Include any comments or suggestions you have about the program.

Thank you very much for the order!

Ordering WinSysEx Through CompuServe's SWREG

If you have a CompuServe account, you can easily order WinSysEx via CompuServe's SWREG service. When you register via SWREG, your CompuServe account will be automatically charged for the program. The cost of the program is $39.00. An additional $5.00 is charged for shipping outside The United States.

To register via SWREG:

1 Log on to CompuServe with your communication program.

2 Type **GO SWREG** at any prompt.

3 Follow the prompts to register a program.

4 When asked for the program identification number, enter **1458**.

When CompuServe notifies me that you have registered, I'll send you a serial number via CMail that will register your current copy and eliminate the opening and closing dialogs, and then process your order.

See Also:

Definition of Shareware
Disclaimer - Agreement
Registering WinSysEx
Order Form
Contacting the author

Glossary

These are terms used in various sections of Help. Click on one to see a brief description.

& AND operator
| OR operator
+ PLUS operator
- MINUS operator
< SHIFT LEFT operator
> SHIFT RIGHT operator
*.WSX
*.WSD
Author
Binary
Byte
Checksum
CompuServe
Control states
Decimal
Globals
Hex
Input-Control
Main Dialog
MIDI
Operator
Shareware
SWREG
SysEx Dialogs
SysEx Editor
System Exclusive Messages

The AND operator &

ANDs two values. For example, "10&01" will AND the two values,   for the result "00"

See Also:

Using Modifiers

The OR operator |

ORs two values. For example, "10|01" will OR the two values for the result "11"

See Also:

Using Modifiers

The Plus operator +

Adds two values. For example, "10+01" adds the two values, resulting in "11"

See Also:

Using Modifiers

The Minus operator -

Subtracts one value from another. "10-01" subtracts 01h from 10h resulting in 0Fh

See Also:

Using Modifiers

The Shift Left operator <

Shifts the bits in a value left. "10<01" shifts 10h left by 01h bits resulting in 20h.

See Also:

Using Modifiers

The Shift Right operator >

Shifts the bits in a value right. "10>01" shifts 10h right by 01h bits resulting in 08h.

See Also:

Using Modifiers

*.WSD

Refers to WinSysEx compiled files, which may also contain control states. **WSD** is the file extension used for these files.

See Also:
About WinSysEx Files
Control states menu

*.WSX

Refers to WinSysEx source code files. **WSX** is the file extension used for these files.

See Also:

About WinSysEx Files

WinSysEx's Author

WinSysEx's author is Don Strenczewilk (CIS 72617,132)

See Also:

Contacting the Author

Binary numbers

Binary numbers are a base-2 number system, which consists of the digits 0 and 1. You can enter binary numbers in SysEx dialogs by following the number with a "b", like 1001001b

See Also:

Byte Conversion Table

Byte

A byte consists of 8 bits. For our purposes, a byte is equal to 1 unit of MIDI system exclusive message data. Byte values range from 0 - 255d. When entering bytes in the SysEx Editor, separate the individual bytes of system exclusive messages with either a space, a tab, or a hard carraige return.

CheckSum

Some manufacturers of MIDI devices require that a value be calculated and sent out the MIDI port along with system exclusive messages to verify the integrity of the transmitted MIDI data. That value is called a **CheckSum**

See Also:

Defining Checksums

CompuServe

A worldwide bulletin board service that can be accessed by anyone with a computer and modem.

See Also:

Registering WinSysEx via SWREG

Decimal

Decimal is the base-10 number system, which is the one humans are most familiar with. Each decimal digit can represent values from 0 to 9, and the position or a digit in a decimal number represents a power of 10. To enter a decimal number in WinSysEx's editor, follow it with "d", like 32d. In combobox definition files, WinSysEx assumes the values are entered in decimal format, so the "d" is not necessary.

See Also:

Byte Conversion Table

Globals

You can create a special SysEx Dialog named "Globals" to hold variables that can be be referenced by any other SysEx Dialog in the file. For example, if many of the defined in the other SysEx Dialogs require a channel number, you can define it in **Globals**:

```
@KChan.ED("DW8000 channel (1-16)", 1, 16, 1)-1
```

That will create the global variable, *KChan*, and can be referenced in other dialogs in the file like so:

```
F0 42 @KChan 03 41 10 .SB("Resonance", 0,31,0) F7
```

Globals is the only SysEx Dialog that does not have to contain complete system exclusive messages.

Note: You can only create one **Globals** dialog per file.

Hex (short for Hexidecimal)

The hexidecimal number system is based on the number 16. Each hex digit can represent values from 0 to 15, and the position or a digit in a decimal number represents a power of 16. Since decimal digits only represent the values 0 through 9, the letters **A** through **F** are used to represent **10** through **15**. In WinSysEx's editor, it is assumed that all numbers are in hexidecimal format, so, you simply enter the numbers. The letters **A** through **F** must be entered in upper case. Otherwise, WinSysEx might think you are entering a number in a different format.

In combo box definition files, WinSysEx assumes the values are entered in decimal format, but you can enter numbers in hexidecimal or binary format by following the number with "h" or "b", respectively.

See Also:

Byte Conversion Table

Input-controls

Controls that you can insert into SysEx Dialogs to input values while the dialogs run. WinSysEx supports the following controls:

Spin Controls
Text Controls
Scrollbar Controls
CheckBox Controls
ComboBox Controls

Main Dialog

Refers to WinSysEx's main window.

MIDI

Musical **I**nstrument **D**igital **I**nterface. A standard protocol for transferring musical data between electronic devices.'

Operator

An operator is a symbol placed between two values which tells WinSysEx to perform a calculation on the two values to give a single value result.

See Also:

Using Modifiers

Control states

Control states hold a setting for each input-control in the SysEx Dialogs . They provide a way to save and restore the state of the input-controls. *Note:* The Control states menu and listbox are available only when you're working with compiled *.WSD files.

See Also:

Control states Menu

Shareware

Computer software that you can try before you buy.

See Also:

Registering WinSysEx

SWREG

A service provided by CompuServe Information Service that lets users register ShareWare programs online. WinSysEx's   SWREG identification number is **1458**

See Also:

Ordering WinSysEx through SWREG

SysEx Dialogs

Short for **System Exclusive Message Dialogs**.

They are the dialogs you create with the SysEx Editor . The SysEx Dialogs are listed in WinSysEx 's **SysEx Dialogs** listbox and can be run by double clicking the mouse on one, or run from the **Dialogs|Run menu**.

See Also:

Running SysEx Dialogs
SysEx Dialog Editor Reference

SysEx Editor

Short for **System Exclusive Message Dialog Editor**.

Activated from the **Dialogs|Edit** and **Dialogs|New** menu or by right-clicking inside of the SysEx Dialogs List Box, you use it to enter system exclusive messages   to create SysEx Dialogs .

See Also:

Creating SysEx Dialogs
SysEx Dialog Editor Reference

System Exclusive Message

A MIDI message whose meaning is defined independently by each manufacturer of MIDI devices. System exclusive messages consist of a series of individual bytes. The first byte of a system exclusive message is always **F0**h and the last is **F7**h

See Also:

Creating SysEx Dialogs
SysEx Dialog Editor Reference

SysEx Dialog Editor Reference

Using The SysEx Editor

Launch the SysEx Editor by selecting **Dialog|New** or **Dialog|Edit** from WinSysEx's main window. This section describes the fields and buttons in the SysEx editor.

The Editor Menu:

Edit INI File Launches NOTEPAD.EXE to edit a combo box data file. You can highlight text in the System Exclusive Messages edit box to make NOTEPAD load a file. For example, highlight "DX100.INI" to edit it with NOTEPAD. If you don't highlight anything, NOTEPAD is launched with an *.INI file matching the name of the current file. For example, if you're editing "DX100.WSX", NOTEPAD is launched with "DX100.INI."

Show global variables Pops up a list of global variables you can use for reference.

Font Lets you select the font used to display text in the editor.

Edit fields:

Caption Enter the caption for the dialog you're creating. The caption also displays in WinSysEx's *Sysex Dialogs* list.

System Exclusive Messages This is where you edit the system exclusive messages for the dialog you're creating.

Buttons:

Wrap Click this checkbox to make the System Exclusive Messages edit box wrap text that extends past the right side. When unchecked, a horizontal scrollbar will appear at the bottom of the box that you can use to see text that extends past the right side of the box.

Ok Click the Ok button to save changes and exit the editor. If a sysex dialog of the same name (caption) already exists in WinSysEx's *Sysex Dialogs* list, that dialog is replaced with the edited version. Otherwise, the new dialog is added to the list. The sysex dialog must compile without error before you can save it.

Cancel Exits the editor without saving anything.

Test Compiles and runs the dialog.

See Also:

SysEx Dialog Editor Reference

Creating SysEx Dialogs

This section describes how create SysEx Dialogs. You can browse through the help screens using the browse buttons at the top of the the help window.

First, get the manual for the MIDI device that you want to send the system exclusive messages to. Study the section of the manual that describes the system exclusive messages that the device receives. That is usually near the back of the manual.

Select **File|New** to create a new WSX file.

Design decisions

Decide what dialogs you want the WSX/WSD file to hold; for instance, patch editing parameters for a synthesizer.

Decide how you want to list the SysEx dialogs

For instance, a dialog named "Envelope" might contain the envelope parameters for a patch; another named "Oscillator Parameters" might contain controls to modify oscillator parameters. Try not to create dialogs which contain a collosal number of input-controls. Doing so can quickly deplete Window's USER resources when the dialogs run. Also, those dialogs will have to be scrolled to reveal some input controls. It's easier to manage input-controls when you separate them into smaller groups by either creating more dialogs with descriptive captions or defining separate *contexts* for groups of related controls in a dialog.

Decide how you want to order the input-controls in the dialog

For example, for an "Envelope" dialog, you might define scrollbar controls for Attack, Sustain, Decay, and Release. Sometimes, the requirements of a particular sysex message will dictate the order for you. For example, if all the envelope parameter bytes are contained in a single sysex message, you cannot easily change the order because WinSysEx parses sysex messages sequentially, inserting controls into the dialog as it encounters them.

Detemine whether global variables are needed

For instance, if all of the system exclusive messages require a channel number byte, create a global input-control variable. (See Global Variables below for more information).

Launch the editor

You're now ready to create the SysEx dialogs. Launch the SysEx Editor by selecting **Dialogs|New** from the WinSysEx's menu.

Create the caption

Enter a name for the SysEx dialog you are creating in the **Caption** field. This name will be placed in WinSysEx's **SysEx Dialogs** list box, and will be displayed in the caption bar of the SysEx Dialog when it is run.

Enter the System Exclusive Messages

In the **System Exclusive Messages** field of the SysEx Editor , you define one or more system exclusive messages . System exclusive messages consist of **byte definitions**. Byte definitions may consist of:

Numbers
Input-controls
Checksum calculations

Local Variables

Sometimes you may need to copy the value of an input-control to another location in the same SysEx dialog. To do that, define a local variable.

Global Variables

Other times, you may want to use a value from a input-control in more than one SysEx Dialog. To do that, use a global variable. If you want to see a list of the global variables, select Editor|Show global variables.

Modifiers

Modifiers are numbers, input-controls, local variables, or global variables that that modify the value of other values. A binary operator is placed immediately after the byte that is to be modified, followed by the modifier. The binary operators that WinSysEx supports are OR, AND, PLUS, MINUS, SHIFT LEFT, SHIFT RIGHT.

For instance, a MIDI device's system exclusive message may require a channel number ranging from 0 to 15, but you want an input-control to range from 1 to 16. Or, you might want to make a single byte out of the values of more than one input-control. To do this, use modifiers.

Test the Dialog

Hit the [Test] button to test the dialog. WinSysEx will run the SysEx dialog if there are no errors. If there are errors, it will notify you what the error is and place the cursor at that point.

Format the Controls

Several backslash commands are provided to let you format the position of text in the sysex dialogs.

See Also:

Debugging your dialogs
Byte Conversion Table
Syntax Errors

Defining System Exclusive Message Bytes

Comments

You can document your work in the SysEx editor by inserting comments into the text. Comments are ignored by WinSysEx's interpreter. To enter a comment, type a semi-colon "**;**" at the start of a line, enter the comment, and hit [Enter].

Number Formats

You can define numbers in hexidecimal, decimal, or binary format by specifying **h**, **d**, or **b** after the last digit. If you don't specify the format, WinSysEx assumes it is a hexadecimal number. You must use upper case characters to enter the hexadecimal digits **"A"** through **"F"**. Otherwise, WinSysEx won't know whether "1d" should be interpreted as "1 decimal" or "1D hexadecimal".

Entering Sysex Bytes

Separate the individual bytes of the system exclusive message with one or more spaces, tabs, or hard carriage returns. A simple SysEx dialog consists only of raw hex bytes, such as: **F0 41 10 42 12 40 00 7F 00 F7**. You can create complex expressions which define a single byte, such as: **10h+99d|.SB("A scrollbar",1,128,64)-1**. The individual components of that expression are described later. For now, just realize that it can be done. Raw values may be defined in hexidecimal, decimal, or binary format by specifying "h", "d", or "b" after the last digit. If you don't specify the format, WinSysEx will assume it is a hexidecimal number. The hexidecimal digits **A** through **F** must be entered in upper case. Otherwise, WinSysEx might think you are specifying a different format.

Number Ranges / Accuracy

WinSysEx's editor accepts 32-bit values ranging from -2147483648 to 2147483647. Intermediate calculations are evaluated as 32-bit signed numbers. The low byte (8 bits) of the resulting value of a calculation is placed in the sysex message. If you enter this: **F0 41104212 40 00 7F 42 F7**, the actual sysex message will be interpreted as: **F0 12 40 00 7F 42 F7.** Though may be interpreted as a 32-bit value, only the low 8 bits (byte) will be used in the sysex message when WinSysEx sends it to the MIDI OUT device.

Examples

```
;System exclusive message that resets a Roland SCC-1 sound card:
F0 41 10 42 12 40 00 7F 00 F7

;You can define numerous system exclusive message in a single SysEx Dialog.
;When you run this dialog and click the "Send" menu item, WinSysEx sends
;all of the system exclusive messages you defined out the MIDI port.
F0 41 10 42 12 40 00 7F 00 F7
F0 41 10 42 12 40 00 7F 42 F7
```

See Also:

Byte conversion table
Creating SysEx Dialogs

Defining Input-Controls

Many times, you will want to get input to set values in sysex messages while the SysEx Dialog is running. To allow this, WinSysEx lets you define input-controls. You may choose among 5 kinds of input-controls:

Spin Controls
Text Controls
Scrollbar Controls
CheckBox Controls
ComboBox Controls

Input-controls are identified by a period and a two letter abbreviation. All input-controls definitions require that you enter parameters to describe the values. Input-control parameters are enclosed in parentheses and are separated by commas.

WinSysEx places input-controls in SysEx Dialogs in the order that you define them.

Defining Spin Input-Controls

Declaration

.ED("Prompt", *Lo*, *Hi*, *Default*)

Parameters

Prompt The prompt, which must be enclosed in double quotes, should describe what the value represents. It appears in the SysEx Dialog to the left of the spin control.

Lo Lowest acceptable value. The lowest value you may enter here is -2147483648.

Hi Highest acceptable value. The highest value you may enter here is 2147483647.

Default The initial value of the control when it is first displayed. The value you enter here must be less than or equal to *Hi*, and greater than or equal to *Lo*.

Examples

```
; Define a spin control to adjust the volume
.ED("Volume", 0, 127d, 100d)

; Define a spin control to select a channel. ; Ask for 1 to 16, but use 0 to 15 in the sysex message.
.ED("Channel", 1, 16d, 1)-1

; Ask for tuning parameter ranging from -63d to +63d.
; Resulting byte will actually range from 0 to 127d.
.ED("Tuning", -63d, 64d, 0)+63d
```

Defining Text Input-Controls

Patch name

A text input-control is the only type of control that lets you define more than one sysex value at once. They are handy for entering patch names and such, when the sysex requires a series of characters, because all of the characters can be entered in a single field. Note that the length of the "Initstring" parameter determines the number of characters to define.

Declaration

.TX("Prompt", *Lochar*, *Hichar*, "*Initstring*")

Parameters

Prompt The prompt, which must be enclosed in double quotes, should describe what the text represents. It appears in the SysEx Dialog to the left of the text control.

Lochar The ascii value of the lowest allowable character. Normally, 20h will be used, which is the ascii value for the *space* character.

Hi The ascii value of the highest allowable character. Most synths can accept characters up to 7Fh (127d).

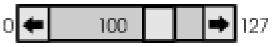
Initstring The initial text to display when the input-control is first displayed. The length of the string you enter here determines the number of bytes to be inserted in the sysex message.

Examples

```
;Create a text control to input a patch name that is 9 characters long. ;Note that the last
parameter, "PATCH     ", is padded with spaces ;to 9 characters and shown in the example
as underlines.
.TX("Patch Name", 20, 7F, "PATCH____")


;The Yamaha DX-100 requires that each character of the voice
;name be sent in a separate sysex parameter message.
;The following example demonstrates how to access the individual
;characters of a text control. For more information about this,
;see Local Variables.

;Create a text control in a dummy message. The dummy message
;will be sent to the synth, but won't do anything.
F0 00 @VName.TX("Voice name", 32d, 127d, "INIT VOICE") F7

;Then, send the indexed chars/bytes to the DX-100, one char per sysex message
F0 43 10 12 77d @VName0 F7
F0 43 10 12 78d @VName1 F7
F0 43 10 12 79d @VName2 F7
F0 43 10 12 80d @VName3 F7
F0 43 10 12 81d @VName4 F7
F0 43 10 12 82d @VName5 F7
F0 43 10 12 83d @VName6 F7
F0 43 10 12 84d @VName7 F7
```

F0 43 10 12 85d **@VName8** F7
F0 43 10 12 86d **@VName9** F7

Defining Scrollbars



Declaration

.SB("Prompt", *Lo*, *Hi*, *Default*)

Parameters

Prompt The prompt, which must be enclosed in double quotes, should describe what the value represents. It appears to the left of the scrollbar in the SysEx Dialog.

Lo The value of the scrollbar when its thumb is all the way to the left. It is the lowest acceptable value. The lowest value you can enter here is -2147483648.

Hi The value of the scrollbar when its thumb is all the way to the left. It is the highest acceptable value. The highest value you may enter here is 2147483647.

Default The initial value of the scrollbar control when it is first displayed. The value you enter here must be less than or equal to *Hi*, and greater than or equal to *Lo*.

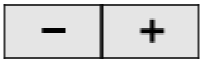
Examples

; Define a scrollbar to adjust the volume
.SB("Volume", 0, 127d, 100d)

; Define a scrollbar to select a channel.
; Ask for 1 to 16, but use 0 to 15.
.SB("Channel", 1, 16d, 1)-1

; Ask for tuning parameter ranging from -63d to +63d. ; Resulting byte will actually range from 0 to 127d.
.SB("Tuning", -63d, 64d, 0)+63d

Defining Checkboxes



Declaration

.CX("Quoted Prompt", *Unchecked, Checked, Default*)

Parameters

Prompt The prompt, which must be enclosed in double quotes, should describe what the checkbox value represents. It appears to the left of the checkbox.

Unchecked The value to assign the sysex data when the box is not checked, or "-". The acceptable range for the value is -2147483648d to -2147483647d

Checked The value to assign the sysex data when the box is checked, or "+". The acceptable range is -2147483648d to -2147483647d

Default The Initial value of the control when it is first displayed. It must be equal to either the Checked or Unchecked parameter.

Examples

;Turn operator 1 on or off.
.CX("Operator 1 ON", 0, 1, 1)

;Using the OR operator **|**, these two check boxes define a single byte, setting individual bits in the byte.
.CX("Op 1 ON", 0, 00001000b, 0)|.CX("Op 2 ON", 0, 00000100b, 0)

Defining Comboboxes

Declaration

.CB("Prompt", "Section", "Filename", *Default*)

Parameters

Prompt The prompt is displayed to the left of the combo box in the SysEx Dialog. You must enclose the prompt in double quotes. The prompt describes what the values in the combobox represent.

Section The section, which must be enclosed in double quotes, describes the section in **Filename** where WinSysEx will find the values and strings for the combo box.

Filename The name of the combobox data file that contains the list of items to be shown in the combobox and their values.

Default The value of the initial text displayed in the combobox. It must be equal to one of the values in the data file.

Creating a combo box data file

Create the data file with a text editor, such as Window's NOTEPAD.EXE. It is wise to name the data file the same name as the WSX file, but substitute the INI extension, and put all the combo box definitions for your entire WSX file in that single file. If you don't supply the full path for the file, WinSysEx will search its home directory, followed by the directory of the WSX file.

The file can contain any number of combo box definitions. The sections are identified by the *Section* parameter, enclosed in square brackets, like:

[The Section]

Enter each item you want to display in the combobox on the lines following the *Section*. Each item definition consists of a number, an "=", and the text to be displayed in the combo box. The default format for the numbers is decimal, but you can follow a number with an "h" to specify the number is in hexidecimal format, or "b" for binary format. The acceptable range for the number is -2147483648d to -2147483647d. You can put comments in the INI file by starting them with a semicolon **;** .

Examples

```
;This combobox definition is in THEFILE.WSX
.CB("The Prompt", "The Stuff", "thefile.ini", 64d)

;This is in the combobox data file, "THEFILE.INI"
[The Stuff]
0=Zero
64=Sixty-four
127=One hundred twenty-seven

;When you run the dialog, the default "Sixty-four" will be displayed in the combo box.
;When you send the data to the  MIDI device, the system exclusive byte that the
;combobox defines will be set to 64.
```

Defining Checksums

Some manufacturers require that their system exclusive messages contain a checksum calculation. WinSysEx calculates checksums for you automatically when you hit the **Send** button while running SysEx Dialogs.

To specify a checksum in the data, surround the byte definitions with [brackets]. WinSysEx calculates the checksum for the bytes between the brackets and inserts a checksum byte in the data at the position of the closing bracket "]". Any type of data can be placed between the brackets, including input-controls, modifiers,   and variables.

Note: WinSysEx currently supports Roland checkum calculations. If you require other types, please inform the author.

Examples

```
;This calculates the checksum for 21 and 03, and insert a checksum byte into the sysex message after the 03.
F0 43 [ 21 03 ] F7

;WinSysEx calculates checksums for controls and modifiers too.
F0 43 [ .SB("Volume", 1, 128)-1 03 ] F7
```

Any type of data can be placed between the brackets, including input-controls, modifiers, and variables.

Defining Local Variables

Sometimes, you may want to use a value from an input-control at one point in the SysEx dialog, and use its value in other system exclusive messages in the **same** SysEx Dialog. To do that, define a local variable.

Create a local variable by preceding an input-control declaration with the "at" sign **@** immediately followed by a string to identify it.

Because Text input-controls hold several byte values, they require special consideration if you need to access the individual bytes. When you define a Text control variable, WinSysEx actually creates a variable for each character (byte) of text. For example, if you declare "@Pat.TX("Text Control", 32, 7F, "Patch")", WinSysEx internally creates 5 variables, appending index numbers to each: @Pat0, @Pat1, @Pat2, @Pat3, and @Pat4, so, you can access each character of the string individually. An example of this is provided in the section that shows how to define text controls. You can still refer to the entire 5 bytes using "@Pat".

Example

;Get an "oscillator" number and name it @osc
F0 43 21 **@osc.ED("Oscillator Number", 0, 15d, 0)** 00 .SB("Volume", 0, 127d, 100d) 00 F7

;Refer to the oscillator number in the same SysEx Dialog
F0 43 21 **@osc** 01 .SB("Resonance", 0, 127d, 64d) 00 F7

WinSys will put one "Oscillator Number" edit control in the SysEx Dialog and use its value wherever it sees the "@osc" identifier in the SysEx Dialog.

;Input a *word* value. Put the high 8 bits in one
;sysex byte, and the low 8 bits in another
F0 43 21 **@frequency.ED("Frequency", 0, 60000d, 30000d)**>8 **@frequency** 00 F7

See Also:

Global variables
Input-controls

Defining Global Variables

Sometimes, you will want to use an input-control to get a value and use its value in one or more other SysEx Dialogs. For instance, you may need to use a channel number for all the sysex messages for a particular synthesizer and you certainly don't want to enter the channel number every time a SysEx dialog pops up. To handle this, WinSysEx recognizes a special SysEx Dialog named "Globals".

Each file can have one SysEx Dialog named "Globals". The "Globals" SysEx Dialog is unique because:

- It doesn't have to contain complete SysEx messages like programs do. It only contains byte definitions.
- The byte definitions can be referenced from other SysEx dialogs in the file.

Example

```
;When you put this in the "Globals" SysEx Dialog, you can use "@chan" to refer to the
;channel number in any SysEx Dialog in the file.
@chan.ED("Channel Number", 1, 16d, 1)-1
```

See Also:

Local variables
Input-controls

Using Modifiers And Operators

WinSysEx editor accepts 32-bit values ranging from -2147483648 to 2147483647. Intermediate results of calculations are also performed with 32 bit values.   Only the low byte (8 bits) of the result value are used in the sysex message.

In some SysEx Dialogs, you might want to have more than one value define a single byte. Or, you may sometimes need to make WinSysEx perform calculations to alter some of the sysex bytes after the user has changed the value of an input-control. WinSysEx lets you do that by supplying several operators. It recalculates the values every time you click the **Send** menu item.

To perform a calculation on two values, put an operator between them. To alter a byte in the data, enter an operator immediately after it,   and the modifier immediately after that. For example:

To alter a byte in the data, enter an operator immediately after it,   and the modifier immediately after that. For example, "21+03|01",   expressed in English: 21 hex plus 03 hex OR 01 hex.   When WinSysEx encounters that when recalculating a sysex message, it will store 21h in the sysex message, add 03h to it, then OR it with 01h.   There are no spaces before or after the OR sign.   When WinSysEx sees a space it expands the sysex message by 1 byte and   positions on the new byte.

Operators that WinSysEx recognizes are:

| Or
& And
+ Plus
- Minus
< Shift left
> Shift right

Example

```
;Subtract 1 from the value returned from a spin control
.ED("Channel Number", 1, 16d, 1)-1

;Use the OR operator | to combine three checkbox values   ;to independently set the low
three bits of a byte.
.CX("Bit 2", 0, 100b, 0)|.CX("Bit 1", 0, 10b, 0)|.CX("Bit 0", 0, 1, 0)
```

Formatting SysEx Dialogs

SysEx dialogs are laid out in rows and columns:

Context column	Prompt column	Input-controls column
row 1		
row 2		
row 3		
...etc.		

If you haven't defined any contexts, the first context column won't appear. The height and width of the rows and columns are determined by the font selected from the Options|Font menu as the dialog runs. Several backslash commands are provided that let you change the appearance of the text in the prompt column in the sysex dialogs.

SysEx Dialog Contexts

\CONTEXT"quoted text"

Use this command to begin a new context for the dialog. When you use contexts, a listbox is created in the context column and contains a list of the context strings you enter as quoted text. The width of the listbox is determined by the selected font and the width of the longest context string. All the input-controls you define after this command are placed in the current context. When you use contexts, you must enter the first context command before defining any input-controls. Otherwise, the input-controls would have no context and not be shown. When the user selects a context from the listbox, the dialog changes to display that context's input-controls.

Text

\TEXT"quoted text"

Use this command to insert text in the dialog's prompt column. The text is justified according to the current justification, which defaults to left-justification. Text or prompts following this command will appear in the same row. To move to the next row, use the \space command, described below.

Horizontal Justification

\PL Left-justify text within the prompt column. This is the default justification. Justification is reset at the start of each new context.

\PR Right-justify text within the prompt column.

\PC Center-justify text within the prompt column.

Horizontal Indentation

\INDENT All text following this command is indented at the current column.

\INDENTC#

Text that follows this command is indented "#" characters from the starting position of the prompt column. The character width is determined by the size of the font, selected from the Options|Font menu as the dialog runs.

Other Backslash commands

\LINE Use this command to insert a line (1 pixel high) that extends the width of the dialog, to separate groups of controls.

\SPACE Use this command to insert a blank line equal to the height of the current font.

Debugging your SysEx Dialogs

If your SysEx Dialogs don't seem to do what you expect, the first thing to do is make sure your MIDI output device is set up correctly from the **MIDI|Select MIDI Ports** menu and verify that your MIDI cords are connected correctly. A quick way to check that is to play a *.MID file using Window's Media Player application.

If everything seems to be set up correctly, find out whether WinSysEx is sending the system exclusive data to your synthesizer. To verify this, select Dialog|New to open the sysex editor with a clean slate. Enter "Simple Test" as the caption. Find a simple sysex message in your synthesizer manual and enter the bytes into the editor. Don't define any input-controls or variables. Click the "Test" button and see if the parameter has changed on your synthesizer. Chances are that you will find that the data is being sent.

At this point, the problem can be narrowed down to the input-controls, modifiers, checksums, or lack of checksums in your sysex dialog. Run the dialog and select **Edit|Copy bytes to clipboard**. Launch NOTEPAD.EXE and select **Edit|Paste** to paste a hexidecimal ascii representation the bytes that WinSysEx is sending out the MIDI port, which you can examine to find out where the problem lies.

Syntax Errors

")" expected A right parenthesis was expected

.SB, .CX, .ED, or .CB expected An input-control statement was expected

= expected on line ... An equal sign was not found after a number in a combobox definition file

@ expected An *at* sign @ was expected

Binary number is too long A binary number is too long. The maximum length of a binary number is 32 digits.

Byte expected A raw hex byte was expected.

Cannot extract number on line ... The parser could not extract a number on a line in a combobox definition file

Combobox filename not found in logged or WinSysEx directory A combobox definition file was not found.

Comma expected A comma was expected in an input-control definition

Data allocation error Not enough memory was allocated for the data. If you see this error, please inform the author

Decimal number expected A decimal digit (0-9) was expected, but something else was there

Default Checkbox value must equal either Checked or Non-checked value The default value you entered for a checkbox definition is not equal to either of the two possible values you entered

Duplicate identifier You defined two @variables with the same name. Variable names must be unique

End of Checksum Expected "]" Right bracket expected. This marks the end of a checksum calculation

First byte is not F0 The first byte in the data stream is not F0 (SOX). All system exclusive messages must begin with F0

Hex digit expected A Hex digit (0-9, A-F) was expected

Hex number too long A hexidecimal number is too long. The maximum length of a hex number is 8 digits

Identifier expected A variable identifier was expected after an @ sign

Insufficient data There isn't enough data to qualify as a system exclusive message'
Invalid binary number A digit in a binary number in invalid. Binary digits may only be 0 or 1

Last byte is not F7 The last byte in the data stream is not F7 (EOX). All system exclusive messages must end with F7

Number expected A number was expected

Operator expected A + - | & < or > was expected (see modifiers)

Prompt too long The length of a quoted prompt is too long

Quoted string expected A "quoted string" was expected

Range error. Default must be between Lovalue and Highvalue When defining an input-control, you entered invalid values

Range error A number encountered in the data was outside of the acceptable range

Start of Checksum Expected "[" A right bracket was encountered, but there is no matching left bracket. Checksum calculations must be enclosed in brackets

Unexpected end of data The end of the text was reached, but the parser hadn't finished parsing the last statement

Unknown Identifier The parser encountered a statement it doesn't understand

Unmatched F0/F7 pairs Every system exclusive must begin with F0 and end with F7. The data you entered contains unmatched pairs, so the data is invalid

When defining multiple messages, F7 must be followed by F0 This one should be self explanatory