

Contents

MIDIBars - January 1, 1994
Version 1.13 - May 12, 1994

Programming MIDIBars

Copyright © Richard S. Huntrods, 1994

MIDIBars is a SHAREWARE program
MIDIBars is copyright © Richard S. Huntrods, 1994
THIS PROGRAM IS NOT FREE!!!

[Copyright](#)

[MIDIBARS Introduction](#)

[Shareware](#)

[Registration](#)

[Programming Introduction](#)

[MIDIBARS.INI Programming:](#)

[MIDIBARS.INI Control Line Format:](#)

[MIDIBARS.INI General Keywords:](#)

[MIDIBARS.INI MIDI Event Keywords:](#)

[MIDIBARS.INI Annotated Examples:](#)

Copyright

MIDIBars - January 1, 1994

Version 1.00 Release Date - February 2, 1994

Version 1.10 Release Date - February 9, 1994

Version 1.11 Release Date - February 20, 1994

Version 1.12 Release Date - March 1, 1994

Version 1.13 Release Date - May 12, 1994

Copyright © Richard S. Huntrods, 1994

MIDIBars is a SHAREWARE program.

THIS PROGRAM IS NOT FREE!!!

MIDIBars Introduction

MIDIBars is a MIDI control program designed to provide the user with 16 faders, 16 buttons, and a keyboard.

THIS PROGRAM IS SHAREWARE.

The 16 faders can function as MIDI controls for a variety of MIDI functions which require incremental control (i.e. Program Change or Volume).

The 16 buttons can function as MIDI controls for a variety of MIDI functions which require ON/OFF control (i.e. Sustain).

The keyboard provides a simple MIDI keyboard for testing the effect of the faders and buttons.

The current version of MIDIBars provides a number of configurations. More configurations are planned for future versions of MIDIBars, as well as custom Synthesizer Programming Modules.

Synthesizer Programming Modules for MIDIBars are custom configurations which will allow real-time programming of specific synthesizers under MIDI control. By purchasing the modules for your particular synthesizers, you will be able to edit various patch parameters in REAL TIME (i.e. during a live performance). Since the edits are via MIDI, you should also be able to capture them with your sequencer. **ONLY REGISTERED MIDIBars USERS WILL BE ABLE TO PURCHASE THESE MODULES.**

Shareware

MIDIbars is a SHAREWARE program produced by Richard S. [Huntrods](#)

This program is copyright © Richard S. [Huntrods](#)Huntrods, 1994

THIS PROGRAM IS NOT FREE!!!

The concept of shareware is to allow the user to try the software before committing to the purchase. If you use this program and intend to continue using it (even on a VERY casual basis), you should REGISTER the software with the author.

The registration fee for MIDIbars is \$25.00 US, payable by personal cheque made out to Richard S. [Huntrods](#)

If you register, you will receive the next update of the software free, as well as any bug fixes.. Future updates MAY have a nominal cost associated with their release. You will also be eligible to purchase planned Synthesizer Programming Modules for MIDIbars.

Synthesizer Programming Modules for MIDIbars are custom configurations which will allow real-time programming of specific synthesizers under MIDI control. By purchasing the modules for your particular synthesizers, you will be able to edit various patch parameters in REAL TIME (i.e. during a live performance). Since the edits are via MIDI, you should also be able to capture them with your sequencer. **ONLY REGISTERED MIDIbars USERS WILL BE ABLE TO PURCHASE THESE MODULES.**

Synthesizer Programming Modules will become available for particular synthesizers in the coming months on a demand basis. The first modules planned are for the Proteus 1, Proteus 2, and Oberheim Matrix 6R and 1000.

Registration

To register MIDIBars, send a personal cheque or money order for \$25.00 US made out to:

Richard S. Huntrods

and send it along with your name, address and phone number to:

Richard S. Huntrods
c/o Richard Huntrods Systems Consulting
95 Templegreen Drive N.E.
Calgary, Alberta, Canada
T1Y 4Z1

Upon receipt and processing of your registration, you will receive a MIDIBars registration number by return mail.

Registered users are eligible for the following:

- they will receive the next version of MIDIBars free.
- they will receive all bug fixes for the current version of MIDIBars free.
- they will be able to purchase future versions of MIDIBars for a nominal fee.
- they will be able to purchase Synthesizer Programming Modules for MIDIBars.
- they will be able to request specific Synthesizer Programming Modules.

Richard S. Huntrods

Richard S. Huntrods is the genius behind MIDIBars.

He lives in Calgary, Alberta, Canada with his wife, two teenage children, and two Siamese cats. In spite of this he still has time to run the company Richard Huntrods Systems Consulting.

Richard has accounts on both Genie and Compuserve, and may also be reached by snail mail (see the registration screen). Richard's Compuserve address is 70400, 2434. His Genie mail address is R.HUNTRODS.

Mr. Huntrods has a B.Sc (1980). and an M.Eng (1986), both in Chemical Engineering, from the University of Calgary. He is also a registered Professional Engineer in the province of Alberta, Canada.

A Resume and single page Skill Sheet is available upon request.

Programming Introduction

If you have not yet registered MIDIBars, you should! Read through this document, and see what you're missing! If you **have** registered MIDIBars, **THANK YOU!**

Welcome to the wonderful world of MIDIBars Programming! I hope you find it a useful program both for MIDI real-time control, and just general MIDI playing around.

This document describes the MIDIBars initialization file, MIDIBARS.INI. This file is required for MIDIBars **user configurations**. Using this document, you will be able to customize the user configuration screens which are enabled when you **register** your copy of MIDIBars. The registered version of MIDIBars is shipped with a sample MIDIBARS.INI file, which contains two active user configurations, and three additional user configurations which serve as examples of MIDIBARS.INI customization. There are also two examples of existing MIDIBars configurations (available in the pull down configuration menu), provided as reference material. As you shall see, it is possible to enable any two of the five supplied configurations supplied in the sample MIDIBARS.INI file at one time.

MIDIBARS.INI Programming

MIDIBARS.INI is an ASCII text file, which can be edited by any ASCII editor. In Windows, the NOTEPAD editor is a good choice to use for editing MIDIBARS.INI.

When you first look at MIDIBARS.INI, you will see a large collection of keywords and values. Don't be alarmed. MIDIBARS.INI is grouped into easy to program sections. Each section is controlled by a **general keyword**. Within each general section, you program 16 individual MIDI sliders or buttons using **MIDI event keywords**. The **MIDI event keywords** are described in the next part of this document.

Lets look at an overview of MIDIBARS.INI, focusing in on the **general keywords**:

```
;  
; Example MIDIBARS.INI File Layout  
;  
UserConfig=1  
AllOneChannel=No  
Bars=  
;      * define 16 fader messages *  
ButtonUp=  
;      * define 16 button "up" messages *  
ButtonDown=  
;      * define 16 button "down" messages *  
UserConfig=2  
AllOneChannel=Yes  
Bars=  
;      * define 16 fader messages *  
ButtonUp=  
;      * define 16 button "up" messages *  
ButtonDown=  
;      * define 16 button "down" messages *
```

In the example above, I have defined two user configurations. Lets look at the details.

MIDIBARS.INI is organized into **user configuration** sections. Each section begins with the keyword **UserConfig=** followed by a number. Only configurations 1 and 2 will be loaded and used by MIDIBars. Any other configuration numbers are for configuration storage only. MIDIBARS.INI places no limit on the number of configurations you can store in it, but will only load configurations numbered "1" and "2". This allows you to keep multiple configurations "on-line", and switch them (fairly) easily using an ASCII editor. This is demonstrated in the example MIDIBARS.INI file shipped with the registered version of MIDIBars, which contains 5 configurations for example purposes.

MIDIBARS.INI allows you to place comments anywhere in the file for documentation purposes. Comments are lines beginning with the semicolon-colon ";". Any text after a line beginning with a semicolon-colon is ignored by MIDIBars. It is a good practice to document your configurations for future reference.

The keyword **AllOneChannel=** is used to tell MIDIBars whether or not this user configuration

applies to multiple MIDI channels, or just one MIDI channel. MIDIBars behaves differently for single channel screens than it does for multi-channel screens. If you do not correctly inform MIDIBars of the number of channels (one or many), MIDIBars will reset various buttons and sliders in unexpected ways as you change from configuration to configuration.

NOTE: Be very careful when assigning MIDI channel (next section of this document) when AllOneChannel=Yes, as MIDIBars assigns the MIDI channel from a drop down menu for this case.

Within each user configuration, there are keywords to tell MIDIBars what MIDI events to assign to what MIDIBars control (slider or button). The keyword **Bars=** signals the beginning of the fader programming section of this user configuration. The keyword **ButtonDown=** is used to designate MIDI events to the press of a button, while the **ButtonUp=** keyword is used for the release of the same button. **NOTE:** you should be very careful to pair ButtonDown and ButtonUp MIDI events, or unexpected things will happen with your MIDI device!

Within each **MIDI event section** (Bars=, ButtonDown=, ButtonUp=), you **must** designate a separate MIDI event for each of the 16 MIDI buttons or sliders that comprise a user configuration screen. Do not leave any out, or MIDIBars will not function correctly.

The format of a MIDI control is shown below. You must supply 16 lines of text per MIDI event section, one for each MIDI control. The first two entries in the MIDI control line **must** be enclosed in parentheses, as these designate text strings associated with the control. The rest of the line represents MIDI control keywords, described later in this document.

Programming MIDIBars is simply a matter of creating the various MIDIBars user configurations and MIDI event sections, then selecting the MIDI events you want to associate with each of the 16 sliders and 16 buttons and specifying the information for each.

Although MIDIBars requires a detailed specification for each MIDI event, using the proper keywords, you can get started with no more information than is supplied in the example MIDIBARS.INI file. I have provided you with all of the most commonly used MIDI events (and some uncommon ones). To program a MIDI event, simply look for an example of the event you want in the MIDIBARS.INI file, and copy it to your configuration. All you need to change in almost every case is the channel assignment.

If you want a more complete understanding of MIDI event programming, consult one of the many fine books on MIDI programming, available from such publishers as the MIX bookshelf. To program my example MIDIBARS.INI file, I referred to two books, listed below.

References:

Steve De Furia and Joe Scacciaferro, *MIDI Programmer's Handbook*, Ferro Technologies, M&T Books, 1989. ISBN: 1-55851-068-0

Steve De Furia and Joe Scacciaferro, *The MIDI Resource Book*, The Ferro MIDI

Reference Series - Book 1, Ferro Technologies, 1988. ISBN: 0-88188-587-8

MIDIBARS.INI Control Line Format

The following is a description of the contents of a MIDI control line. You are required to enter one control line for each of 16 MIDI sliders or buttons in each of the MIDI event sections.

- LABEL** The first field is the MIDI control text, displayed below the slider or on the button. It consists of up to four (4) characters enclosed in double quotes. The information will be displayed exactly as entered (case sensitive). I suggest using ALL CAPITALS for a button "down" message, and mixed upper and lower case for sliders and button "up" messages
- NAME** The second field is a longer text field, used to describe the MIDI controller being programmed. You can use up to 11 characters in this field, again enclosed in double quotes.
- EVENT** The third field contains the keyword describing the MIDI event associated with the control. There are two types of MIDI events, **single keyword** events (like PGRM_CHG), and **control message** events, where the first keyword is **always CTRL_MSG**.
- MESSAGE 1** The fourth field contains the first message keyword associated with the **EVENT** keyword. It is always one of two keyword types. For **single keyword** events, this field contains the **MIDI value** keyword for the control such as **CTRL_ON** or **CTRL_OFF** (for buttons), or **USE_VALUE** (for sliders). For **control message** events, this field contains the event **sub-keyword**, used to distinguish which MIDI control message is required (i.e. **SUSTAIN**).
- MESSAGE 2** The fifth field contains a second message keyword associated with the **EVENT** keyword. It is always one of two keyword types. For **single keyword** events, this field always contains the keyword **UNDEFINED** (i.e. the field is not used by the control). For **control message** events, this field contains the **MIDI value** keyword for the control, usually one of **CTRL_ON**, **CTRL_OFF** (for buttons), or **USE_VALUE** (for sliders).
- NOTE:** To determine which MIDI events are single keyword events, and which are control message events, refer to one of the reference books on MIDI referred to in the previous section of this document, or simply use the example MIDIBARS.INI file provided later in this document as a guide for all of the MIDI controllers normally used in MIDI control.
- CHANNEL** The last field contains the MIDI channel assignment keyword. This is **always** in the form of a mathematical equation, containing the keyword **CHN_MIN + value**, where **value** is a number from 0 to 15, corresponding to MIDI channels 1 to 16. In other words, **CHN_MIN + 0** represents MIDI channel 1, while **CHN_MIN + 15** represents MIDI channel 16.

NOTE: If AllOneChannel is set to **Yes**, you **must only use CHN_MIN** as the specified channel. This is because MIDIBars will supply the MIDI channel from a drop down menu.

Here is an example MIDI control line for a fader: It specifies a Program Change fader, assigned to MIDI channel 1, using the value of the actual slider position in MIDIBars. Below the slider, the text "PgCh" will be displayed. Because program change is a single keyword MIDI event, the MESSAGE 2 field is set to UNDEFINED (as explained above).

```
"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE , UNDEFINED , CHN_MIN + 0
```

Here is an example MIDI control line for a button "down" control: This control represents the pressing down of a sustain pedal, assigned to MIDI channel 6. A sustain pedal is an example of a control message event, where the MESSAGE 1 field specifies the type of control message as sustain. The MESSAGE 2 field provides a MIDI value appropriate to turning the control on. The button text will change to "SUS6" whenever this button is "down".

```
"SUS6", "Sustain ", CTRL_MSG , SUSTAIN , CTRL_ON , CHN_MIN + 5
```

Here is an example MIDI control line for a button "up" control: It is the mirror image of the previous example, set for the same channel, but turning the sustain control on MIDI channel 6 off when the button is "up". Whenever the button is "up", the text "Sus6" will be displayed on the face of the button.

```
"Sus6", "Sustain ", CTRL_MSG , SUSTAIN , CTRL_OFF , CHN_MIN + 5
```

Note how the MIDI control line for a button "up" mirrors the line for the button "down". The only difference is the CAPITALIZATION of the button text for a button "down" message, and the change from CTRL_ON to CTRL_OFF.

MIDIBARS.INI General Keywords

The following is a list of all of the MIDIBARS.INI **general keywords**. **General keywords must** begin in the first column of the file, typed **exactly** as shown (upper *and* lower case), with no spaces between the keyword and any arguments.

- ;** denotes a comment. Any text after the ";" is ignored. The ";" **must** occur in the first column of the line, followed by a space.

- UserConfig=** (1, 2, ..., n) signals the beginning of a MIDIBars **user configuration**. You **must** complete all of the programming for a user configuration before starting a new user configuration.

- AllOneChannel=** (Yes, No) tells MIDIBars whether or not this user configuration is designed for multiple MIDI channels (1-16) or for just one channel.

- Bars=** signals the beginning of the slider MIDI profile for this user configuration. This keyword **must** appear all by itself on one line. This section controls what is sent out the MIDI device when you change the slider settings in MIDIBars. Following this keyword, you **must** define MIDI events for all 16 MIDI sliders.

- ButtonUp=** signals the beginning of the button "up" MIDI profile for this user configuration. This keyword **must** appear all by itself on one line. This section controls what is sent out the MIDI device when you push a button "down" or "on" in MIDIBars. Following this keyword, you **must** define MIDI events for all 16 MIDI buttons.

- ButtonDown=** signals the beginning of the button "down" MIDI profile for this user configuration. This keyword **must** appear all by itself on one line. This section controls what is sent out the MIDI device when you push a button "up" or "off" in MIDIBars. Following this keyword, you **must** define MIDI events for all 16 MIDI buttons.

MIDIBARS.INI MIDI Event Keywords

The following information comprises all of the special keywords used in specifying user slider and button functions in the MIDIBars initialization (INI) file. Beside each INI file keyword is a definition of what the keyword represents in the INI file. These keywords are used in the **MIDI event section** (Bars=, ButtonDown=, ButtonUp=) of the MIDIBARS.INI file.

Please refer to the example MIDIBars.INI file included later in this document for details of how the INI keywords are used in an actual INI file.

Please note: All keywords **MUST** be typed into the INI file **EXACTLY** as shown. All MIDIBars INI keywords are UPPER CASE.

Keywords shown in **bold** represent special, frequently used keywords. Please note what they represent.

Special Keywords:

CHN_MIN	minimum MIDI channel (channel 1). Most often used in an equation. EXAMPLE: CHN_MIN + 5 (for channel 5)
USE_VALUE	use the actual slider value. Used with every MIDI slider.

MIDI Keywords:

CTRL_OFF	turn control off. Often used with MIDI buttons.
CTRL_ON	turn control on. Often used with MIDI buttons.
CTRL_MSG	MIDI control message
PGRM_CHG	MIDI program change message
CHNL_PRS	channel pressure or aftertouch
PTCH_BND	pitch bend

MIDI 14-bit Continuous Controller Keywords:

UNDEFINED	undefined. Use this MIDI event if you do not want anything assigned to the button or slider.
MODULATION	modulation wheel
BREATH	breath controller
FOOT	foot controller
PORT_TIME	portamento time
DATA_MSB	data most significant bit
VOLUME	volume
BALANCE	balance
PAN	pan
EXPRESSION	expression pedal
GENERAL_1	general MIDI message #1

GENERAL_2	general MIDI message #2
GENERAL_3	general MIDI message #3
GENERAL_4	general MIDI message #4

MIDI 7-bit Switch Controller Keywords:

SUSTAIN	sustain pedal
PORTAMENTO	portamento
SOSTENUTO	sostenuto pedal
SOFT_PEDAL	soft or damper pedal

MIDI 7-bit Continuous Controller Keywords:

GENERAL_5	general MIDI message #5
GENERAL_6	general MIDI message #6
GENERAL_7	general MIDI message #7
GENERAL_8	general MIDI message #8
EFFECTS	effects
TREMOLO	tremolo effect
CHORUS	chorus effect
DETUNE	detune effect
PHASER	phaser effect

MIDI Channel Mode Message Keywords:

RESET_CTRL	reset control
LOCAL	local control
OMNI_OFF	omni mode off
OMNI_ON	omni mode on
MONO_ON	mono mode on (poly mode off)
POLY_ON	poly mode on (mono mode off)
MONO_CHNL	mono mode channel

MIDIBARS.INI Annotated Examples

The following example is the MIDIBARS.INI file supplied with the registered version of MIDIBars. There are five (5) example user configurations supplied in this file, although only configurations labeled "UserConfig=1" and "UserConfig=2" will be used by MIDIBars. By editing this file and changing which configuration is labeled "UserConfig=1" and /or "UserConfig=2", you can cause MIDIBars to load and use any two of the five supplied configurations.

```
;
; MIDIBARS.INI User Configuration Initialization
;
; RSH MIDI Controller Program
; (c) Richard Huntrods, 1994
;
UserConfig=1
;
; This is the first (default) user configuration
; It assigns the sliders and buttons in groups of 4, one for each of
; the first 4 MIDI channels (1-4)
; Faders are assigned to Program Change, Volume, Pitch Bend and Pan
; Buttons are assigned Sustain, Portamento, Sostenuto, and Soft Pedal
;
; Since 4 MIDI channels are assigned, AllOneChannel must be No.
;
; Note the use of UPPER and Mixed Case to distinguish button up and down
;
AllOneChannel=No
Bars=
"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE , UNDEFINED , CHN_MIN + 0
"Vol ", "Volume ", CTRL_MSG , VOLUME , USE_VALUE , CHN_MIN + 0
"PBnd", "Pitch Bend ", PTCH_BND , USE_VALUE , UNDEFINED , CHN_MIN + 0
"Pan ", "Pan ", CTRL_MSG , PAN , USE_VALUE , CHN_MIN + 0
"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE , UNDEFINED , CHN_MIN + 1
"Vol ", "Volume ", CTRL_MSG , VOLUME , USE_VALUE , CHN_MIN + 1
"PBnd", "Pitch Bend ", PTCH_BND , USE_VALUE , UNDEFINED , CHN_MIN + 1
"Pan ", "Pan ", CTRL_MSG , PAN , USE_VALUE , CHN_MIN + 1
"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE , UNDEFINED , CHN_MIN + 2
"Vol ", "Volume ", CTRL_MSG , VOLUME , USE_VALUE , CHN_MIN + 2
"PBnd", "Pitch Bend ", PTCH_BND , USE_VALUE , UNDEFINED , CHN_MIN + 2
"Pan ", "Pan ", CTRL_MSG , PAN , USE_VALUE , CHN_MIN + 2
"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE , UNDEFINED , CHN_MIN + 3
"Vol ", "Volume ", CTRL_MSG , VOLUME , USE_VALUE , CHN_MIN + 3
"PBnd", "Pitch Bend ", PTCH_BND , USE_VALUE , UNDEFINED , CHN_MIN + 3
"Pan ", "Pan ", CTRL_MSG , PAN , USE_VALUE , CHN_MIN + 3
ButtonUp=
"Sus1", "Sustain ", CTRL_MSG , SUSTAIN , CTRL_OFF , CHN_MIN + 0
"Por1", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_OFF , CHN_MIN + 0
"Sos1", "Sostenuto ", CTRL_MSG , SOSTENUTO , CTRL_OFF , CHN_MIN + 0
"Sof1", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL , CTRL_OFF , CHN_MIN + 0
"Sus2", "Sustain ", CTRL_MSG , SUSTAIN , CTRL_OFF , CHN_MIN + 1
"Por2", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_OFF , CHN_MIN + 1
"Sos2", "Sostenuto ", CTRL_MSG , SOSTENUTO , CTRL_OFF , CHN_MIN + 1
"Sof2", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL , CTRL_OFF , CHN_MIN + 1
```



```

"Sus3", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_OFF , CHN_MIN + 2
"Por3", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_OFF , CHN_MIN + 2
"Sos3", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_OFF , CHN_MIN + 2
"Sof3", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_OFF , CHN_MIN + 2
"Sus4", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_OFF , CHN_MIN + 3
"Por4", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_OFF , CHN_MIN + 3
"Sos4", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_OFF , CHN_MIN + 3
"Sof4", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_OFF , CHN_MIN + 3
ButtonDown=
"SUS1", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_ON  , CHN_MIN + 0
"POR1", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 0
"SOS1", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 0
"SOFF1", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 0
"SUS2", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_ON  , CHN_MIN + 1
"POR2", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 1
"SOS2", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 1
"SOFF2", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 1
"SUS3", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_ON  , CHN_MIN + 2
"POR3", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 2
"SOS3", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 2
"SOFF3", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 2
"SUS4", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_ON  , CHN_MIN + 3
"POR4", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 3
"SOS4", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 3
"SOFF4", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 3
;
UserConfig=2
;
; This is the second (default) user configuration
; It assigns the sliders and buttons in groups of 4, one for each of
; the next 4 MIDI channels after user configuration 1 (5-8)
; Faders and Buttons are assigned the same as User Config 1
; except for the MIDI channel.
;
AllOneChannel=No
Bars=
"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 4
"Vol ", "Volume      ", CTRL_MSG , VOLUME      , USE_VALUE , CHN_MIN + 4
"PBnd", "Pitch Bend  ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 4
"Pan ", "Pan         ", CTRL_MSG , PAN         , USE_VALUE , CHN_MIN + 4
"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 5
"Vol ", "Volume      ", CTRL_MSG , VOLUME      , USE_VALUE , CHN_MIN + 5
"PBnd", "Pitch Bend  ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 5
"Pan ", "Pan         ", CTRL_MSG , PAN         , USE_VALUE , CHN_MIN + 5
"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 6
"Vol ", "Volume      ", CTRL_MSG , VOLUME      , USE_VALUE , CHN_MIN + 6
"PBnd", "Pitch Bend  ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 6
"Pan ", "Pan         ", CTRL_MSG , PAN         , USE_VALUE , CHN_MIN + 6
"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 7
"Vol ", "Volume      ", CTRL_MSG , VOLUME      , USE_VALUE , CHN_MIN + 7
"PBnd", "Pitch Bend  ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 7
"Pan ", "Pan         ", CTRL_MSG , PAN         , USE_VALUE , CHN_MIN + 7
ButtonUp=
"Sus5", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_OFF , CHN_MIN + 4
"Por5", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_OFF , CHN_MIN + 4
"Sos5", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_OFF , CHN_MIN + 4
"Sof5", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_OFF , CHN_MIN + 4

```

```

"Sus6", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_OFF , CHN_MIN + 5
"Por6", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_OFF , CHN_MIN + 5
"Sos6", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_OFF , CHN_MIN + 5
"Sof6", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_OFF , CHN_MIN + 5
"Sus7", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_OFF , CHN_MIN + 6
"Por7", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_OFF , CHN_MIN + 6
"Sos7", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_OFF , CHN_MIN + 6
"Sof7", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_OFF , CHN_MIN + 6
"Sus8", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_OFF , CHN_MIN + 7
"Por8", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_OFF , CHN_MIN + 7
"Sos8", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_OFF , CHN_MIN + 7
"Sof8", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_OFF , CHN_MIN + 7
ButtonDown=
"SUS5", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_ON  , CHN_MIN + 4
"POR5", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 4
"SOS5", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 4
"SOF5", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 4
"SUS6", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_ON  , CHN_MIN + 5
"POR6", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 5
"SOS6", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 5
"SOF6", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 5
"SUS7", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_ON  , CHN_MIN + 6
"POR7", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 6
"SOS7", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 6
"SOF7", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 6
"SUS8", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_ON  , CHN_MIN + 7
"POR8", "Portamento  ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 7
"SOS8", "Sostenuto    ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 7
"SOF8", "Soft Pedal   ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 7
;
UserConfig=3
;
; This is the third user configuration (not loaded by MIDIBars)
; It assigns the sliders and buttons in groups of 4, one for each of
; the next 4 MIDI channels after user configuration 2 (9-12)
; Faders and Buttons are assigned the same as User Config 1
; except for the MIDI channel.
;
AllOneChannel=No
Bars=
"PgCh", "Pgm Change  ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 8
"Vol ", "Volume       ", CTRL_MSG , VOLUME     , USE_VALUE , CHN_MIN + 8
"PBnd", "Pitch Bend   ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 8
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 8
"PgCh", "Pgm Change  ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 9
"Vol ", "Volume       ", CTRL_MSG , VOLUME     , USE_VALUE , CHN_MIN + 9
"PBnd", "Pitch Bend   ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 9
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 9
"PgCh", "Pgm Change  ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 10
"Vol ", "Volume       ", CTRL_MSG , VOLUME     , USE_VALUE , CHN_MIN + 10
"PBnd", "Pitch Bend   ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 10
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 10
"PgCh", "Pgm Change  ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 11
"Vol ", "Volume       ", CTRL_MSG , VOLUME     , USE_VALUE , CHN_MIN + 11
"PBnd", "Pitch Bend   ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 11
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 11
ButtonUp=

```

```

"Sus9", "Sustain      ", CTRL_MSG , SUSTAIN      , CTRL_OFF , CHN_MIN + 8
"Por9", "Portamento    ", CTRL_MSG , PORTAMENTO   , CTRL_OFF , CHN_MIN + 8
"Sos9", "Sostenuto      ", CTRL_MSG , SOSTENUTO   , CTRL_OFF , CHN_MIN + 8
"Sof9", "Soft Pedal     ", CTRL_MSG , SOFT_PEDAL  , CTRL_OFF , CHN_MIN + 8
"Su10", "Sustain        ", CTRL_MSG , SUSTAIN     , CTRL_OFF , CHN_MIN + 9
"Pr10", "Portamento    ", CTRL_MSG , PORTAMENTO   , CTRL_OFF , CHN_MIN + 9
"Ss10", "Sostenuto      ", CTRL_MSG , SOSTENUTO   , CTRL_OFF , CHN_MIN + 9
"Sf10", "Soft Pedal     ", CTRL_MSG , SOFT_PEDAL  , CTRL_OFF , CHN_MIN + 9
"Su11", "Sustain        ", CTRL_MSG , SUSTAIN     , CTRL_OFF , CHN_MIN + 10
"Pr11", "Portamento    ", CTRL_MSG , PORTAMENTO   , CTRL_OFF , CHN_MIN + 10
"Sa11", "Sostenuto      ", CTRL_MSG , SOSTENUTO   , CTRL_OFF , CHN_MIN + 10
"Sf11", "Soft Pedal     ", CTRL_MSG , SOFT_PEDAL  , CTRL_OFF , CHN_MIN + 10
"Su12", "Sustain        ", CTRL_MSG , SUSTAIN     , CTRL_OFF , CHN_MIN + 11
"Pr12", "Portamento    ", CTRL_MSG , PORTAMENTO   , CTRL_OFF , CHN_MIN + 11
"Ss12", "Sostenuto      ", CTRL_MSG , SOSTENUTO   , CTRL_OFF , CHN_MIN + 11
"Sf12", "Soft Pedal     ", CTRL_MSG , SOFT_PEDAL  , CTRL_OFF , CHN_MIN + 11
ButtonDown=
"SUS9", "Sustain        ", CTRL_MSG , SUSTAIN     , CTRL_ON  , CHN_MIN + 8
"POR9", "Portamento    ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 8
"SOS9", "Sostenuto      ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 8
"SOF9", "Soft Pedal     ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 8
"SU10", "Sustain        ", CTRL_MSG , SUSTAIN     , CTRL_ON  , CHN_MIN + 9
"PR10", "Portamento    ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 9
"SS10", "Sostenuto      ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 9
"SF10", "Soft Pedal     ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 9
"SU11", "Sustain        ", CTRL_MSG , SUSTAIN     , CTRL_ON  , CHN_MIN + 10
"PR11", "Portamento    ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 10
"SA11", "Sostenuto      ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 10
"SF11", "Soft Pedal     ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 10
"SU12", "Sustain        ", CTRL_MSG , SUSTAIN     , CTRL_ON  , CHN_MIN + 11
"PR12", "Portamento    ", CTRL_MSG , PORTAMENTO   , CTRL_ON  , CHN_MIN + 11
"SS12", "Sostenuto      ", CTRL_MSG , SOSTENUTO   , CTRL_ON  , CHN_MIN + 11
"SF12", "Soft Pedal     ", CTRL_MSG , SOFT_PEDAL  , CTRL_ON  , CHN_MIN + 11
;
UserConfig=4
;
; This is the forth user configuration (not loaded by MIDIBars)
; It assigns the sliders and buttons in groups of 4, one for each of
; the next 4 MIDI channels after user configuration 3 (13-16)
; Faders and Buttons are assigned the same as User Config 1
; except for the MIDI channel.
;
AllOneChannel=No
Bars=
"PgCh", "Pgm Change  ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 12
"Vol ", "Volume       ", CTRL_MSG , VOLUME     , USE_VALUE , CHN_MIN + 12
"PBnd", "Pitch Bend   ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 12
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 12
"PgCh", "Pgm Change  ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 13
"Vol ", "Volume       ", CTRL_MSG , VOLUME     , USE_VALUE , CHN_MIN + 13
"PBnd", "Pitch Bend   ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 13
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 13
"PgCh", "Pgm Change  ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 14
"Vol ", "Volume       ", CTRL_MSG , VOLUME     , USE_VALUE , CHN_MIN + 14
"PBnd", "Pitch Bend   ", PTCH_BND , USE_VALUE   , UNDEFINED , CHN_MIN + 14
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 14
"PgCh", "Pgm Change  ", PGRM_CHG , USE_VALUE   , UNDEFINED , CHN_MIN + 15

```

```

"Vol ", "Volume      ", CTRL_MSG , VOLUME      , USE_VALUE , CHN_MIN + 15
"PBnd", "Pitch Bend  ", PTCH_BND , USE_VALUE , UNDEFINED , CHN_MIN + 15
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 15
ButtonUp=
"Sul3", "Sustain     ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 12
"Pr13", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_OFF  , CHN_MIN + 12
"Ss13", "Sostenuto  ", CTRL_MSG , SOSTENUTO , CTRL_OFF  , CHN_MIN + 12
"Sf13", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL, CTRL_OFF  , CHN_MIN + 12
"Sul4", "Sustain     ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 13
"Pr14", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_OFF  , CHN_MIN + 13
"Ss14", "Sostenuto  ", CTRL_MSG , SOSTENUTO , CTRL_OFF  , CHN_MIN + 13
"Sf14", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL, CTRL_OFF  , CHN_MIN + 13
"Sul5", "Sustain     ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 14
"Pr15", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_OFF  , CHN_MIN + 14
"Sa15", "Sostenuto  ", CTRL_MSG , SOSTENUTO , CTRL_OFF  , CHN_MIN + 14
"Sf15", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL, CTRL_OFF  , CHN_MIN + 14
"Sul6", "Sustain     ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 15
"Pr16", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_OFF  , CHN_MIN + 15
"Ss16", "Sostenuto  ", CTRL_MSG , SOSTENUTO , CTRL_OFF  , CHN_MIN + 15
"Sf16", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL, CTRL_OFF  , CHN_MIN + 15
ButtonDown=
"SU13", "Sustain     ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 12
"PR13", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_ON   , CHN_MIN + 12
"SS13", "Sostenuto  ", CTRL_MSG , SOSTENUTO , CTRL_ON   , CHN_MIN + 12
"SF13", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL, CTRL_ON   , CHN_MIN + 12
"SU14", "Sustain     ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 13
"PR14", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_ON   , CHN_MIN + 13
"SS14", "Sostenuto  ", CTRL_MSG , SOSTENUTO , CTRL_ON   , CHN_MIN + 13
"SF14", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL, CTRL_ON   , CHN_MIN + 13
"SU15", "Sustain     ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 14
"PR15", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_ON   , CHN_MIN + 14
"SA15", "Sostenuto  ", CTRL_MSG , SOSTENUTO , CTRL_ON   , CHN_MIN + 14
"SF15", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL, CTRL_ON   , CHN_MIN + 14
"SU16", "Sustain     ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 15
"PR16", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_ON   , CHN_MIN + 15
"SS16", "Sostenuto  ", CTRL_MSG , SOSTENUTO , CTRL_ON   , CHN_MIN + 15
"SF16", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL, CTRL_ON   , CHN_MIN + 15
;
UserConfig=5
;
; This is the fifth user configuration (not loaded by MIDIBars)
; All Faders are assigned the Pan event, one for each MIDI channel
; All buttons are assigned the Sustain event, one for each MIDI channel
;
AllOneChannel=No
Bars=
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 0
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 1
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 2
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 3
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 4
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 5
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 6
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 7
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 8
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 9
"Pan ", "Pan          ", CTRL_MSG , PAN        , USE_VALUE , CHN_MIN + 10

```

```

"Pan ", "Pan      ", CTRL_MSG , PAN          , USE_VALUE , CHN_MIN + 11
"Pan ", "Pan      ", CTRL_MSG , PAN          , USE_VALUE , CHN_MIN + 12
"Pan ", "Pan      ", CTRL_MSG , PAN          , USE_VALUE , CHN_MIN + 13
"Pan ", "Pan      ", CTRL_MSG , PAN          , USE_VALUE , CHN_MIN + 14
"Pan ", "Pan      ", CTRL_MSG , PAN          , USE_VALUE , CHN_MIN + 15
ButtonUp=
"St01", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 0
"St02", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 1
"St03", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 2
"St04", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 3
"St05", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 4
"St06", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 5
"St07", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 6
"St08", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 7
"St09", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 8
"St10", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 9
"St11", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 10
"St12", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 11
"St13", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 12
"St14", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 13
"St15", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 14
"St16", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_OFF  , CHN_MIN + 15
ButtonDown=
"ST01", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 0
"ST02", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 1
"ST03", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 2
"ST04", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 3
"ST05", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 4
"ST06", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 5
"ST07", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 6
"ST08", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 7
"ST09", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 8
"ST10", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 9
"ST11", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 10
"ST12", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 11
"ST13", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 12
"ST14", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 13
"ST15", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 14
"ST16", "Sustain  ", CTRL_MSG , SUSTAIN    , CTRL_ON   , CHN_MIN + 15
;
UserConfig=6
;
; This is the sixth user configuration (not loaded by MIDIBars)
; This configuration is identical to a configuration built into
; MIDIBars, which supplies lots of sliders and buttons to one
; MIDI channel. This example configuration is supplied for MIDI event
; reference ONLY (since it is built into MIDIBars).
;
; NOTE the use of complete UNDEFINED MIDI events to program unused
; MIDI buttons
;
; Also NOTE the only channel used is CHN_MIN - since MIDIBars will supply
; other channel assignments from a drop down menu - NOT OTHER CHANNEL
; should be used when AllOneChannel=Yes
;
AllOneChannel=Yes
Bars=

```

```

"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE , UNDEFINED , CHN_MIN
"Vol ", "Volume ", CTRL_MSG , VOLUME , USE_VALUE , CHN_MIN
"Brth", "Breath ", CTRL_MSG , BREATH , USE_VALUE , CHN_MIN
"Foot", "Foot ", CTRL_MSG , FOOT , USE_VALUE , CHN_MIN
"PTim", "Port Time ", CTRL_MSG , PORT_TIME , USE_VALUE , CHN_MIN
"Dmsb", "Data MSB ", CTRL_MSG , DATA_MSB , USE_VALUE , CHN_MIN
"Expr", "Expression ", CTRL_MSG , EXPRESSION , USE_VALUE , CHN_MIN
"Gen1", "General 1 ", CTRL_MSG , GENERAL_1 , USE_VALUE , CHN_MIN
"Gen2", "General 2 ", CTRL_MSG , GENERAL_2 , USE_VALUE , CHN_MIN
"Gen3", "General 3 ", CTRL_MSG , GENERAL_3 , USE_VALUE , CHN_MIN
"Gen4", "General 4 ", CTRL_MSG , GENERAL_4 , USE_VALUE , CHN_MIN
"Effft", "Effects ", CTRL_MSG , EFFECTS , USE_VALUE , CHN_MIN
"Trem", "Tremolo ", CTRL_MSG , TREMOLO , USE_VALUE , CHN_MIN
"Chrs", "Chorus ", CTRL_MSG , CHORUS , USE_VALUE , CHN_MIN
"Dtun", "Detune ", CTRL_MSG , DETUNE , USE_VALUE , CHN_MIN
"Phsr", "Phaser ", CTRL_MSG , PHASER , USE_VALUE , CHN_MIN
ButtonUp=
"Locl", "Local Mode ", CTRL_MSG , LOCAL , CTRL_OFF , CHN_MIN
"Omni", "Omni Mode ", CTRL_MSG , OMNI_OFF , UNDEFINED , CHN_MIN
"Mono", "Mono Mode ", CTRL_MSG , POLY_ON , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Sust", "Sustain ", CTRL_MSG , SUSTAIN , CTRL_OFF , CHN_MIN
"Port", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_OFF , CHN_MIN
"Sost", "Sostenuto ", CTRL_MSG , SOSTENUTO , CTRL_OFF , CHN_MIN
"Soft", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL , CTRL_OFF , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
ButtonDown=
"LOCL", "Local Mode ", CTRL_MSG , LOCAL , CTRL_ON , CHN_MIN
"OMNI", "Omni Mode ", CTRL_MSG , OMNI_ON , UNDEFINED , CHN_MIN
"MONO", "Mono Mode ", CTRL_MSG , MONO_ON , MONO_CHNL , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"SUST", "Sustain ", CTRL_MSG , SUSTAIN , CTRL_ON , CHN_MIN
"PORT", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_ON , CHN_MIN
"SOST", "Sostenuto ", CTRL_MSG , SOSTENUTO , CTRL_ON , CHN_MIN
"SOFT", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL , CTRL_ON , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
;
UserConfig=7
;
; This is the seventh user configuration (not loaded by MIDIBars)
; This configuration is identical to a configuration built into
; MIDIBars, which supplies more sliders and buttons to one
; MIDI channel. This example configuration is supplied for MIDI event

```

```

; reference ONLY (since it is built into MIDIBars).
;
; NOTE the use of complete UNDEFINED MIDI events to program unused
; MIDI sliders and buttons
;
; Also NOTE the only channel used is CHN_MIN - since MIDIBars will supply
; other channel assignments from a drop down menu - NOT OTHER CHANNEL
; should be used when AllOneChannel=Yes
;

```

```
AllOneChannel=Yes
```

```
Bars=
```

```

"PgCh", "Pgm Change ", PGRM_CHG , USE_VALUE , UNDEFINED , CHN_MIN
"Vol ", "Volume ", CTRL_MSG , VOLUME , USE_VALUE , CHN_MIN
"PBnd", "Pitch Bend ", PTCH_BND , USE_VALUE , UNDEFINED , CHN_MIN
"Pan ", "Pan ", CTRL_MSG , PAN , USE_VALUE , CHN_MIN
"CPrs", "Chnl Press ", CHNL_PRS , USE_VALUE , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN

```

```
ButtonUp=
```

```

"Locl", "Local Mode ", CTRL_MSG , LOCAL , CTRL_OFF , CHN_MIN
"Omni", "Omni Mode ", CTRL_MSG , OMNI_OFF , UNDEFINED , CHN_MIN
"Mono", "Mono Mode ", CTRL_MSG , POLY_ON , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Sust", "Sustain ", CTRL_MSG , SUSTAIN , CTRL_OFF , CHN_MIN
"Port", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_OFF , CHN_MIN
"Sost", "Sostenuto ", CTRL_MSG , SOSTENUTO , CTRL_OFF , CHN_MIN
"Soft", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL , CTRL_OFF , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"Undf", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN

```

```
ButtonDown=
```

```

"LOCL", "Local Mode ", CTRL_MSG , LOCAL , CTRL_ON , CHN_MIN
"OMNI", "Omni Mode ", CTRL_MSG , OMNI_ON , UNDEFINED , CHN_MIN
"MONO", "Mono Mode ", CTRL_MSG , MONO_ON , MONO_CHNL , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"SUST", "Sustain ", CTRL_MSG , SUSTAIN , CTRL_ON , CHN_MIN
"PORT", "Portamento ", CTRL_MSG , PORTAMENTO , CTRL_ON , CHN_MIN
"SOST", "Sostenuto ", CTRL_MSG , SOSTENUTO , CTRL_ON , CHN_MIN
"SOFT", "Soft Pedal ", CTRL_MSG , SOFT_PEDAL , CTRL_ON , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN

```

```
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
"UNDF", "Undefined ", UNDEFINED , UNDEFINED , UNDEFINED , CHN_MIN
;
; End of MIDIBARS.INI
;
```


