

Pulsar Projects

Overview

The Pulsar Icon On The
Windows Task Bar

Pulsar Settings

Presets / Number Of Voices

MIDI Controller Assignment

Controller Settings

DSP Load

Working In The Project Window

Hardware Input/Output

Software Input/Output

Pulsar And Cubase

Hard Disk Recording

[Return To Main Table Of Contents](#)

Overview

This chapter picks up where the Introduction left off. It assumes that you've been through the Introduction and have a reasonable understanding of the basic operating features of Pulsar. First, a few **additional program features not covered in the Introduction** are presented. This is followed by a discussion of **working in the Project window**. Next, the various **hardware and software inputs** and outputs are described. Finally, examples of the **use of Pulsar with external software** are given.

The rest of this Overview provides some additional background relevant to working with projects in Pulsar.

Devices And Modules – The Project Perspective

In Pulsar, all *devices* and *modules* exist as files visible in the File Browser. You bring them into a project by dragging them from the File Browser into the **Project Window** (or alternatively, the Rack Window).

Devices (.dev files) include **synths**, **mixers** and the like. They are relatively "large" and complex and have **device surfaces** with displays, knobs, switches and faders, etc.

Modules (.mdl files) are generally simpler items such as **audio and MIDI inputs and outputs** (hardware or software), most of which have **no controls**. These are

also sometimes referred to using the computerese term "drivers", especially when they represent a piece of interface software between Pulsar and the "outside" world.

(The .mdl extension is also shared by Modular Synth module files, whose onscreen manifestations also share the use of the term "module". This is mentioned here merely to spare you some possible later confusion. These modules are strictly special accessories of the Modular Synth device which can be used *only within* that device.)

Projects And Project Files

The scope of a **project** includes the selection of hardware inputs and outputs to be used, whether a mixer is to be loaded, which synthesizers will be used and whether, for example, ASIO or Wave drivers are needed. Also included are the basic settings and screen positions of each of these elements, as well as all connections between them. The same goes for device preset lists, MIDI controller assignment lists, the audio synchronization (wordclock) configuration and various other Pulsar settings. A project is thus your complete Pulsar **setup** of the moment.

All of this information is stored in **project files** (identified with the .pro extension), using the Save Project (As) option of the File menu. You can also **load projects** via

this menu, or by dragging them from the File Browser into the Project window (switch on the **Pro** button in the File Browser to make project files visible).

Just as important as what a project file includes is what it *doesn't* include – namely, the devices themselves. Instead, the project file includes **references** to the corresponding disk files. These references take up virtually no space in a project file. They prevent project files from becoming inconveniently large and avoid redundant storage of device files within every project.

The use of referenced devices also keeps devices and projects separated from one another. This has implications for **device preset lists** and **MIDI controller assignment lists**, which are **stored within the device files** themselves. **Changes to these lists** for a particular device will **affect a device in every project** where it is used, since all of these projects refer to the same device file.

Since a device reference points to a file in a specific disk location, moving your device files to a different location can cause the references to go looking in vain. This might also occur with a project file created elsewhere than on your own system – for example, if Pulsar is installed on drive C: in one system and on drive D: in the other. When

you attempt to load such a project, Pulsar will open a dialog asking you where it should look for the devices which it can't find on its own.

One **glaring exception to the referenced devices system** is worth pointing out: The Pulsar Modular Synth is stored *directly* in the project file when a project is saved. The use of a reference is impossible with this synth, since (as a modular, user-modifiable device) its structure is assumed to be substantially different from one project to the next.

The Pulsar Icon On The Windows Task Bar

You'll probably have noticed this pretty quickly once you've gotten up and running with Pulsar, and may be wondering just what it means. In fact, it's your visible indication that a Pulsar **background project** is loaded and active in your system.

The purpose of this background project is to **set up the Pulsar card to serve as a default sound and MIDI device** in your system. Standard Windows sound playback (from multimedia programs, etc.) can thus occur via the Pulsar card as if it were a standard sound card, even when the Pulsar program itself is not running. Likewise, the Pulsar card's MIDI input and output become visible and accessible to Windows programs which use MIDI. Other default system connections are made as well. A complete picture of the devices and connections in the background project is presented in the Installation Guide.

Right-clicking on the icon opens a small context menu as shown at right. Via this menu, you can get direct access to the Pulsar **Sample Rate Settings** control panel (described in the *Pulsar Card* chapter of the manual) or **shut down the background project** (often advisable when installing a Pulsar update or other software).

When you start Pulsar, the background program is deactivated. The main program takes over full control of the Pulsar card and the icon vanishes. **When you exit**



the main program, the background program is automatically restarted to make the Pulsar card's resources once again available to Windows as before.

You can replace the automatically installed background project with one of your own creation. Refer to the following section of this chapter (Pulsar Settings) to find out how this is done.

Pulsar Settings

Under the **Appearance** options in this dialog, you can **switch off the special Pulsar cursors** in favor of the standard Windows cursor, and you can choose to have **Pulsar windows and control panels change to simple outlines while being moved**, instead of merely becoming transparent. Particularly the latter option can make Pulsar's screen response a bit snappier, if your computer is not one of the faster systems around, by skipping this bit of computation-intensive (and cool-looking, but admittedly nonessential) graphics work.

Pulsar maintains a **list of directories** in which it looks first for files of various types. A default list is created when you install Pulsar and can be modified in this control panel. You might want to do this, for example, to **set up your own folders for storing your projects**, rather than piling them all into the default Pulsar Projects folder. By changing the Projects entry to point to your new folder, you save some time and mouse-clicking in the File menu when you want to load or save one of these projects, because the File menu will then automatically open this folder when you select Load Project or Save Project.



The **size and position of the Pulsar screen** can be adjusted in the **Screen** box, which represents your entire computer screen. If you're running **dual monitors under Windows98**, activate the **Dual Screen Support** option via the checkbox to divide the Screen box into two sections representing both of your screens, as shown at right.

The **blue rectangle** inside the Screen box represents the **Pulsar screen**, which is full-screen by default but can be made smaller. Click and drag near any edge of this rectangle to move that edge. When the rectangle is smaller than the box, you can click and drag in its middle to move it around within the box. **Note that the new size and position take effect only after you close the Pulsar Settings control panel.**

You can set up the current project as a **default project** to be **loaded whenever Pulsar is started** (or when you select **New Project** in the File menu) by clicking on **Save project as default**. This is handy when you have a partial setup – e.g., a specific set of devices, modules and connections – which you often use as a starting point for new projects.



The **Save Project as Windows Standard** button causes the current project to be saved as the **background Pulsar project**. (Note: **your new project overwrites the existing background project file Winstart.pro** when you do this.)

The basic idea behind the background project is to automatically load a set of Pulsar device drivers and connections between them whenever Windows is started, or whenever you exit Pulsar. This **permits use of the hardware resources of the Pulsar card** in a logical way by other Windows programs, even **when the Pulsar application itself is not running**. For example, the Pulsar card MIDI and audio inputs and outputs can be used immediately by sequencer programs.

However, **you can save any project you like** into Winstart.pro. Thus, you can have a complete Pulsar project with synths, mixer and effects – or maybe a sample player with a set of sounds to emulate a standard GM sound card – load automatically each time you start Windows (you'll be able to use it right away, but not interact with it or modify it). But take note: a large Winstart.pro project will also prolong your Windows startup.



Presets / Number Of Voices

Presets can be easily created for any Pulsar device. These are complete device setting snapshots which can be recalled at any time. Storage, recall and management of device presets is done via the **Preset List** control panel, where you can also set the **number of voices** for a synth.

To access the preset list for a device, find the *Presets* button on its surface (it looks similar to the buttons on the control panel shown at right) and click on it. The **Preset List** control panel appears, displaying an alphabetically sorted list of the existing presets for the device. The list is automatically resorted whenever you make any changes.



To recall (load) a preset, double-click on its name.

To create a new preset from the current synth settings, click on **Save** (under Preset), answer "No" when asked if you want to overwrite the selected preset, type in a name (directly in the list) for the new "Untitled" preset and hit <Enter>. Alternatively, answer "Yes" to **update the selected preset** with the current synth settings.

To rename a preset, select it and hit F2, or just click and hold on it until the name-edit symbol ¶ appears next to it (see illustration). Now type in the new name and hit <Enter>. Or, if you change your mind, hit <Esc> to leave the name unchanged.



To delete a preset, click on the preset name in the list to select it, then click on the **Delete** button.

NOTE: Device presets created in this window are *not yet permanent!* The same goes for presets deleted or renamed here. The  and  symbols which appear next to the names of new and deleted presets serve to indicate this. In the following pages you'll find more information about device presets and how to work with them, including the use of device preset files and the two different ways of permanently saving device presets.

Number Of Voices

Via the Preset List control panel you can also set the **number of voices** a synth has. Click on the + and - buttons at the bottom of the panel or click and drag up/down on the number display itself (it's a text fader).

Changes to this setting take effect immediately. The **upper limit is not fixed** – it depends upon the particular synth, as well as upon what else is in the setup. Remember – there's no free lunch here! **Each added voice uses extra DSP capacity**, so don't set the voice count for a synth higher than you actually need it to be in the current project.

Why is this setting located in the Preset List control panel? Because, like the device preset list, the **number of voices is a modifiable property of a device** which is stored within the device file itself. As with new, renamed and deleted presets, a new number-of-voice value dialed in here is not immediately permanent. However, the permanent stored setting can be updated via the **Save (Device)** button. Read further for more information about the Save (Device) function of the Preset List control panel.



Saving The Preset List With The Device

Presets in the preset list are part of the device they belong to. However, the device itself is merely a temporary memory copy of a disk file. **Changes to the preset list – new or deleted presets – are thus likewise only temporary until they've been saved.**

Clicking on the **Save** button (under Device) is the easiest way to do this. This causes the current preset list (and the current voice count setting) to be **saved in the original device disk file**. These thus become "permanently" part of the device (at least until the next Device Save) and come along whenever the device is loaded into a project.

Important to note here is that **the Save function affects the device in every project** where the device is used. You may not always want this. For example, you may want high polyphony for a synth in one project (to meet the needs of the music) and a low voice count for the same synth in another (to "shoehorn" it into a large project).

An easy way out of this conflict is to **use multiple copies of the device file**, with different settings saved in each, instead of using the same file in every project. Device files can be copied in Windows Explorer or directly in the Pulsar File Browser and given descriptive names such as "EZ Synth 16 voices" so you can tell them apart.



Continue reading to learn about the **other method of saving device presets** – namely, exporting the preset list as a file – and about using preset list files.

Exporting The Preset List As A File

The preset list for a device can also be stored separately from the device file by **exporting** it as a **preset list file**.

Click on the **Export** button to open a standard Windows file save dialog. The preset list will be exported into a file under the name you give, with a .pre extension added automatically. Pulsar won't let you use preset files with devices which don't match them, but it's still a good idea to include some indication of the device name in the preset file name, to help you remember later what the file is for.

Preset list files let you **work more flexibly with presets** by separating them from the device (and from each other). You can organize them to suit your needs – for example, by breaking down a large preset library into categories – and avoid having to always work with your entire collection.

Exchange of presets with other users is also easy using preset files – as is downloading them via Internet – so your collections may very well grow quite large. If an updated version of a device becomes available, you can use preset files to transfer your presets to the new device. And if you use the trick (described on the previous page)



of working with multiple instances of a device, preset files let you use your presets with all copies of the device, regardless of which copy the presets were created with.

Read further to learn about **working with preset list files**.

Working With Preset List Files

Preset list files are accessed via the **Preset List Files drawer** of the Preset List control panel. If a file was in use when the *device* was last saved, you'll see this preset list as soon as you open the drawer. Otherwise, you can open a file by clicking on **Load/New**. In the file open dialog which appears, you can select an existing preset list file or type in a name to create a new file.

You can now **import individual presets from the file** by simply dragging them from the Preset List File window into the main Preset List window. You can **load a preset directly from the file** without importing it by double-clicking on the preset name in the Preset List File window.

Making changes to the preset list file is just as simple. Drag a preset from the main Preset List window into the Preset List File window to add it to the file. Click on a preset name in the Preset List File window, and then on the Delete button in this drawer, to delete the preset from the file.

Note that there is **no Save button** in the Preset List File drawer. When you add presets to a file or delete them from a file, the **changes are saved immediately to disk**.



MIDI Controller Assignment

Any knob, switch or fader on a Pulsar device surface can have a **MIDI Control Change** message assigned to it. This allows it to be adjusted not only onscreen via the mouse, but also under external (to Pulsar) MIDI control. In response to these messages, onscreen controls move just as if you were using the mouse to adjust them. But the parameter control response stays active even when the device surface (or Pulsar) is minimized.

MIDI Control Change Message Sources

The **Mod Wheel** on a MIDI keyboard is the most obvious example of a MIDI Control Change message source. **MIDI fader boxes** are another. These are perfect, for example, as a quicker and more natural alternative to the mouse for controlling multiple faders on a Pulsar mixer. However, the controller message source can also be a **sequencer program** on the same computer, with its MIDI output routed into Pulsar – or an external sequencer.

It Works Both Ways

Pulsar device controls with MIDI controller message assignments don't only respond to *incoming* messages. **Corresponding messages are also sent out** (via the Pulsar MIDI Output) when you adjust controls with the mouse. These messages can likewise be sent to a

sequencer for recording and subsequent playback (as a form of **MIDI-based control automation**) or used directly for live control of external MIDI devices.



Making MIDI Controller Assignments

Right-click directly on any knob, fader or switch. The **MIDI Controller Assignment** panel pictured above appears. If the control already has a MIDI controller assignment, its message number is displayed in the **Current** field.

Now simply *move* the external MIDI controller (mod wheel, fader, data slider etc.) until its message number appears in the **New** field. Then click on **Assign**. Finished!

Click on **Reset** to clear an existing MIDI controller assignment.

The **Ctrl List** button opens the Controller Settings panel, which is covered in the following section.

If nothing appears in the New field when you move the external controller:

- Make sure that the Pulsar device (synth, mixer, etc.) is receiving on the MIDI channel that the controller is sending on. (Note that if the control messages are coming into Pulsar by way of a sequencer, the sequencer could also be "bumping" them to a different MIDI channel.)
- Make sure that the MIDI input of the Pulsar device is cabled to the MIDI message source within Pulsar (Pulsar MIDI Input or Sequencer MIDI Out).
- Make sure that transmission of controller messages from the external controller hasn't been disabled.



Controller Settings

This control panel can be opened directly from the Window menu, or by clicking on the *Ctrl List* button in the MIDI Controller Assignment control panel (see preceding section). Like that control panel, this one lets you manage MIDI Control Change message assignments to device controls – but with more of an overview. It also gives you access to **Controller Presets**.

In the window, a list of all available MIDI controller numbers is shown. Any device control which already has a MIDI Control Change assignment is listed by name next to the MIDI controller number which is assigned to it.

To **assign a MIDI control change number** to a device control (knob, fader, switch, etc.) via this control panel, first **click on the control itself** to select it. Then:

- Scroll to the desired controller number in the Controller Settings window and double-click on it. The name of the device control appears next to the number, indicating that the assignment has been made.
- **Or**, if the control change message is coming from an external MIDI controller (mod wheel, fader box, etc.), simply move the controller. The number of the message it sends will appear in the **Learn** box and highlighted in the list. Click on the highlighted number to confirm the assignment. If nothing happens when you move the



controller, refer to the preceding *MIDI Controller Assignment* section under the subheading **Making MIDI Controller Assignments** ("If nothing appears ..") for possible causes.

To **delete an existing assignment**, double-click on it.

It's important to be aware that **each device has its own list of MIDI controller assignments**. The name of the device whose list is currently displayed appears directly above the list. If you click on controls of a different device, the list switches automatically to show the controller assignments for that device.

Associated Settings

The **Use device MIDI channel** option is on by default and causes the controller assignments for a device to be active on whatever MIDI channel the device itself is currently set to. (If this option is turned on, adjusting the **Channel** setting in this control panel will thus also change the device MIDI channel setting and vice versa.) If you turn this option off, the two settings are uncoupled. You can then **select a separate MIDI channel for the MIDI controller assignments**. For all other MIDI messages, the device will continue to use its own channel setting.

For each assigned MIDI controller number, the **Modulation Range** slider can be set to tailor the range of device control response to the MIDI controller message. Click on a control to see its current Modulation Range slider position, which you can then adjust.

Storage Of the MIDI Controller Assignment List

The list of MIDI controller assignments and associated settings for each device in a project is **saved with the project** when the project is saved. Additionally, the current assignments are **saved in the device file itself** when the device is saved via the Preset List control panel (refer to the appropriate section earlier in this chapter).



DSP Load

The **DSP Load** control panel is like a tachometer for your Pulsar card. As with a tachometer, there's nothing you can actually *do* with this control panel, but the information it gives you can be useful.

The **bar graph display** shows you approximately what proportion of the **DSP capacity** of your Pulsar card is currently being used, and thus also how much **unused capacity** is left. This depends upon the current project and configuration. Leftover capacity means you can expand the project – you can add synth or mixer devices or effects, or increase the number of voices of an already-loaded synth, or make use of additional mixer channels or mixer channel EQs which are currently turned off.

Unlike a tachometer, the DSP Load display *can* be safely ignored. There's no possibility of damaging the Pulsar card. Pulsar uses the info to prevent DSP overload anyway, thereby ensuring that all loaded devices work properly. And there's no problem with running "in the red". Pulsar devices work the same under all load conditions, as long as there's enough DSP capacity for them.

For The Technically-Minded

The additional displays on this control panel identify your installed DSP card by model name and indicate the number of DSPs on the board (four for Pulsar).



The slide-out drawer at right shows the capacity usage **for each DSP individually**, either as a **DSP cycle count** or as a **percentage** of the maximum (click on the % sign).

By watching how the displays change as you add and delete various devices or enable and disable various device features, you can get an idea of the associated "power consumption" of each. This can help you "trim the fat" from your projects and get the most out of your card.

DSP capacity is directly related to available computing time, namely the duration of one sample clock. It follows that extra DSP capacity becomes instantly available when the sample rate is lowered – about 50 percent more, in fact, if it goes from 48 KHz down to 32 KHz. However, this also shrinks the audio bandwidth. This may be noticeable with audio and samplers. Also, some audio sequencers and external devices can't work with a lower sample rate. Refer to the *Sample Rate Settings* section of the *Pulsar Card* chapter of this manual for more details.

Working In The Project Window

Whereas device surfaces are the front panels of Pulsar devices, the **Project window** can be thought of as the collective rear panel for an entire Pulsar project. More importantly, it's the place in Pulsar where you can see all and do all.

The Project window is opened via the Window menu, if it isn't already open. It can be maximized to fill the screen (unique to the Project window), minimized to an icon, or if not needed at the moment, closed completely.

This window gives you a detailed **schematic** or "**circuit diagram**" view of *all* devices and connections in a project. You can directly see and manipulate these connections, add devices to a project or remove them from a project.

By the way, the **Project and Rack windows are tightly coupled** – not surprising, since they're merely two different representations of the same thing. Changes made in the Project window – addition or removal of devices or connections – show up immediately in the Rack window as well.

In the Project window, both devices and modules (as defined in the Overview section of this chapter) are shown as simple "green boxes". Each green box is only as large

as necessary to fit the device or module name in its middle and the requisite number of **pads** on its edges to represent the inputs and outputs of the specific device or module.

For example, the Project window view of a synth may include only one MIDI input pad (designated **MIn**) and one audio output (**Out**) pad. (A few other common pad designations: **In** for audio in, **MOut** for MIDI Out, **LIn/RIn** for left/right audio in, **LOut/ROut** for left/right audio out). In the Project window you add or remove point-to-point **cables** between these pads to create or modify a setup.

And now, a bit of bad news (well, not *that* bad) – namely, yet another definition for that overused word "module". Specifically: these "green boxes" are referred to in the shorthand jargon of the Project window as – yes, you guessed it. (There – that wasn't so bad, was it?) Hopefully, the context should make this clear. If not – it doesn't really matter, there isn't much that can go wrong here.

Modules In The Project Window

When a project is loaded, the devices and modules it contains appear in the Project window as modules (yes, here it comes). Naturally, the connections between these modules, as defined in the project, appear here as well.

Devices and modules are added to a project via drag-and-drop from the File Browser into this window, as described in the *File Browser* section of the *Introduction* chapter. Make sure the **Dev** and **Mdl** buttons in the File Browser are pushed in, so that these file types will be visible in the File Browser.

(Incidentally, you can also **drag-and-drop an entire project** into the Project window, as an alternative method of loading a project.)

Modules in the Project window can be **moved around as desired** via simple click-and-drag, to clear up the project view or just for fun. All existing connections to a module hang on tight and go along with it when it is moved around.

Each module in the Project window has a **module menu** which is accessed by right-clicking on the module. The options in this menu are then selected via normal left-clicking. At a minimum, this menu will contain the option **Remove module** or **Remove device** (thus providing us, finally, with a form of rescue from the confusion of "module" terminology – if you really want to know, once and for all, whether the module you're looking at in the Project window represents a device or a module, the module's module menu will reveal this to you. *Whew!*)

For modules which really *are* modules (mainly Pulsar hardware or software inputs or outputs, most of which have no controls), *Remove module* is usually the only option. For modules which represent devices (synths, etc. with full control surfaces), the other usual options allow you to **open the device control surface** or **the device preset list** (or to close them, if they're already open).

A couple of other things you can do with modules in the Project window:

- You can **delete a module** by left-clicking on it once and then hitting the Delete key on your keyboard.
- You can **open a module's device surface** (where applicable) by double-clicking on the module.

When you delete a module, all connections to that module are also deleted. You've probably already noticed this. You may have also noticed that there is (unfortunately) **no Undo** function in the Project window. So – be careful, don't work too quickly, and save your project often.

Pulsar Cables

Devices are interconnected using *virtual cables*. In the Project window these are shown graphically as cables.

Virtual cables are nicer than the real kind in a number of ways. They weigh nothing and take up no space in your studio. You can't trip over them and your friend can't borrow them and "forget" to bring them back and your cat can't chew them up. They don't "go bad" and they won't pick up transmissions from the idiot next door with the 1000-watt CB rig. You always have "just one more" when you need one, it doesn't cost anything extra, it's just the right type, and – it's long enough.

Cables are used to connect device pads to one another. To **make a connection**, you click on a source pad (such as Audio Out), and then on a destination pad (e.g., Audio In). A new cable appears onscreen to represent the new connection, which instantly becomes active in the setup. You can **remove a connection** in the same way – click on two pads which have a cable connecting them, and the connection vanishes.

Cabling Tips

Basically, working with Pulsar cables is just as simple as described above. But, there are a few useful tips worth knowing. These are presented here, in no particular order:

- Rather than clicking on pads directly, you can **click on their text labels**. It works just the same and it provides a larger target which can be easier to hit, especially when there are already cables connected to a pad.
- When connecting or disconnecting two pads, you can **click on either one first** – the order doesn't matter.
- An easier way to get rid of a connection is to **click on the cable itself**. It becomes highlighted (so you can make sure you've hit the right one). Hit the Delete key on your keyboard to make it go away.
- Pulsar **allows only connections which "make sense"**. It won't let you connect outputs to outputs, nor (by consequence) two outputs to the same input, and it won't let you connect audio pads to MIDI pads.
- Pulsar **will let you connect inputs to one another**. This merely means that any signal source which is connected to one of these inputs will feed the other connected inputs as well.

- Cables appear in different **colors** to let you know what type of signal they carry (red for MIDI, blue for audio).
- **The mouse cursor gives you clues** as to where you stand while cabling (here it is assumed that you're using the Pulsar graphical cursors and haven't switched them off, via the *Pulsar Settings* control panel, in favor of the standard Windows cursors):

Normally, the cursor looks like this:



When you move the cursor over a pad (or over its text label) it changes to a black plug to indicate that you can initiate a cabling action via a left-click:



When you actually click, the cursor briefly does a little twist. Now, when you move it away, it changes to a red plug:



This is your signal that the cursor is "armed" and ready to complete a connection.

As you continue to move the cursor around, you'll see the black plug again whenever you position it over a pad which can be connected to the first pad. Click when you see this and you'll make the connection.

On the other hand, if you position the cursor over a pad which is *not* an acceptable candidate for connection to the first pad (for any of the reasons listed on the previous page), you'll get this cursor:



You may have also spotted this shape when you clicked on the first pad, before you moved the cursor away from it. This is just Pulsar's way of pointing out that you can't connect a pad to itself.

If the cursor switches to this shape:



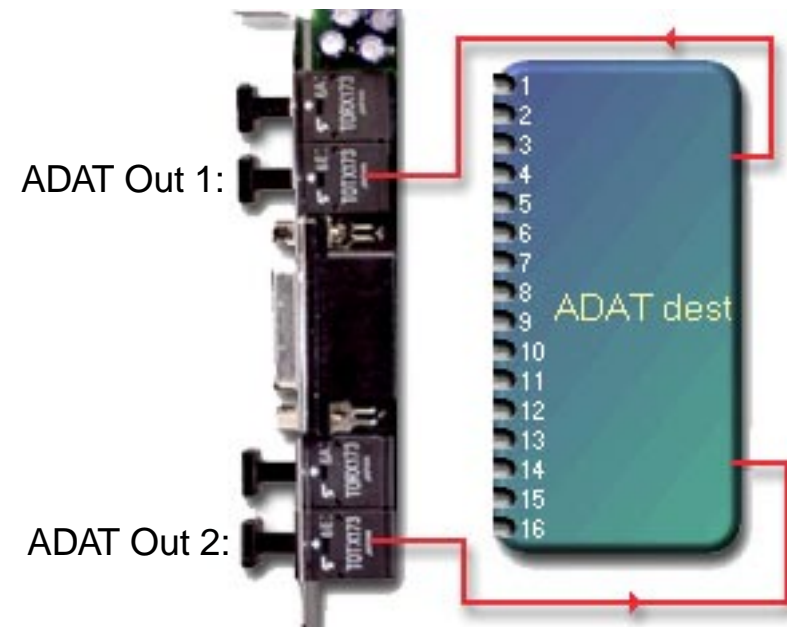
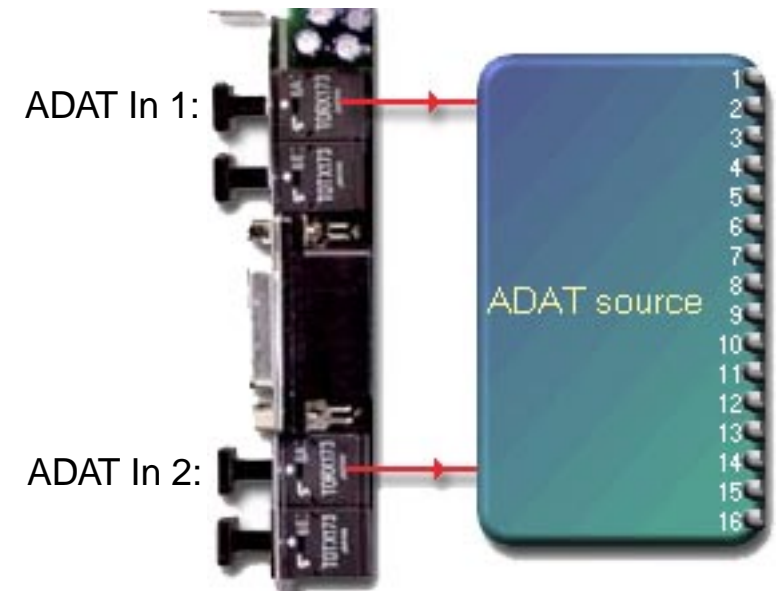
it means that the pad you're aimed at already *is* connected to the first pad. Clicking will disconnect it.

Hardware Input/Output

Hardware inputs and outputs include all of the actual *physical* inputs and outputs on the Pulsar card: analog, S/P-DIF, ADAT, MIDI. These are "virtualized" as modules which you can drag from the File Browser into the Project window. By making onscreen connections to these modules, you're actually routing signals to and from these inputs and outputs.

These modules are designated as **Source** and **Dest** (destination). To keep this straight, remember that they're named *from the point of view of the Project window*. Thus, a Source module brings signals *into* Pulsar from outside (and so has *output* pads for hookup in the Project window), while a Dest module carries signals *out* of Pulsar (and has *input* pads to which Pulsar signals can be routed).

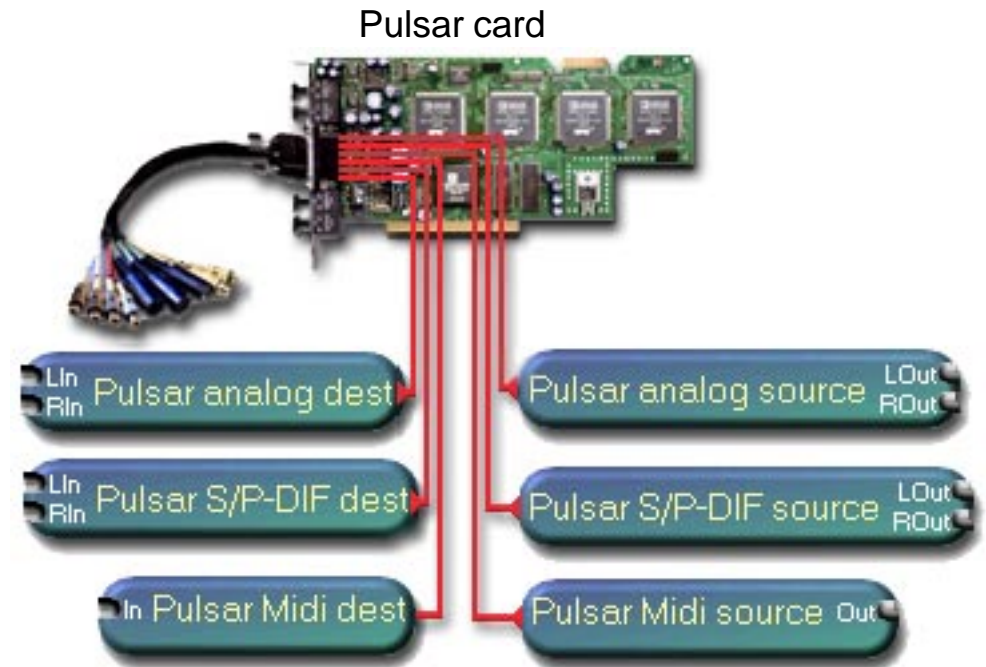
While there are actually two sets of ADAT I/Os on the card itself, each with 8 input channels and 8 output channels, ADAT I/O is represented in Pulsar by **Pulsar ADAT Source** and **Pulsar ADAT Dest** – that is, one input module and one output module, each with 16 channel signal pads. An ADAT connected via ADAT I/O 1 appears on channels 1-8 of both modules. The same applies to channels 9-16 of both modules for an ADAT connected via ADAT I/O 2. However, there's no requirement that all of these signals be used. If you're using just one ADAT with Pulsar, you can use just the appropriate module pads and simply leave the others unconnected.



The signals which are accessible via the **Cable Assembly** are likewise represented in the software by modules.

The modules for **Analog Input**, **Analog Output**, **S/P-DIF** (digital) **Input** and **S/P-DIF Output** each feature left/right signal pairs on a single module. Within Pulsar, these signals can of course be used as stereo pairs or as independent mono signals as desired. Likewise, you can bring a mono audio signal into Pulsar via one Analog Input and leave the other one unconnected – both in the software and in the "real" world.

The Pulsar modules for the **MIDI Input** and the **MIDI Output** are separate from one another. The **MIDI Thru** connector is not virtualized as a Pulsar module, since it receives its signals via a direct hardware connection to the MIDI Input.



Software Input/Output

Software inputs and outputs include **all audio and MIDI connections to other Windows programs running on the same computer** – MIDI and audio sequencers, wave/media players of various types, sound editors, etc. – as well as to non-Pulsar audio and MIDI hardware.

These connections permit MIDI and/or audio data streams to flow within the computer as if conducted via cables. This is what allows you, for example, to use a MIDI sequencing program to sequence your Pulsar synths, or to mix down audio tracks from a hard disk recording program using the Pulsar mixer, with MIDI mix control – or both at once – on *one* computer, without cables.

As with the Pulsar card hardware I/Os discussed previously, the software I/Os show up in Pulsar as **modules** which can be dragged into the Project window (from ..\Pulsar\Devices\Windows Devices) and cabled. Each one is designated as **Source** or **Dest** (destination). **To keep this straight, remember that they're named from the point of view of the Project window.** Thus, a Source module brings data *into* Pulsar from outside (and so has *output* pads for hookup in the Project window), while a Dest module carries data *out* of Pulsar (and has *input* pads to which Pulsar signals can be routed).

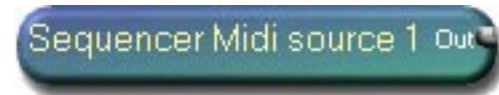
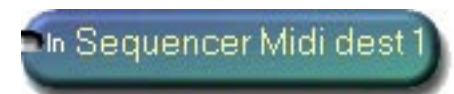
Outside of the Project window (or Pulsar) context, the software connections are referred to as *drivers*.



MIDI Connections To Other Programs

The **Sequencer MIDI Source** and **Sequencer MIDI Dest** modules are Windows MIDI drivers which can be accessed from outside of Pulsar. By default, Pulsar provides two such inputs and outputs for MIDI communication with other Windows programs. Your sequencer program will list them as "Pulsar MIDI In 1" and "Pulsar MIDI Out 1". If you add another Sequencer MIDI module to your project, it will be designated as "2".

In the default Pulsar configuration, two MIDI inputs and outputs are registered in Windows. If you add yet more Sequencer MIDI modules, you'll need to go into the Windows Device Manager and increase the number of Pulsar MIDI drivers registered there (and then restart the computer) before the additional Sequencer MIDI modules can be used. To get there, right-click on *My Computer*, select *Properties* in the menu which appears, then select *Device Manager*. Double-click on *Audio, Video and Game Controllers*, then on *Pulsar*, and finally select *Settings*, where you can change the number of Pulsar MIDI input and output "devices". Don't forget to restart Windows!

A blue, pill-shaped button with a gradient from blue to green. The text "Sequencer Midi source 1 Out" is written in a yellow-green font. A small black square icon is at the right end.A blue, pill-shaped button with a gradient from blue to green. The text "In Sequencer Midi dest 1" is written in a yellow-green font. A small black square icon is at the left end.

General Purpose Software Audio Connections

The **Wave Source** and **Wave Dest** modules for Windows audio communication are, like the Sequencer MIDI modules, assigned numbers sequentially as they are added to a project. If you go beyond four Wave inputs or outputs (i.e., two Source or Dest modules – they're stereo), Pulsar will make the Settings changes for you in the Device Manager (but you'll still need to restart Windows before you can use the additional I/Os).

The Wave drivers are provided in 16-bit ("standard") and **24-bit** versions. The 24-bit versions preserve full fidelity and dynamic range when used to transfer multichannel mixes, etc., into or out of Pulsar. However, they should not be used unnecessarily, as each 24-bit Wave input or output consumes twice as much data transfer capacity on your computer's PCI bus as its 16-bit counterpart.



Pulsar also provides **Sound Card Source** and **Sound Card Dest** modules which permit direct audio communication between Pulsar and a standard PC sound card. Thus, you don't *have* to throw your existing sound card away when you install Pulsar. Instead, you can integrate it into the Pulsar system and continue to use it for audio input and/or output.

However, it should be noted that audio connections via **standard sound cards are prone to relatively high delays** (latency) compared to the inputs and outputs of the Pulsar card. This means that they are **suitable for general-purpose non-time critical uses** (talkback mike or other auxiliary audio source input, cue/monitor output) but not necessarily for use as an effects loop interface to external effect devices.

Via the **control surface** of the Sound Card Source and Sound Card Dest modules, you can **specify sample rates and signal sources** to be used with the card.

By means of the **Direct Sound Source** module, Pulsar can serve as a substitute sound card for programs (e.g., games) which deliver their audio output via the **DirectX** sound interface. (There is no corresponding dest module.)



ASIO Software Audio Interfacing

These modules are provided for audio communication between Pulsar and programs which use the ASIO interface model (see the next section, *Pulsar And Cubase*, for more information on this topic).

The ASIO drivers are provided in 16-bit ("standard") and **24-bit** versions. The 24-bit versions preserve full fidelity and dynamic range when used to transfer multichannel mixes, etc., into or out of Pulsar. However, they should not be used unnecessarily, as each 24-bit ASIO input or output consumes twice as much data transfer capacity on your computer's PCI bus as its 16-bit counterpart.

Unlike most Pulsar I/O modules, the **ASIO Source** and **ASIO Dest** drivers have small control surfaces accessed via the right-click-on-module menu. These let you set the number of ASIO input or output channels (up to a maximum of 16 *stereo* channels). Channels are added or removed dynamically – the module resizes itself onscreen accordingly as you change the setting, as shown in the illustration at right (here, the module has eight stereo channels – half the maximum possible number).

Note that changes to the number of ASIO channels are best done **before** the associated external audio program is started.



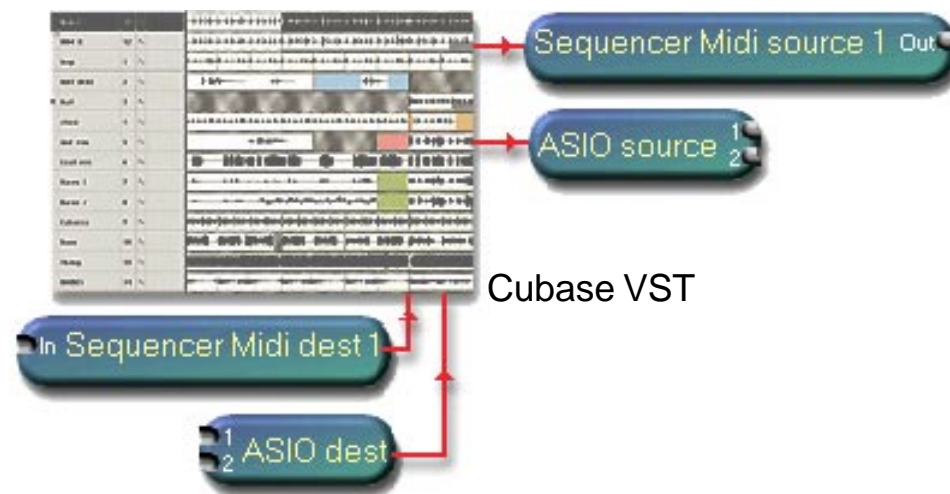
Pulsar And Cubase

Because it uses ASIO drivers instead of standard Windows Wave drivers, Cubase VST is an attractive option for sequencing work with Pulsar. ASIO drivers such as those provided by Pulsar permit more direct access to the system hardware used in the hard disk recording process and can therefore transfer audio data more efficiently to and from other programs. When using Pulsar with programs which are not ASIO-compatible, longer latency times (delays) in the audio tracks can be expected.

A brief overview of the use of Cubase with Pulsar is provided here. **For full details, you should refer to the Installation Guide** found on your Pulsar CD.

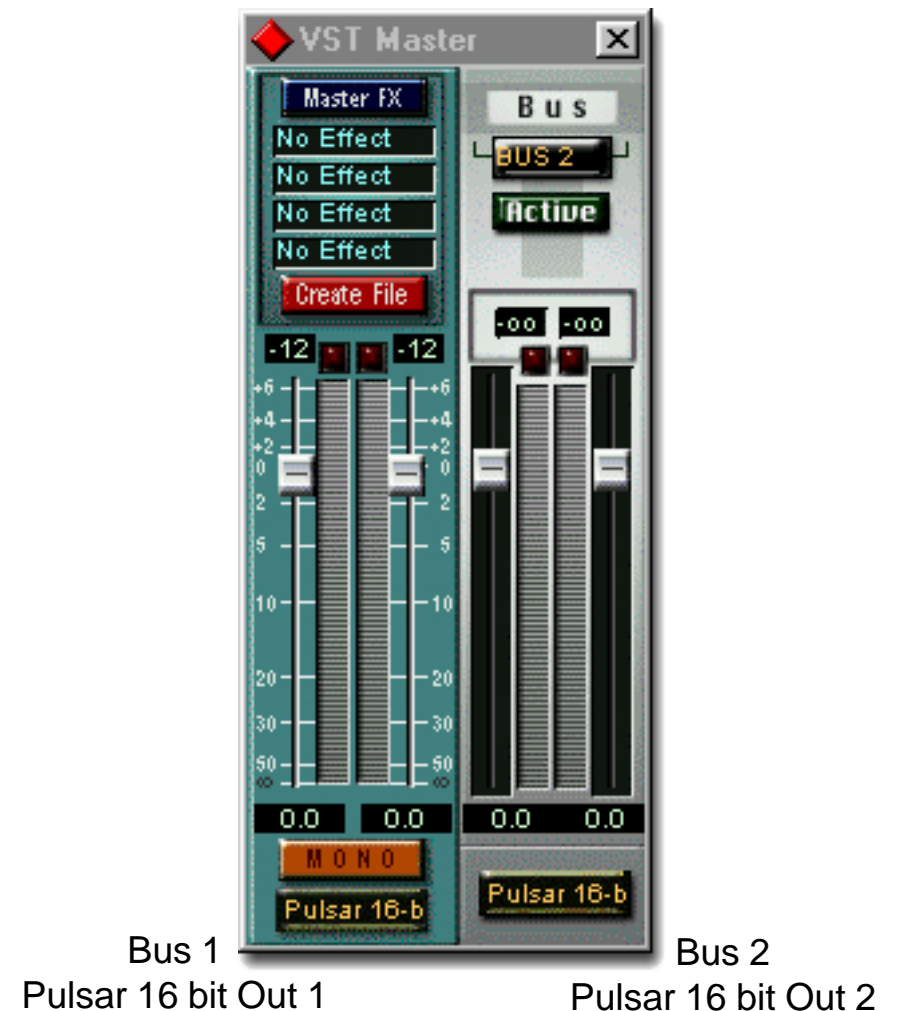
The signal flow when using Cubase VST together with Pulsar is diagrammed at right. Audio and MIDI originating in Pulsar, whether from Pulsar synths or the Pulsar card inputs, exit Pulsar via the ASIO Dest and Sequencer MIDI Dest modules and arrive in Cubase. Signals from Cubase enter Pulsar via the respective Source modules. The Sequencer MIDI Dest module appears in Cubase as a MIDI input (Pulsar MIDI In 1), ASIO Dest as a pair of VST inputs (Pulsar 16 bit 1L and 1R). The Dest modules appear correspondingly as outputs in Cubase.

Cubase VST must be started *after* starting Pulsar, so that the Pulsar ASIO and MIDI drivers can be in place when Cubase starts.



ASIO outputs are represented in Cubase as **busses**. The first bus is defined as the **master** (appears in green in the image at right). Any additional outputs/busses appear to the right of the master.

In this example, the ASIO Source module in Pulsar is set for a channel count of four. **If you change this setting in Pulsar, the change is immediately relayed to Cubase and registered there. In the process, the existing bus settings in Cubase are lost.** Therefore, for every song you work on in Cubase, you should make sure to save a corresponding Pulsar project.



The **Cubase Remote Control** allows the remote control of Cubase VST from within Pulsar. Since it achieves this through the use of standard MIDI messages (which are transmitted via the Sequencer MIDI Dest module to Cubase), and since the MIDI message for each of its buttons is redefinable, the Cubase Remote can also be used for remote control of other programs. It comes preconfigured to match the default Cubase VST remote control setup.

To use the Remote, go into the **Remote Control** dialog in Cubase and turn on the **Remote active** option. To avoid inadvertent disabling of remote control, the **Remote key** option should be disabled. It shouldn't be necessary to modify any of the MIDI message settings if they are still in the program default configuration.

To reconfigure the Remote for other uses, open the **Command Assignment** slide-out drawer. There you can specify the type of message (Note, Controller or Program Change) as well as the specific message value to be sent for each button. The MIDI channel on which these messages are sent can also be adjusted (one setting applies to all messages).



Hard Disk Recording

Hard disk recording using Wave drivers works fundamentally the same in Pulsar as otherwise. All that is required is an **appropriate setup project in Pulsar** and the corresponding settings in your recording software.

Pulsar should always be started first, followed by the recording program, to ensure that the Pulsar drivers required (and expected) by the recording program are already in place before it starts and maintain an uninterrupted presence.

Note that the requisite drivers can also be a part of the **background project** which manages the configuration of the Pulsar card when the Pulsar application is not running. This is workable if you just want the Pulsar resources in place to support recording and won't need to work directly in Pulsar (e.g., to operate the BigMixer) during your session. But **if you need to start Pulsar, close the recording program first and then start it again** when Pulsar is running.

Detailed information concerning the setup of Pulsar and a few of the leading audio sequencing programs for combined use is provided in the **Installation Guide** found on your Pulsar CD. Additional information on the background project can also be found there, as well as in the beginning sections of this chapter of the manual.

Audio/MIDI sequencing program

