

Mapping Rack Extensions for Panorama

Here are the locations of the files which determine how devices in Reason are mapped to Panorama.

- Windows: **C:\ProgramData\Propellerhead Software\Remote\Maps\Nektar**
- OSX: **//Library/Application Support/Propellerhead Software/Remote/Maps/Nektar**

Make sure to back-up your existing files and keep back-ups of any changes you make. If you run one of our installers, it will overwrite any changes to the files in the local

What you need to create your own map...

First, I recommend using an IDE instead of Text Edit or Note Pad. I use Programmer's Note Pad in Windows and XCode in OSX. Both of these are very important when it comes to troubleshooting errors in the remotemap files.

The first line in the template is the 'scope'. This is how Reason determines which device the mapping is related to.

This is what appears in the RemoteTemplate file:

```
Scope Propellerhead Software se.propellerheads.example
```

The important parts of this are the name of the RE developer and the scope. You can get this information by exporting the Remote information from the rack, open the 'File' menu and select 'Export Device Remote Info'. Here is an example (using Rob Papen's Quad)...



You can see the developer name is highlighted in red and scope is highlighted green/yellow.

So you want to replace the 'Propellerhead Software' with the dev name for the RE and the 'se.propellerheads.example' with the scope for the

```
Scope Rob Papen com.robpapen.Quad
```

After making a change to the 'Panorama.remotemap' file, you will need to reload the 'Nektar Panorama' control surface. Simply go to the 'Core' menu, uncheck where it says 'Use with Reason' then recheck it. If all went well, you should see a green check mark. If there is an error in the remotemap, you will see the red 'X' click on it and you should see an error message which indicates which lines have problems. If you are using an IDE, you should be able to see the error very easily.

If there were no errors after making those changes, you should see the name of the selected patch when Panorama is in Instrument Mode and the selected patch. If you did encounter an error it's most likely because a whitespace was not preserved when copying the scope or dev name. Programmer's Notepad (I don't think you can in Xcode), to make sure that you haven't accidentally removed any. You can enter a whitespace character in the IDE.

Once you have the patch navigation working, you'll notice that the word 'NAME' appears in the upper-right corner of Panorama's display in real time. You'll see a comment that says '//Device name that appears on Panorama's display'. Replace NAME with the name you want to see on the display.

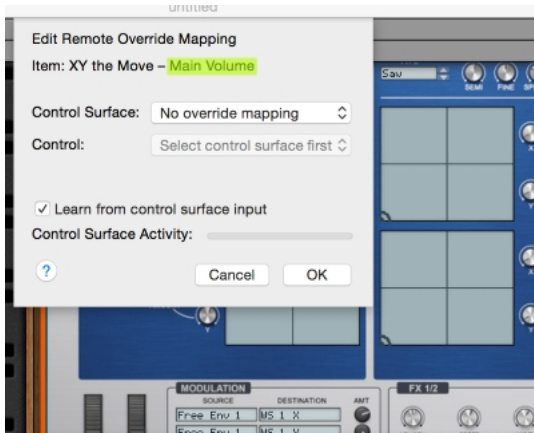
Example:

```
//Device name that appears on Panorama's display
Map Device Name "NAME" HOME
```

Quad Map

```
Map Device Name "Quad" HOME
```

Next, choose the parameter that you'd like to have mapped to fader 9 (we typically map an Instrument's master volume to this control). You can right-click on the parameter in the GUI and select 'Edit Remote Override Mapping...'. The window that appears will show the parameter name and the patch, like this:



Note: The Parameter name is highlighted green/yellow

Replace the '0' in the example, with the parameter name, like so:

Example:

```
//Fader 9 typically mapped to Master Volume
Map Fader 9    0          Instrument
```

Quad Map:

```
//Fader 9 typically mapped to Master Volume
Map Fader 9    Main Volume      Instrument
```

It's a good idea to save frequently and test any changes you've made.

The next few sections are for mapping controls and creating display labels. The Instrument home page can display the parameter names and display and the 2x4 encoders. We call these controls the 'Display Buttons' and 'Display Encoders'. The parameter names and values assigned to the current parameter bar, so custom parameter names aren't used. The other assignable controls are called the 'Faders', 'Buttons' and 'Encoders'.

In the Display Button section, you can create your custom parameter name for the first display button here:

```
//Display buttons
Map X1_Label1    ""          Instrument
```

Put the name between the quotation marks, like this:

```
//Display buttons
Map X1_Label1    "ArpLtch"    Instrument
```

Then you can define the parameter controlled by that display button by replacing the 0 with a parameter name, at this line:

```
Map X1    0          CTRL    Instrument
```

As it appears in the map for Quad:

```
Map X1    Arp Latch      CTRL    Instrument
```

The rest of the controls can be mapped the same way.;

Advanced concepts...

You may notice some maps have an extra group called Sync_Mode at the top of the map. This allows you to assign more than one parameter (value) of another control. We call these value dependent controls. They are very useful, for example, when an LFO or delay presents two different rate values (look for parameters that seem to "jump" when pressing a sync button on a Rack Extension. There are many other examples of this.

To create a modifier, you first need to create the group. Here is an example from the map for OchenK's Driver RE:

```
32230 Define Group Sync_Mode Free Sync
```

Note: 'Sync' and 'Free' are arbitrary names for variables, any names can be used. The Mode can also have any name.

Next, define the values of the parameter, which determines the state of the value dependent control. For example a Sync switch would have t

```

32242 //Sync Modifier
32243 Map Modifier1 0      Sync_Mode=Free      Instrument
32244 Map Modifier1 1      Sync_Mode=Sync      Instrument

```

Replace the CTRL mode for the control which is mapped to the sun parameter with MODIFIER1.

```

32259 Map X4      Master Sync      MODIFIER1      Instrument

```

Now you can have two different maps and labels for any control. Simply add Free to the end of a line for a parameter which should be active when the sync button is off, and Sync to the end of a line for a parameter which should be active when the sync button is on.

```

32271 Map Ch Knob 1      A Stage Duration      Name      Instrument      Free
32272 Map Ch Knob 1      A Stage Duration Sync      Name      Instrument      Sync

```

This is closely related to the way the page system for more advanced maps are created. Once you understand how this works you may want to create more complex maps to understand how the Groups relate to menu pages and navigation.