

BUGS IN VOLTAGE MODULAR

- The VST3 plugin version does not supporting any direct MIDI-out (MIDI Module does not to DAW) and also ignores any MIDI-in Change Preset messages.

Hoping that the later versions of the frameworks used for VM may address this but, if not, should be looked at as a matter of urgency as, for example, the MIDI Out module does not allow Output to DAW when VM is a VST3, and as Cubase is ending support for VST2 this means VM can't be used as for example a sequencer in a DAW

- Sequencing of availability of e.g. cable connections when preset is loaded is not obvious, with certain actions such as loading the preset (or state) sometimes triggered multiple times at different stages. It may be that some of the loading processing is needed multiple times, but there should be a way to know when everything is "final"

Example: if you process something in Preset_Loading_Finished when a preset is just being loaded and the module is being newly instantiated, the status of any jacks connected to the module in the preset are not accurately reflected if you query them during Preset_Loading_Finished (I suspect that the jacks have not been "connected" to the module at this point, being added sometime after the Preset_Loading_Finished notification.

- a. the ideal and most reliable option: when the Preset_Loading_Finished notifications are first made available (i.e. before the first sample has been started) this notification should be AFTER all the controls have been instantiated AND jack connections have been recognised, so that queries to controls, changes to controls and queries to the status of jack connections all reflect the actual situation after the preset is instantiated within VM and not just in isolation of the module;*
 - b. a workaround but also useful feature would be to add a notification e.g. Before_First_Processing or a function JustBeforeFirstSample that is "just before first sample is processed and when everything loaded and configured ready to go". My current workarounds effectively manually do this with a "firstSample" variable to allow some initial processing to happen, but a consistent way to do this built in would potentially reduce any likely overhead from such manual tricks.*
- Performance when manually driving the Digital Display control using SetBarsForDigit() as it seems to rewrite the whole display when using these calls.

Some of this could potentially be optimised (tracking what has changed on the display), and/or adding additional API calls to wrap the changes into batches, with a "prepare for display update" (stops redraw for the control) and "update display" to actually do the update processing.

- Info box does not render the mark-up/HTML from the store page. This makes the Info page look poor if the store description page has a lot of formatting.

Ideally, the Info box should fully render the text/HTML/mark-up from store when presented inside VM, so that links, etc. within store page get rendered as links, etc.

- It appears that the internal handling of knob changes in VM does not fully trap error conditions around data entry into controls, so invalid entries generate an exception in Java (trapped at higher levels). Found this in Designer but probably in the VM code as well.

From what I can work out, when a knob has a manual edit entered, it first calls ParseStringToDouble() which is itself calling parseDouble() before it sends the resulting values (String and double) to the EditComponentValue() routine, but it would appear that the ParseStringToDouble does so without a try/catch around it to handle the error condition.

BUGS IN DESIGNER

- The tooltip handling in Voltage Module Designer is substantially different from the tooltip handling in Voltage Modular and does not allow this part of the code to be tested in Designer. Visually the tooltips are very different and tooltips appear to be available for image controls in Designer but they aren't supported in Voltage Modular.

In Designer and when testing in Designer it is possible to have tooltips showing from image controls, and the text was shown in a simple black box whereas, in VM, the tooltips for the image controls do not work at all, and the tooltip is shown in a nicer rounded box with pointer AND with the control name shown.

- Designer when using JDK 21 generates "possible 'this' escape before subclass is fully initialized Compiler" error (additional code quality testing in JDK 21).

- Voltage Module Developer has many annoyances with the build/debug process not starting when requested:
 - a. Won't start when just loaded, often needing a code change (add and then remove a space) to get it to start;
 - b. When you have done a lot of changes it can take a few attempts to get the build/debug process to start;
 - c. Times it compiles and starts a debug session, but the window is hidden.
- Debug wrapper does not process (or present as jacks) some of the IO Panel information such as Velocity. This is a pain when developing modules that want to use the IO Panel inputs as defaults for some actions.

Improve the wrapper to present ALL the IO Panel features to allow testing of these without having to publish them and test them in the main app.

- When using ManufacturerProperty and GlobalProperty features and running the code in the test client it is not clear WHEN the files are created/updated following actions in the test client and, to be more challenging, it does not seem to be consistent as to when the information is written/updated.

It would be REALLY helpful to have the option to force the settings files to be created/updated whilst you are in the debugger – maybe another option on the Edit menu of the test client to “Update Global/Manufacturer Property Files”. This would allow developers to test the code which is supposedly updating the properties to make sure that the updates are really happening, which is proving to be really difficult without such a capability. This would also allow manually changes to the properties in the files to make sure that the reading of them on start up of a module is behaving correctly in all scenarios.

To make this even better, it would be really useful to have the Global and Manufacturer properties available as a list in a new tab (maybe even as two separate tabs) in the test client debugging window – maybe even editable so that you can test various scenarios without having to leave the test client.

- Designer's handling of Manufacturer's Properties File where it does not pick up the manufacturer's name to be used as the prefix, instead using “<<debugcurrentmodule>>” as the prefix – from memory, this only happens when the module has not been published but does mean that testing of these property files can be somewhat hit-and-miss
- Build numbers when submitting updates to the store still seem to get confused some time when publishing a new build and soon after publishing again – it will often show the original build number +1 in Designer although the store will show the correct one.

MODULE DESIGNER EXPERIENCE & FUNCTIONALITY

- Comprehensive “Search and Replace” features in designer, including the ability to rename controls and have code updated automatically.
- Ways to #include shared code sections that have to be in the main code (and not code modules) in a similar way to C++ and C# (obviously Java does not do this, but it could be effectively a pre-processor style approach)
- Adding more drawing tools for adding things like customisable knob surrounds and markers in addition to the outer dial but more graphical.
- Comprehensive alignment and improved grouping features for controls/labels/graphics, similar to typical design features in most SVG editors.

MODULE DESIGNER CONTROLS AND API

- Support decimal points in the existing digital display and/or add an alphanumeric digital display with decimal point support, maybe using the “dot style” display or with a sixteen segment display with decimal point options.
- Provide more options for the Outer Dial on knobs and sliders such as tick marks with/without values in both linear and log style with number of divisions set to cover e.g. Moog times, octave switches, etc.
- Continue to add a wider range of control skins “in the box” especially with designs from vintage synths.
- Increase range of fonts including specialist fonts such as Segment14 to be able to do full alphanumeric digital-style displays using normal text boxes/labels.
- Support adding custom fonts (using a cross-platform standard such as TTF or equivalent) by including them as a resource in the module.

- FileSaveDialog option to support the "Overwrite" scenario where I would like to prompt whether to overwrite an existing file or not. The FileSaveDialog option does not seem to offer anything and the MessageBox does not allow Yes/No selections. Is there a way to handle this at present? If not can this be added as a request?
- Knobs and Slider Quantization Options which would provide more control over larger and micro-movement of these controls. Use cases are:
 - a. Oscillator Tuning where the quantized steps could be the semitones but with ability to fine tune as well - in this case, three levels would be even better with "normal" movement, e.g. Shift movement for the quantized steps and Ctrl movement for very fine tuned steps;
 - b. Percentage use where normal users only need maybe 1% or 5% steps but it would be nice to allow complete flexibility for advanced use.

Two ways this could be considered:

- a. *Have Num Discreet Steps PLUS a separate Num Discreet Steps for Ctrl-Change (defaulting to same as Num Discreet Steps on knobs and older modules) ...*
- b. *Even better would be to add a "Shift Move" option to have three levels of use: Shift Knob Move typically for quantized/preset steps, Knob Move being normal operation and Ctrl Knob Move being fine tuning (high resolution) changes, maybe with the first two (or all) having the ability to have a Num Discreet Steps.*

VOLTAGE MODULAR CORE TECHNOLOGY

- Update the Java runtime to JDK21+ to fix "possible 'this' escape before subclass is fully initialized Compiler" issue using latest JDK in Designer but also to enable new functionality and any performance improvements
- Updating the frameworks used to latest versions to hopefully fix the VST3 issues and also potentially open up options for supporting CLAP, etc.

VOLTAGE MODULAR FUNCTIONALITY

- Option to enable/disable Beta modules so users (musicians and content providers) can work in "production mode" or in "beta testing mode" with access to the beta programmes they are involved with (Jan 2021)

It would be extremely useful for module developers if you could set Voltage Modular to be in a "production" mode for the current user, where it loads and uses ONLY the latest PUBLISHED versions of modules, and not any versions that are "In Development" or in any "Beta". This would be in addition to the existing, what you might call, "developer" mode which loads and uses the latest including In Development and Beta versions. There are two use cases for this need:

 - a. *any developer who also makes music (especially in a live scenario) where the ability to ONLY use the latest PUBLISHED versions is almost a must. The only way to do this currently is to have two accounts, potentially with multiple purchases as a result; and*
 - b. *developers who are wanting to thoroughly check (and fix) any backwards compatibility issues - often the only way to do this is to re-publish the current version to be able to setup a test, the publish the versions to be tested - and this can be needed multiple times, meaning going through multiple builds for any release.*
- Enabling higher sample rate for the engine (ideally so that the internal rate can match the DAW recording rate) and/or providing oversampling that is invisible to the code.
- Improved performance/load on a DAW by for example disabling physical screen redraws when the plugin GUI is not open in a DAW.

VOLTAGE MODULAR API

- Enhance the MessageBox() to allow some basic formatting of text and support for URLs, probably to be the same as mark-up available in store – and allow other inputs such as Yes/No requests.
- Make the Tooltips available on EVERY control by extending the controls that ShowTooltip() works with, so you can add additional help to ANY control, including hidden/text/label/... simply by hovering over control. Maybe also provide a programmatic way to display a Tooltip (and have access to the Tooltip timer and on/off settings).

- Surfacing more of fixed and other values as calls so that they are future-proofed, including:
 - a. `voltage.core.VoltageModule.isBypassed`
 - b. `voltage.core.VoltageModule.width`
 - c. `component.displayName`
- Adding a new event for “first sample” or automatic “first sample” state (that can be programmatically tested) as this can be important in many use cases, including around the sequencing of availability of state (listed as bug).
- Expose a way to test/query how VM is instantiated (i.e. stand-alone, VST2, VST3....) so that any limitations as a result of the way it is being used (i.e. current MIDI issues with VST3) can be better handled in modules.
- Providing a standardised set of basic oversampling support in a way that is as transparent and easy to user as possible for developers.

BUGS IN MODULES

- MIDI Output option to “Output MIDI to DAW” does not work when used as a VST3

This may be the VST3-related MIDI issue, or something else but, as it makes it impossible to use VM as. for example, and complex sequencer for other plugins via the DAW, this functionality is important to fix.

- Bug (or Unusual Behaviour) in Remote Control Module

Using Remote Control to add CV to those (few) knobs I don't offer CV on, Remote Control seems to set the knob to the VALUE and not the POSITION. So a Remote Control set to 50% will set a time knob (0-12sec) to 6sec irrespective of the mid-point (which is changeable by the user on my modules, defaulting to 750ms) which shows in its default display mode as 3 o'clock. This is not consistent with the position shown on the Remote Control's active display.

However, when controlled by a Performance Control knob, the knob is controlled exactly as the performance control knob so that Performance Control knob at the mid-point (i.e. 12 o'clock) will set the actual knob to its mid-point (12 o'clock) and not 50% of the value of the control (which may be e.g. 3 o'clock) - and it would appear (from user reports) that this is the behaviour expected.

To demonstrate that this is a bug, if you use Remote Control to control a Performance Control and then the Performance Control to control the time knob, it behaves as expected i.e. the active display of position on the Remote Control is reflected correctly in the position of the knob on the module.

- Inconsistent octave settings on oscillators and instruments and resulting limitations on ranges: there are a worrying set of inconsistencies in terms of the "fundamental" octave settings across oscillators and instruments. In some cases, it means that the module or stand-alone instrument is not accurately reflecting the vintage device it is meant to replicate and, in some cases, is unable to provide the range on the oscillator that is expected. (there was a Word document covering them all).

If nothing else, Cherry Audio agreeing standards (for all new modules) and clear documentation for developers would also be a great help.