Sonic Visualiser - Bug #1903
Parameter values do not update correctly in the UI

2019-08-02 10:49 PM - Justin Salamon

Status:	Closed	**Start date:**	2019-08-02
Priority:	Normal	**Due date:**	
Assignee:	Chris Cannam	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	3.3		

Description

Something that worked fine in previous versions (e.g. 2.0), now doesn't work in version 3.3, even though the vamp plugin being used hasn't changed:

The MELODIA plugin has a dropdown "Program" parameter that's set by default to "Polyphonic". When changing it to "Monophonic", the plugin changes the value of the "Voicing Tolerance" parameter from 0.2 to 3.0. This change used to be reflected correctly in the SV UI, but in recent versions it doesn't (nothing happens when you change the program parameter).

Steps to reproduce:
1. Install recent version of SV (e.g. 3.3)
2. Install the melodia plugin: https://www.upf.edu/web/mtg/melodia
3. Load an audio file into SV, open up the "MELODIA - Melody Extraction" plugin
4. Note that "Program" is set to "Polyphonic", and that "Voicing Tolerance" is set to 0.2
5. Change "Program" from "Polyphonic" to "Monophonic"
6. Note that the "Voicing Tolerance" parameter remains on 0.2 (should have been automatically updated to 3.0)

If you use SV 2.0 (for example) and repeat the steps above, after performing step 5, you'll notice that the value of the "Voicing Tolerance" parameter does indeed get updated in the UI.

I've only tried this on 2.0 and 3.3, so I'm not sure at which version bump the issue first started occurring.

History

#1 - 2020-04-08 05:04 PM - Chris Cannam

- Status changed from New to Resolved

This should be fixed as of commit:7bfbaaba0fe8 and the fix should be in the forthcoming v4.1.

Sorry to take so long over this. It was actually a bit deep.

The problem started when we switched from in-process plugins to plugins using a separate server process with communication using the Piper protocol. When we devised the protocol for Piper, we deliberately made plugin configuration an atomic operation (you provide all the parameter values in a single bundle) rather than having the plugin act as an object with get/set methods for parameter values as Vamp does. Within Sonic Visualiser, Piper-protocol plugins are handled just like any other Vamp plugin, via a Piper wrapper which presents a Vamp interface. The SV code calls into the Piper wrapper using Vamp-style get/set methods, and then the Piper wrapper bundles up the requested values into a single configure call when the user approves the configuration.

This works fine, except for one fatal flaw - you never get an opportunity to select a program and then query to find out what effect it had on the parameter values, as you can with a real Vamp plugin, because the program is not actually selected until the configure call, after you're done with the parameter settings.

Not many plugins use programs, so this wasn't a massively obvious problem in the UI generally, although we really should have thought of it.

(The program selection mechanism in Vamp is already ill-conceived, because it's unclear what it's supposed to mean if you select a program and change some parameters at the same time - the original design for Piper retains this flaw.)

The fix implemented is to add to the protocol so that, rather than just list a set of program names, the feature extractor also reports what set of parameter values each program corresponds to. The Piper wrapper within SV then uses this to update a local copy, so that when SV queries what it thinks is the Vamp plugin to find out what effect its program selection had, it gets the right values back. Meanwhile on the server side the actual Vamp plugin gets probed on first load, to find out what parameter values each of the programs actually sets, so that no change is needed to the plugin itself.

#2 - 2020-06-29 02:33 PM - Chris Cannam

- Status changed from Resolved to Closed