Modular - Feature Request #802
POSIX-compliance for remote communication

2011-07-02 04:44 pm - Pascal Baltazar

Status:	Closed	**Start date:**	2011-03-14
Priority:	Low	**Due date:**	
Assignee:	Théo de la Hogue	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	MVC for 0.6	**Spent time:**	0.00 hour
Branch:			

Description

As discussed on the devel list, namespace addresses should follow POSIX standard as much as possible.

e.g. for a remote address :
some_remote_device:/the/path/to/the/remote/file
with a user name :
username@some_remote_device:/the/path/to/the/remote/file

the colon becomes ambiguous in this case - it then has been discussed to change it to a ? for example, in order to comply with the URL scheme

Excerpts of the discussion below as comments

History

#1 - 2011-07-02 04:45 pm - Pascal Baltazar

#1 - first comment from http://redmine.jamoma.org/issues/740

Just copying an email Tim sent to explain the POSIX format :

" Hopefully I can help to clarify the slash situation, or at least the history behind it.

The slash is not part of the name. If it is a module, a parameter, a return, a container, etc. The names are the things between the slashes. The slashes separate the names.

So a parameter is not named "/bitdepth" it is named "bitdepth". In OSC it may be set using
/degrade/bitdepth 8
in JSON it might set using as {
"degrade": {
"bitdepth" : 8
}
}
There are no slashes in the JSON version.

Back to jcom.send -- this is an object using OSC to address a node in the tree. OSC uses slashes in style similar to POSIX directories (the stuff you see in the Mac Terminal.app for navigating the files and folders).

In the filesystem you specify a node (a file or a folder) from the root of the tree by beginning the address with a slash. You specify a node (a file or a folder) relative to your current location by not using a slash.

So if you are in a module with [jcom.send my/param/without/leading/slash] I would presume that you are addressing my/param/without/leading/slash as something deep within the bowels of this particular module. If there were a slash at the beginning it would be using the global namespace because the leading slash means to start at the root.

Have I managed to make sense? Or have managed to just make everyone more confused? I hope I am managing to help!

best,
Tim "

#2 Updated by Théo de la Hogue about 1 month ago

Another Tim's advice about POSIX format for remote devices :

" When we begin thinking about remote devices, I think it might be a good idea to again use the POSIX filesystem as a reference. Specifically, we can look at the way ssh and scp work. If I want to access a file or folder on the remote file system I use this syntax:

```
scp    /some/local/file    username@some_remote_device:/the/path/to/the/remote/file
```

If I already have some ssh magic stuff setup, then I don't need to use my username and just do this:

```
scp    /some/local/file    some_remote_device:/the/path/to/the/remote/file
```

The colon could be confusing in Jamoma because we've used it to mean something else, but maybe we can be smart about it somehow and figure out what it means from the context?

#6 Updated by Pascal Baltazar 3 days ago
Status changed from Resolved to Assigned
Priority changed from Normal to Low

Concerning the remote application/device address formatting :
the mechanism is already implemented in this way
remoteApp:/my/nice/local/path:property
which an argument for not changing it...
but maybe not a good enough argument, if there are other concerns...

as Tim wrote :
The colon could be confusing in Jamoma because we've used it to mean something else, but maybe we can be smart about it somehow and figure out what it means from the context?
That's what Théo did : he's been very smart (he usually is :-) - but then the computer has to be smart also when checking every address (and there can be a lot of them in the timespan of a second ...) :
instead of checking just one character at the beginning of the string (is it @,/ or anything else ?), it has to check if two contiguous characters (:/) are present anywhere in the string - my C++ skills are not very impressive, but I guess this makes a big difference, performance-wise...

Apart from that, I'm not sure we have to stick to POSIX so firmly because :
2/ we already made an exception for attributes, with the colon
1/ the usage is not completely the same, as this discussion show it :
Théo :
it seems that @ used to be the username in POSIX system.
Julien :
Just to make sure, when sending messages locally, we do not need to add the "localhost:" part, right ?
and this implies some potential misunderstandings....

so, to conclude, my impression is that we should stick to the KISS solution which would be :
@remoteApp/my/nice/local/path:property
which is simple to read and parse both for me and my computer...

#7 - copied from the list

On Wed, Jun 29, 2011 at 2:41 PM, Pascal Baltazar <pb@zkrx.org> wrote:

> *If we are able to stay closer to something that is a standard, then I am more comfortable doing that.*

that makes sense...

maybe then, as the separator for properties has already changed (it was :/ and it is now only :) maybe we could choose another separator for attributes ?

what would be a POSIX standard for that ? (if there is any...)
my web skills are almost as unexisting than my C++ ones, but I had the impression that "?" was used for this kind of purpose on URLs... (but it is forbidden in OSC, I think...)

Le 30 Jun 2011 à 17:27, Timothy Place a écrit :

This is a tangent, and I am probably the wrong person to ask about the importance of so-called OSC compliance, but I agree that the "?" would make a lot more sense and would be consistent with the URL addressing scheme of the internet.

#2 - 2013-06-10 05:34 pm - Trond Lossius
- Status changed from Assigned to Closed

Moved to GitHub