

inVRs - Feature #33

allow multiple plug-in paths

10/26/2010 08:19 PM - sam g

Status:	Closed	Start date:	10/26/2010
Priority:	Normal	Due date:	
Assignee:	Johannes Zarl-Zierl	% Done:	100%
Category:	SystemCore	Estimated time:	0.00 hour
Target version:			
Resolution:	fixed		
Description			
it would be handy to support multiple plug-in paths, especially if you develop your own modules.			

Associated revisions

Revision 2163 - 11/04/2010 11:21 PM - Johannes Zarl-Zierl

src/inVRs/SystemCore/UtilityFunctions.h:

- + add new inline function getPathSeparator, which returns ":" on unix-like systems and ";" on windows.
- + use getPathSeparator instead of hardcoded ":" separator in function getValidatedFullPath (fixes #33)
- + use size_t instead of long for filesize in copyFile (prevents compiler warning due to comparison between signed and unsigned types)

History

#1 - 11/04/2010 01:38 AM - Johannes Zarl-Zierl

- Status changed from New to Resolved
- Assignee set to Johannes Zarl-Zierl

Hi Sam,

This feature is implemented in svn2159. You can now use something like "/path/to/foo/lib:/path/to/bla/lib" as PluginDir.

Let me know if it works for you...

- Johannes

#2 - 11/04/2010 02:14 PM - sam g

Johannes Zarl wrote:

*This feature is implemented in svn2159. You can now use something like "/path/to/foo/lib:/path/to/bla/lib" as PluginDir.
Let me know if it works for you...*

thanks

just one note, a problem will arrive if you use a windows system and an absolute path, because then you have something like C:/path/to/foo the : will be misinterpreted. Because of that it is common to separate paths in windows systems by a semicolon instead of of colon.

#3 - 11/04/2010 04:21 PM - Christoph Anthes

sam g wrote:

Johannes Zarl wrote:

*This feature is implemented in svn2159. You can now use something like "/path/to/foo/lib:/path/to/bla/lib" as PluginDir.
Let me know if it works for you...*

thanks

just one note, a problem will arrive if you use a windows system and an absolute path, because then you have something like C:/path/to/foo the : will be misinterpreted. Because of that it is common to separate paths in windows systems by a semicolon instead of of colon.

In that case we should use different separators which are uncommon in both OS and do not lead to other conflicts, any suggestions ?

#4 - 11/04/2010 05:19 PM - Johannes Zarl-Zierl

Christoph Anthes wrote:

sam g wrote:

just one note, a problem will arrive if you use a windows system and an absolute path, because then you have something like C:/path/to/foo the : will be misinterpreted. Because of that it is common to separate paths in windows systems by a semicolon instead of of colon.

In that case we should use different separators which are uncommon in both OS and do not lead to other conflicts, any suggestions ?

Java uses either ';' or ':', depending on the platform. I guess this would be the solution with the least surprise for most people. Also, if you modify the plugin path via an external script (perhaps set it to LD_LIBRARY_PATH), this would work without further work for mangling the ':' in the unix-style path.

I would rather favour the Java-approach to the matter. Using ';' under windows and both ':' and ';' under linux would also work for me...

#5 - 11/04/2010 05:20 PM - Johannes Zarl-Zierl

- Status changed from Resolved to In Progress

- % Done changed from 0 to 70

#6 - 11/04/2010 05:25 PM - Christoph Anthes

Johannes Zarl wrote:

Christoph Anthes wrote:

sam g wrote:

just one note, a problem will arrive if you use a windows system and an absolute path, because then you have something like C:/path/to/foo the : will be misinterpreted. Because of that it is common to separate paths in windows systems by a semicolon instead of of colon.

In that case we should use different separators which are uncommon in both OS and do not lead to other conflicts, any suggestions ?

Java uses either ';' or ':', depending on the platform. I guess this would be the solution with the least surprise for most people. Also, if you modify the plugin path via an external script (perhaps set it to LD_LIBRARY_PATH), this would work without further work for mangling the ':' in the

unix-style path.

I would rather favour the Java-approach to the matter. Using ';' under windows and both ':' and ';' under linux would also work for me...

; would work for me, but make sure not to allow for both ; and : under linux, this might easily lead to confusion. The main target platform considering the user group is still Win, we want get these users confused as little as possible. Does Java allow both in Linux ?

#7 - 11/04/2010 11:16 PM - Johannes Zarl-Zierl

Christoph Anthes wrote:

; would work for me, but make sure not to allow for both ; and : under linux, this might easily lead to confusion. The main target platform considering the user group is still Win, we want get these users confused as little as possible. Does Java allow both in Linux ?

Java uses exactly one on each platform. I also implemented it this way, and will check it in now...

#8 - 11/04/2010 11:27 PM - Johannes Zarl-Zierl

- Status changed from *In Progress* to *Closed*
- % Done changed from 70 to 100
- Resolution set to *fixed*

#9 - 11/05/2010 04:14 PM - Johannes Zarl-Zierl

For the record: finally fixed in Revision r2166