

## Robotics Service Bus - Feature #1767

### Export default plugin folder

02/13/2014 05:38 PM - Anonymous

<b>Status:</b>	Feedback	<b>Start date:</b>	02/13/2014
<b>Priority:</b>	Low	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	50%
<b>Category:</b>	Configuration	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	rsb-0.18		
<b>Description</b>			
<p>There is some kind of <i>default plugin folder</i> (currently lib/rsbx.x/plugins), where several downstream projects install their plugins (e.g. rst converters, rsb-spread, ...) for convenience. Up to now the downstream projects construct this path by hand.</p> <p>I suggest that we export e.g. a RSB_DEFAULT_PLUGIN_PATH via cmake config for downstream projects to use. Does that make sense?</p>			
<b>Related issues:</b>			
Related to Robotics Service Bus - Tasks # 1227: Add a CMake macro for RSB plu...		<b>Feedback</b>	<b>11/05/2012</b>

#### Associated revisions

##### Revision 18242a2b - 02/17/2014 01:48 PM - Arne Nordmann

Exports the default RSB\_SYSTEM\_PLUGIN\_DIRECTORY via cmake

- May be as install target for rsb plugins by downstream projects

fixes #1767

##### Revision 5be5468a - 02/17/2014 01:54 PM - Arne Nordmann

Exports the default RSB\_SYSTEM\_PLUGIN\_DIRECTORY via cmake

- May be as install target for rsb plugins by downstream projects

fixes #1767

##### Revision 4f92325a - 02/17/2014 02:16 PM - Arne Nordmann

Exports the default RSB\_SYSTEM\_PLUGIN\_DIRECTORY via cmake

- May be used as install target for rsb plugins by downstream projects

fixes #1767

##### Revision daf5fdb5 - 02/20/2014 03:15 PM - J. Moringen

Mention new CMake variable RSB\_SYSTEM\_PLUGIN\_DIRECTORY in news.rst

refs #1767

- news.rst (RSB 0.11): mention new CMake variable  
RSB\_SYSTEM\_PLUGIN\_DIRECTORY

#### Revision 34c77416 - 03/22/2014 07:20 PM - J. Moringen

Added RSB\_RELATIVE\_PLUGIN\_DIRECTORY in CMakeLists.txt

refs #1767

- CMakeLists.txt: added RSB\_RELATIVE\_PLUGIN\_DIRECTORY
- RSBConfig.cmake.in: use it

## History

---

### #1 - 02/13/2014 06:23 PM - Anonymous

- *Description updated*

### #2 - 02/13/2014 06:24 PM - J. Moringen

| *There seems to be some kind of default plugin folder (currently lib/rsbx.x/plugins)*

Yes, RSB configures the RSC plugin loader with a default search path (a list of directories) which contains a prefix-dependent "system plugin directory" (as opposed to user plugin directory in the respective home directory).

| *Up to now the downstream projects construct this path by hand.*

That is indeed suboptimal.

| *I suggest that we export e.g. a RSB\_DEFAULT\_PLUGIN\_PATH via cmake config for downstream projects to use. Does that make sense?*

I think the proposal itself makes perfect sense. I would suggest a different name, though.

Firstly, \_DIRECTORY may be better than \_PATH because the value of the variable would be a directory, not a list of these.

Secondly, instead of DEFAULT, I suggest SYSTEM to distinguish the directory from the user-specific plugin directory (This would fit into the usual nomenclature of "user", "system", "site", where "site" would be appropriate for organization-wide but not system-wide things).

As a result, I suggest the name RSB\_SYSTEM\_PLUGIN\_DIRECTORY.

### #3 - 02/17/2014 01:18 PM - Anonymous

- *Status changed from Feedback to In Progress*
- *Assignee set to Anonymous*

### #4 - 02/17/2014 01:51 PM - Anonymous

- *Status changed from In Progress to Resolved*
- *% Done changed from 0 to 100*

Applied in changeset rsb-cpp|commit:18242a2b50f8ca44729ebb39e71629a46f4b92b9.

**#5 - 03/20/2014 06:32 PM - Anonymous**

- % Done changed from 100 to 50

I am re-opening this issue, since we have to put a bit more thoughts into this. With the latest changes we export the plugin folder as *absolute* path (based on the install prefix). However, this can lead into problems, as Christian discovered:

1. Downstream projects can't use this path, e.g. when building packages with cpack, as pathes have to be relative for this.
2. As an organisational issue: When we build rsb in the toolkit context, the exported plugin directory will point to the toolkit, so that downstream user libraries will try to install their plugins into a toolkit folder.

It seems, that we have to at least export (also?) a relative version of the RSB\_SYSTEM\_PLUGIN\_DIRECTORY and maybe also some kind of RSB\_USER\_PLUGIN\_DIRECTORY (e.g. ~/.rsb/plugins/), which is in the user's home folder and also added to the default plugin path of rsb.

**#6 - 03/20/2014 07:02 PM - Anonymous**

- Status changed from Resolved to Feedback

It might be enough to just export the RSB\_PLUGIN\_DIRECTORY as relative path:

1. Downstream projects would install their plugins in the relative path, so into: downstream install prefix + RSB\_PLUGIN\_DIRECTORY. This works for installation and cpack package build (it might be necessary to enable CPACK\_SET\_DESTDIR, cf. source:rsb-cpp|cpack/CPackDebianConfig.cmake#L17, though).
2. In our toolkit context, downstream libraries *included* in the toolkit would install plugins in the toolkit plugin folder (as intended). Downstream *user* libraries will install their plugins into a local plugin folder (downstream install prefix + RSB\_PLUGIN\_DIRECTORY), which has to be added to the plugin path configuration.

**#7 - 12/17/2014 11:09 AM - J. Wienke**

Arne, any way to solve this for the release?

**#8 - 01/07/2015 01:46 PM - J. Moringen**

- Target version changed from rsb-0.11 to rsb-0.12

**#9 - 04/27/2015 11:10 AM - J. Wienke**

- Target version changed from rsb-0.12 to rsb-0.13

**#10 - 03/02/2016 08:55 AM - J. Moringen**

- Target version changed from rsb-0.13 to rsb-0.14

**#11 - 06/08/2016 09:01 PM - J. Moringen**

- Target version changed from rsb-0.14 to rsb-0.15

**#12 - 11/07/2016 10:33 AM - J. Moringen**

- Target version changed from rsb-0.15 to rsb-0.16

**#13 - 04/10/2017 08:59 PM - J. Moringen**

- Target version changed from rsb-0.16 to rsb-0.17

**#14 - 10/16/2017 09:23 PM - J. Moringen**

- Target version changed from rsb-0.17 to rsb-0.18