

## Robotics Service Bus - Support #2040

### rsb-sendcl doesn't allow to send raw strings / missing or poor documentation?

10/06/2014 03:05 PM - R. Haschke

<b>Status:</b>	Resolved	<b>Start date:</b>	10/06/2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	J. Moringen	<b>% Done:</b>	100%
<b>Category:</b>	Common Lisp Tools	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	rsb-0.11		
<b>Description</b>			
all the following methods to send a string fail:  rsb-sendcl "foo" / cat file.txt   rsb-sendcl - /  What is the syntax to send more complex types, e.g. an rst-type?			
<b>Related issues:</b>			
Related to Robotics Service Bus - Enhancement # 964: Reduce redundancy betwee...		<b>Resolved</b>	<b>03/19/2012</b>

#### Associated revisions

##### Revision 5629fbb6 - 10/07/2014 03:25 AM - J. Moringen

Improved explanation for specifying send payloads in send.rst

refs #2040

- send.rst (Description): mention empty payload and reading payload from file; explain interaction with shell syntax  
(Examples): added examples for empty payloads, integer payloads and payloads from standard input and files

##### Revision af8113ed - 10/07/2014 04:20 AM - J. Moringen

Improved explanation for specifying arguments in call.rst

refs #2040

- call.rst (Description): mention reading payload from file; explain interaction with shell syntax  
(Examples): added examples for payloads from standard input and files

##### Revision 95fcb610 - 10/07/2014 05:24 AM - J. Moringen

Improved help strings in {send,call}/main.lisp

refs #2040

- send/main.lisp (make-help-string): mention reading payload from file;

explain interaction with shell syntax

(make-examples-string): added example for empty payload; explain interaction with shell syntax

- call/main.lisp (make-help-string): similar  
(make-examples-string): similar

### Revision 20f00641 - 10/07/2014 05:36 AM - J. Moringen

Added functions for event construction in common/event.lisp

fixes #964, refs #2040

- common/event.lisp: new file; contains functions for constructing events by parsing payload and meta-data specifications
- common/package.lisp (package rsb.common): added exported symbols parse-payload-spec, parse-meta-data, parse-timestamp and parse-cause
- call/main.lisp (make-help-string): adapted to extended payload parsing rules  
(make-examples-string): likewise  
(parse-argument): removed; 'parse-payload-spec' is used instead  
(main): call 'parse-payload-spec' instead of 'parse-argument'
- send/main.lisp (make-help-string): adapted to extended payload parsing rules  
(make-examples-string): likewise  
(parse-event-spec): removed; 'parse-payload-spec' is used instead  
(parse-pair): removed; moved to common/event.lisp  
(parse-meta-data): likewise  
(parse-timestamp): likewise  
(parse-cause): likewise  
(main): call 'parse-payload-spec' instead of 'parse-event-spec'
- cl-rsb-common.asd (header): updated copyright  
(system cl-rsb-common): added file
- CMakeLists.txt: added test cases for executing send and call binaries with different kinds of payload specifications

### Revision 3b14041c - 10/07/2014 05:43 AM - J. Moringen

Describe extended file and stdin syntax in {call,send}.rst

refs #964, refs #2040

- call.rst (Description): explain extended file and standard input syntax  
(Examples): added examples of the above
- send.rst (Description): similar  
(Examples): similar

## History

---

#1 - 10/06/2014 03:07 PM - J. Wienke

- Description updated

**#2 - 10/06/2014 03:12 PM - J. Wienke**

- Category set to *Common Lisp Tools*
- Target version set to *rsb-0.11*

The online documentation states the following:

*EVENT-SPEC is parsed as string when surrounded with double-quotes and as integer or float number when consisting of digits without and with decimal point respectively.*

So for me at least the following works as expected:

```
rsb-sendcl "test string" spread:/foo
```

However, the input from stdin seems to be broken:

```
jwienke@cinnabar:~$ cat /etc/passwd | rsb-sendcl - spread:/foo
```

The bounding indices 2 and NIL are bad for a sequence of length 1.

See also:

The ANSI Standard, Glossary entry for "bounding index designator"

The ANSI Standard, writeup for Issue SUBSEQ-OUT-OF-BOUNDS:IS-AN-ERROR

**#3 - 10/06/2014 03:28 PM - J. Moringen**

- Status changed from *New* to *In Progress*
- Assignee set to *J. Moringen*
- % Done changed from *0* to *40*

*However, the input from stdin seems to be broken:  
[...]*

I reported this as an upstream bug. Currently waiting for upstream.

I will also try to improve the documentation.

**#4 - 10/06/2014 03:28 PM - J. Moringen**

*What is the syntax to send more complex types, e.g. an *rst*-type?*

Not supported yet, but in development.

**#5 - 10/06/2014 04:56 PM - R. Haschke**

I wasn't expecting the double quotes. Please document this.

**#6 - 10/07/2014 03:45 AM - J. Moringen**

- % Done changed from 40 to 80

Mainly waiting for upstream fix.

**#7 - 10/07/2014 04:17 AM - J. Moringen**

- Related to Enhancement #964: Reduce redundancy between call and send tools added

**#8 - 10/07/2014 05:54 AM - J. Moringen**

- Status changed from In Progress to Resolved

- % Done changed from 80 to 100