Robotics Service Bus - Bug #2637
Current CL tools do not handle unavailability of rpath

08/01/2016 10:49 AM - J. Wienke

Status:	Rejected	**Start date:**	08/01/2016
Priority:	Normal	**Due date:**	
Assignee:	J. Moringen	**% Done:**	0%
Category:	Common Lisp Tools	**Estimated time:**	0.00 hour
Target version:	rsb-0.15		

Description

I have sshfs mounted /vol/toolkit to a different location, so that the original path /vol/tookit isn't available on the computer. The tools fail to start then:

```
jwienke@perftest:~/toolkit$ export LD_LIBRARY_PATH=/homes/jwienke/toolkit/nightly/trusty/x86_64/2016-08-01_01-50-05/lib
jwienke@perftest:~/toolkit$ ./nightly/trusty/x86_64/2016-08-01_01-50-05/bin/rsb-toolscl0.15
Unhandled SIMPLE-ERROR in thread #<SB-THREAD:THREAD "main thread" RUNNING
                                  {1008AB4EA3}>:
  Error opening shared object "/vol/toolkit/nightly/trusty/x86_64/2016-08-01_01-50-05/lib/libspread.so":
  /vol/toolkit/nightly/trusty/x86_64/2016-08-01_01-50-05/lib/libspread.so: cannot open shared object file: No such file or
directory.

Backtrace for: #<SB-THREAD:THREAD "main thread" RUNNING {1008AB4EA3}>
0: ((LAMBDA NIL :IN SB-DEBUG::FUNCALL-WITH-DEBUG-IO-SYNTAX))
1: (SB-IMPL::CALL-WITH-SANE-IO-SYNTAX #<CLOSURE (LAMBDA NIL :IN
SB-DEBUG::FUNCALL-WITH-DEBUG-IO-SYNTAX) {1008ABE70B}>)
2: (SB-IMPL::%WITH-STANDARD-IO-SYNTAX #<CLOSURE (LAMBDA NIL :IN
SB-DEBUG::FUNCALL-WITH-DEBUG-IO-SYNTAX) {1008ABE6DB}>)
3: (PRINT-BACKTRACE :STREAM #<SYNONYM-STREAM :SYMBOL SB-SYS:*STDERR* {10001CA823}> :START 0 :FROM
:INTERRUPTED-FRAME :COUNT NIL :PRINT-THREAD T :PRINT-FRAME-SOURCE NIL :METHOD-FRAME-STYLE NIL)
4: (SB-DEBUG::DEBUGGER-DISABLED-HOOK #<SIMPLE-ERROR "Error opening ~:[runtime~;shared object ~:*~S~]:~%
~A." {1008ABB7E3}> #<unavailable argument>)
5: (SB-DEBUG::RUN-HOOK *INVOKE-DEBUGGER-HOOK* #<SIMPLE-ERROR "Error opening ~:[runtime~;shared object
~:*~S~]:~%  ~A." {1008ABB7E3}>)
6: (INVOKE-DEBUGGER #<SIMPLE-ERROR "Error opening ~:[runtime~;shared object ~:*~S~]:~%  ~A." {1008ABB7E3}>)
7: (ERROR "Error opening ~:[runtime~;shared object ~:*~S~]:~%  ~A."
"/vol/toolkit/nightly/trusty/x86_64/2016-08-01_01-50-05/lib/libspread.so"
"/vol/toolkit/nightly/trusty/x86_64/2016-08-01_01-50-05/lib/libspread.so: cannot open shared object file: No such file or
directory")
8: (SB-SYS:DLOPEN-OR-LOSE #S(SB-ALIEN::SHARED-OBJECT :PATHNAME
#P"/vol/toolkit/nightly/trusty/x86_64/2016-08-01_01-50-05/lib/libspread.so" :NAMESTRING
"/vol/toolkit/nightly/trusty/x86_64/2016-08-01_01-50-05/lib/libspread.so" :HANDLE NIL :DONT-SAVE NIL))
9: (SB-ALIEN::TRY-REOPEN-SHARED-OBJECT #S(SB-ALIEN::SHARED-OBJECT :PATHNAME
#P"/vol/toolkit/nightly/trusty/x86_64/2016-08-01_01-50-05/lib/libspread.so" :NAMESTRING
"/vol/toolkit/nightly/trusty/x86_64/2016-08-01_01-50-05/lib/libspread.so" :HANDLE NIL :DONT-SAVE NIL))
10: (SB-SYS:REOPEN-SHARED-OBJECTS)
11: (SB-IMPL::FOREIGN-REINIT)
12: (SB-IMPL::REINIT)
13: ((FLET #:WITHOUT-INTERRUPTS-BODY-80 :IN SAVE-LISP-AND-DIE))
14: ((LABELS SB-IMPL::RESTART-LISP :IN SAVE-LISP-AND-DIE))

unhandled condition in --disable-debugger mode, quitting
```

I think this should only be a warning for being relocatable

History

#1 - 08/01/2016 10:49 AM - J. Wienke

- Description updated

#2 - 08/01/2016 01:02 PM - J. Moringen

- Status changed from New to Feedback

J. Wienke wrote:

> *I think this should only be a warning for being relocatable*

What you describe is actually the default behavior. The toolkit binaries are the result of something like rsb-tools redump rsb-tools static.

This can be reversed via rsb-tools redump /tmp/rsb-tools-dynamic. Note that this has to be done while the library is still available under the filename mentioned above.

While at it, maybe do rsb-tools redump /tmp/rsb-tools-dynamic compress which will take longer but result in a binary that is about 25 % of the original size.

#3 - 08/01/2016 01:09 PM - J. Wienke

Ok, what does the redump thing do? Some optimization?

#4 - 08/01/2016 01:19 PM - J. Moringen

J. Wienke wrote:

> *Ok, what does the redump thing do? Some optimization?*

It creates a new binary that behaves like the source binary except for loading of shared libraries. It can also create a compressed binary from an uncompressed one or the other way around.

So, optimization-wise, it is a storage vs. startup time trade-off. Also, running multiple instances of a compressed binary takes more memory since they are uncompressed in memory individually while uncompressed binaries can just be mmap(2)ed and share those pages.

#5 - 08/01/2016 01:35 PM - J. Wienke

- Status changed from Feedback to Rejected

Ok, anyway, if this is a desired behavior and alternatives exist I am fine.