

# How To: RTPC Debugging

---



## Question:

---

### Symptoms:

Simulation stops and model gets automatically disconnected from the Target Server with a "SIGSEGV" error message in EE Application Log Window.



## Answer:

---

### Reason:

After inspection of the RTPC log files a "SIGSEGV" (segment violation) error during simulation was seen.

That means there is an illegal pointer access within source code.

To investigate which is the cause of the error, the RTPC offers some kind of debugging possibilities which can be used to identify and repair (if possible) the root cause of a model crash.

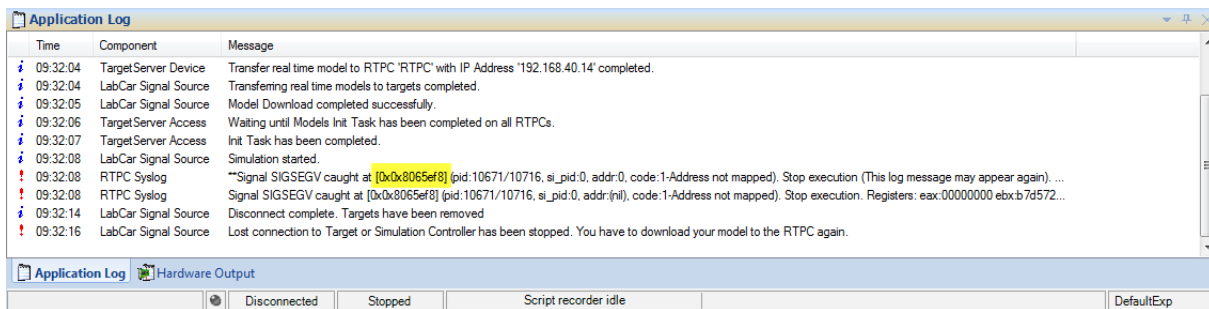
Please follow the next guidelines:

- 1.) Stop the Simulation Controller.
- 2.) Select Configure LabCar-RTPC from the Main Page of the LABCAR-RTPC-Web-Interface
- 3.) Go to the LabCar-RTPC Configuration and choose the following settings for debugging purposes:
  - RTPC\_LOG\_LEVEL = debug
  - RTPC\_COMPILE\_OPTIMIZATION = 0
  - RTPC\_COMPILE\_LINK\_DEBUG = yes

Press Button Save LABCAR-RTPC Configuration:

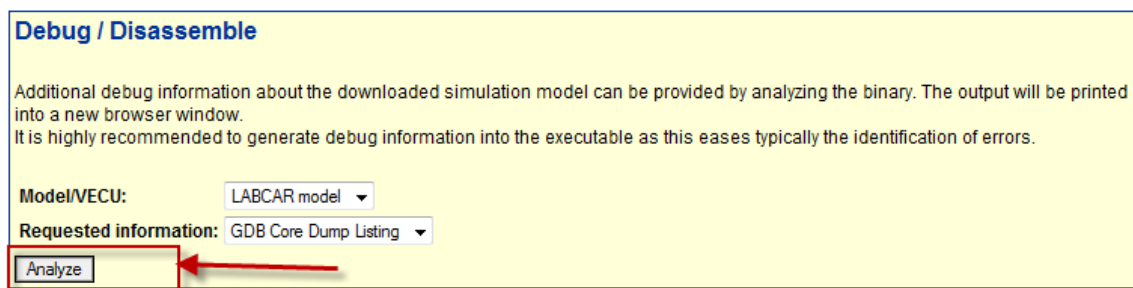
Size of the trace buffer (in traceable events). <a href="#">Help</a>	
<b>RTPC_COMPILE_OPTIMIZATION</b>	0 ▾
Optimization level of the compiler. <a href="#">Help</a>	
<b>RTPC_COMPILE_LINK_DEBUG</b>	yes ▾
Place debugging information in generated code. <a href="#">Help</a>	

- 4.) Build the whole LabCar-Project again via Build LCO Project.
- 5.) Start the Simulation Controller from the Main Page of the LABCAR-RTPC- Web-Interface.
- 6.) Open and Start the newly built Experiment.
- 7.) Wait until the SIGSEGV error occurs and note the address after caught at in the Application Log Info:



8.) Select System Info from the Main Page of the LABCAR-RTPC-Web-Interface

9.) Go to the Debug / Disassemble section and Press Button Analyze:



10.) Search for the noted SIGSEGV error address to debug your project, cf. 7., for my example **0x08065ef8**:

```

disassemble /m $pc
Dump of assembler code for function cmod_Process_Controller_Controller:
37      {
0x08065e1c <+0>:      push   %ebp
0x08065e1d <+1>:      mov    %esp,%ebp
0x08065e1f <+3>:      sub   $0x28,%esp

38          // Get Imports
39          #include "controllercopyinports.h"
40
41          // Enter your code here:
42          // Make the Pointer to bring to a SIGSEV during Simulation
43          static int *pointer = NULL;
44          static int count;
45          if (count++ > 200)
0x08065eda <+190>:    mov    0x80dd4ec,%eax
0x08065edf <+195>:    cmp   $0xc8,%eax
0x08065ee4 <+200>:    setg  %dl
0x08065ee7 <+203>:    add  $0x1,%eax
0x08065eea <+206>:    mov  %eax,0x80dd4ec
0x08065eef <+211>:    test  %dl,%dl
0x08065ef1 <+213>:    je   0x8065efe <cmod_Process_Controller_Controller+226>

46          {
47              *pointer = 10; // Hier kracht es.
=> 0x08065ef8 <+220>:    movl  $0xa,(%eax)

48          }

49
50
51          -----
52          //Werte einlesen:
53          -----
54          internal_Current_Floor=Inport_Current_Floor;
0x08065efe <+226>:    fldl  0x80ddd4f8
0x08065f04 <+232>:    fisttpl -0xc(%ebp)
0x08065f07 <+235>:    mov  -0xc(%ebp),%eax
0x08065f0a <+238>:    mov  %eax,0x80dd4d4

```

With the correct optimization settings you should be able to see in which module at which line of the code the model has crashed.

The generated report at the RTPC with the debugging results (usually called by default **ETAS RTPC.mht**) contains the information about the code.

Please do not forget to undo the compile settings mentioned above, after your analysis is complete.



### **Additional information:**

---

This procedure was tested with LABCAR-OPERATOR 5.3.0 and RTPC 6.1.0.



### **In case of further questions:**

---

You will find further FAQs here: [www.etas.com/en/faq](http://www.etas.com/en/faq)

Please feel free to contact our Support Center, if you have further questions.

Here you can find all information: <http://www.etas.com/en/hotlines.php>

331549090 / 4 / 14 12 15      400927726 / 12

This information (here referred to as „FAQ“) are provided without any (express or implied) warranty, guarantee or commitment regarding completeness or accuracy. Except in cases of willful damage, ETAS shall not be liable for losses and damages which may occur or result from the use of this information (including indirect, special or consequential damages).