

# Is it possible to write extensions to Delphi's debugger?

Asked 15 years, 11 months ago   Modified 15 years, 3 months ago

Viewed 864 times



40



I know there's an API for creating extensions to Delphi. I use the GExperts package and various JVCL experts frequently. But I've never seen any extensions to the debugger. It would be very nice, for example, to be able to register viewers for various objects instead of having to examine them in the Inspector. (A form with an image control that displays a TImage, for example, or a grid that displays the contents of a dataset.)

Are there any APIs that allow you to extend Delphi's debugger in this way?

EDIT: This wasn't available back when I wrote the question, but Delphi 2010 provides a way to do it.

delphi

debugging

Share

edited Sep 2, 2009 at 15:13

Improve this question

Follow

asked Jan 16, 2009 at 3:26



Mason Wheeler

84.5k ● 54 ● 284 ● 502

---

Good question :-) 10 upvotes and one answer.

– [Wouter van Nifterick](#) Jan 17, 2009 at 15:12

- 
- 1 I agree. Maybe Nick Hodges will notice - I think he does pay attention to the community. – [Argalatyr](#) Jan 18, 2009 at 16:46

---

Please see QC 1138: [qc.codegear.com/wc/qcmain.aspx?d=1138](http://qc.codegear.com/wc/qcmain.aspx?d=1138) – [Rob Kennedy](#) Jan 19, 2009 at 18:12

- 
- 1 @Rob Kennedy: this QC 1138 entry is closed! – [Francesca](#) Feb 27, 2009 at 19:56

---

Good question, one more upvote. – [Fabricio Araujo](#) May 26, 2009 at 19:23

---

## 2 Answers

Sorted by:

Highest score (default)



17



In ToolsAPI.pas source there is some API interfaces for debugging. You can be informed when a debugging event occurred, info about breakpoints, which process is being debugged etc. with this API. But it seems there is no support for variables or values of them. So there is no easy way to implement your requisition without ugly hacks.

Share Improve this answer

answered Jan 17, 2009 at 14:37

Follow



Khan

186 ● 2 ● 2



4

Basic debugger visualizers can be implemented with the Evaluation interfaces exposed by the OTA. (Examples for debugger visualizers can be found [here](#) and [here](#).)



A deeper integration into the debugger is possible as well (for example, I wrote a little extension for C++Builder that enables the debugger to evaluate the actual objects behind an interface) - but as Khan pointed out, to achieve such a level of integration, you'll need to resort to quite a few dirty hacks.

Share Improve this answer

answered May 25, 2009 at 21:36

Follow



Moritz Beutel

1,941 ● 2 ● 15 ● 18

---