How do I use PDB files

Asked 16 years, 3 months ago Modified 8 years, 1 month ago

Viewed 48k times

51

I have heard using PDB files can help diagnose where a crash occurred.
My *basic* understanding is that you give Visual studio the source file, the pdb file and the crash information (from Dr Watson?)
Can someone please explain how it all works / what is involved? (Thank you!)

debugging **crash** **pdb-files**

Share

Improve this question

Follow

edited Apr 15, 2010 at 18:12

SamB

9,205 ● 5 ● 50 ● 57

asked Sep 16, 2008 at 13:22

hamishmcn

7,981 ● 11 ● 42 ● 46

Thanks for the answers so far (I will leave the question open a bit longer) I have also found this useful: codeproject.com/KB/debug/postmortemdebug_standalone1.aspx – hamishmcn Sep 16, 2008 at 16:35

3 Answers

Sorted by: Highest score (default)

44

PDB files map an assembly's MSIL to the original source lines. This means that if you put the PDB that was compiled with the assembly in the same directory as the assembly, your exception stack traces will have the names and lines of the positions in the original source files. Without the PDB file, you will only see the name of the class and method for each level of the stack trace.

Share Improve this answer

Follow

answered Sep 16, 2008 at 13:25

Omer van Kloeten

12k ● 9 ● 44 ● 54

3 +1: For the good description of what a "program database" (PDB) contains. – AMissico Jun 29, 2010 at 4:53

38

PDB files are generated when you build your project. They contain information relating to the built binaries which Visual Studio can interpret.

When a program crashes and it generates a crash report, Visual Studio is able to take that report and link it back to the source code via the PDB file for the application. PDB files must be built from the same binary that generated the crash report!

There are some issues that we have encountered over time.

- The machine that is debugging the crash report needs to have the source on the same path as the machine that built the binary.
- Release builds often optimize to the extent where you cannot view the state of object member variables

If anyone knows how to defeat the former, I would be grateful for some input.

Share Improve this answer

Follow

edited May 23, 2011 at 17:43

Gustavo Mori

8,386 ● 3 ● 39 ● 52

answered Sep 16, 2008 at 13:31

roo

7,196 ● 9 ● 42 ● 45

5 One way to help with the source file path: use SUBST to map a drive letter to a particular directory. technet.microsoft.com/en-us/library/bb491006.aspx – MarkJ Jun 9, 2010 at 15:56

Most debuggers that I have seen also allow you to enter arbitrary path replacements - but possibly that is not available in Visual Studio. Never tried, but it is one of these things you generally need to do to debug a program compiled by somebody else. – jakobengblom2 Dec 5, 2012 at 9:09

12

You should look into setting up a symbol server and indexing the PDB files to your source code control system. I just recently went through this process for our product and it works very well. You don't have to be

concerned about making PDB files available with the binaries, nor how to get the appropriate source code when debugging dump files.

John Robbins' book: http://www.amazon.com/Debugging-Microsoft-NET-2-0-Applications/dp/0735622027/ref=pd_bbs_sr_1?ie=UTF8&s=books&qid=1222366012&sr=8-1

Look here for some sample code for generating minidumps (which don't have to be restricted to post-crash analysis -- you can generate them at any point in your code without crashing): http://www.codeproject.com/KB/debug/postmortemdebug_standalone1.aspx

Share Improve this answer

Follow

edited Jun 11, 2014 at 14:06

infojolt

5,402 ● 3 ● 48 ● 84

answered Sep 25, 2008 at 18:06

lesscode

6,341 ● 1 ● 32 ● 59

Great link on how to setup a source symbol server! I might just do that. +1 for the link. – GurdeepS Feb 20, 2011 at 2:15