

# Troubleshooting application crashes

If the application you are profiling stops responding, fails to load, or closes unexpectedly, the profiling process may have caused the profiled application to crash. If this happens, you may not receive profiling results for your session, or you may receive incomplete results captured before the crash.

If your profiling session returns no results, or incomplete results, but does not hang or crash, see [Troubleshooting missing results](#).

## Troubleshooting all crashes

If the application crashed, try the following approaches to fix the problem. They are listed in order of how likely they are to resolve common causes of crashes.

### Force your application to use .NET 4

Many crashes can be avoided by rebuilding the application as a .NET 4 (including .NET 4.5) executable, and profiling it using the **Attach to a .NET 4 process** option on the settings screen.

For instructions on reconfiguring your application for use with this profiling method, see [Forcing your application to use .NET 4](#).

The **Attach to a .NET 4 process** approach profiles your application when it is already running, rather than launching a new instance of the application. Because attaching to a running process does not change the way the application's code executes, it is less likely to cause a crash.

 You can't attach to a .NET 4 process with Windows Store apps.

## Run ANTS Performance Profiler from the Visual Studio menu

If the crash occurs in a profiling session launched directly from ANTS Performance Profiler, try profiling using the Visual Studio ANTS Performance Profiler add-in instead.

For instructions see [Using the Visual Studio add-in](#).

### Delete corrupt third-party PDBs

The following third-party assemblies sometimes contain corrupt *.pdb* files, which can cause crashes during profiling:

- Microsoft.Practices library
- AjaxControlToolkit

For instructions on finding and deleting corrupt *.pdb* files, see [Troubleshooting PDB problems](#).

## Troubleshooting web application crashes

If the application you are profiling is a web application, try the following two steps:

### Profile using another web server application

Some applications that crash under profiling can be profiled successfully if run on another type of server application:

- If you are profiling in IIS, on the settings screen, from the list of application types, switch to **Web development server - ASP.NET** or **IIS Express - ASP.NET**.
- If you are profiling in the ASP.NET web development server, on the settings screen, from the list of application types, switch to **IIS - ASP.NET** or **IIS Express - ASP.NET**.

 Changing environments will work only if the server application you select supports all your application's features:

- Applications using integrated pipeline mode, impersonation, or HTTPS will not run on the built-in web-development server
- Applications that cannot operate in a restricted security context may not run in IIS

Profiling results obtained in from applications running on the web development server may also differ from results obtained in a production environment.

### IIS: ensure you are profiling on the correct port

- If you are using **IIS 6** or **IIS 7**, on the settings screen, select **Unused port** and choose a port that is not used by IIS.



This will not work if your application's code binds to a specific port.

- If you are using **IIS 5**, or if you are using **IIS 6 or 7** and your application binds to a specific port, on the settings screen, select **Original port**. IIS will restart so that the profiler can attach to the port.

## Troubleshooting error messages

### StackOverflow Exception

During line-level profiling, if the application you are profiling is very large or the profiling session lasts a long time, a stack overflow can occur. If this happens, the application will crash and may record a "StackOverflow Exception" error in the log. (For instructions on finding the ANTS Performance Profiler logs, see [Log files](#).)

To reduce the risk of an overflow, on the **Tools** menu, click **Advanced Options**, and select **Simplify very complex stack traces to save memory**. **THIS MAY CHANGE!**

If this does not prevent the crash, try profiling in sampling mode. On the settings screen, set the **Profiling mode** to **Sample method-level timings**.

### Getting more help

If the steps above don't prevent the profiled application from crashing, please [contact Redgate support](#), including, if possible, a log of the application crash. See [Log files](#) for more information about viewing profiler logs.