

2012

ALM Rangers Hands-on Lab

IntelliTrace - No Symbols Found Resolution (Client side)

April 09, 2013

This content was created by the Visual Studio ALM Rangers, a special group with members from the Visual Studio Product Team, Microsoft Services, Microsoft Most Valuable Professionals (MVPs) and Visual Studio Community Leads.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Microsoft grants you a license to this document under the terms of the Creative Commons Attribution 3.0 License**.** All other rights are reserved.

* 2012 Microsoft Corporation.*

Microsoft, Active Directory, Excel, Internet Explorer, SQL Server, Visual Studio, and Windows are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Table of Contents

[Introduction 4](#_Toc353274712)

[Visual Studio ALM Rangers 4](#_Toc353274713)

[Contributors 4](#_Toc353274714)

[Reviewers 4](#_Toc353274715)

[Prerequisites 4](#_Toc353274716)

[Exercise 1 – Use MTM to generate an IntelliTrace log 5](#_Toc353274717)

[Exercise 2 – Run IntelliTrace session with missing symbols 8](#_Toc353274718)

[Exercise 3 – Configure Visual Studio to locate the symbols 10](#_Toc353274719)

[Exercise 4 – Run IntelliTrace again and have Visual Studio locate the symbols 12](#_Toc353274720)

[Appendix 13](#_Toc353274721)

[Other ALM Rangers Resources 13](#_Toc353274722)

[Code Reference 13](#_Toc353274723)

[Figures Reference 13](#_Toc353274724)

[Tables Reference 13](#_Toc353274725)

# Introduction

This hands-on lab will show you how to configure Visual Studio to locate missing symbols for an IntelliTrace log file.

The lab will take approximately 20 minutes to complete.

The hands-on lab is part of a series of labs that will give you a complete end-to-end scenario of using IntelliTrace in production. The series is structured as follows:

1. Learn how to configure TFS build to generate source and symbols information.
2. Learn now to configure the stand-alone IntelliTrace collector.
3. Learn how to setup Visual Studio to locate symbols when analyzing an IntelliTrace log (this lab).

## Visual Studio ALM Rangers

The Visual Studio ALM Rangers are a special group with members from the Visual Studio Product group, Microsoft Services, Microsoft Most Valuable Professionals (MVP) and Visual Studio Community Leads. Their mission is to provide out-of-band solutions to missing features and guidance. A growing Rangers Index is available online[[1]](#footnote-2).

## Contributors

Mathias Olausson

## Reviewers

Jesse Houwing

Anna Galaeva

Giulio Vian

Larry Guger

## Prerequisites

To complete these hands-on-lab walk-through scenarios you need the following environment:

* The “Brian Keller TFS 2012 Demo Virtual Machine”[[2]](#footnote-3).

# Exercise 1 – Use MTM to generate an IntelliTrace log

|  |  |
| --- | --- |
| Objective | In this exercise will you will use Microsoft Test Manager, MTM, to test the Tailspin Toys administration client. You will configure MTM to collect IntelliTrace information as part of the test session, attach it to a bug report, which later will be used in Visual Studio to troubleshoot the problem. |

1. Start the virtual machine.
2. Log in to the Virtual Machine as **Adam Barr** using the standard password **P2ssw0rd**.
3. Start the Microsoft Test Manager and connect to the Tailspin Toys team project.

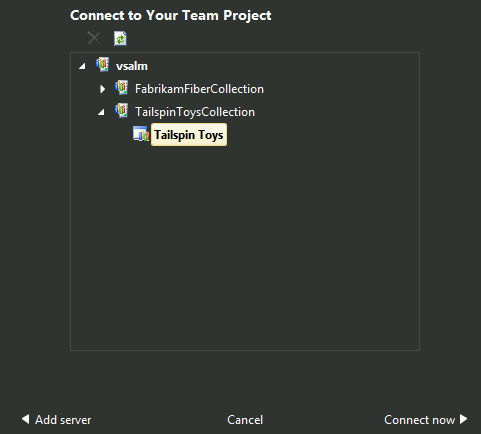


Figure - Connect MTM to the Tailspin Toys team project.

1. Select the “Iteration 2” test plan and press “Select plan”.
2. Navigate to the Test activity in MTM. Start an exploratory test session from the “Do Exploratory Testing” menu. Press the Explore with options button to start testing.

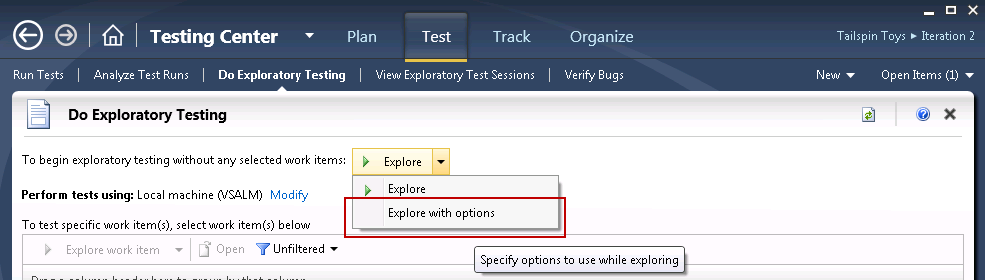


Figure - Start exploratory testing.

1. Select the “Full Diagnostics” test settings to configure the test plan to collection IntelliTrace data. Press Explore to start testing.

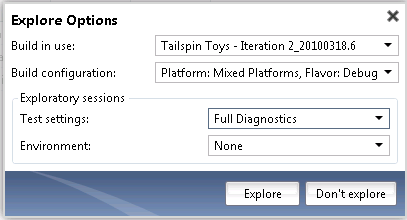


Figure - Explore with full diagnostics.

1. Press the Start button in the exploratory test runner to start testing.
2. Start the Tailspin Toys administration client from the latest build folder.
   1. Open a Windows Explorer and go to the “C:\drops\Tailspin Toys - Iteration 2\Tailspin Toys - Iteration 2\_20100318.6” folder.
   2. Start **Tailspin.Admin.App.exe**.
3. Select the product grid and use the arrows keys to navigate to the end of the list. This will cause an exception to occur on the application.

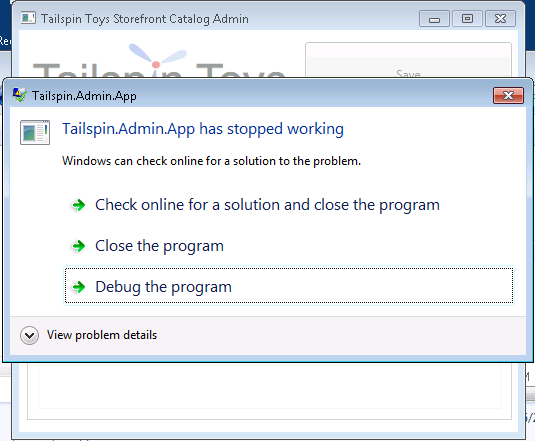


Figure - Application under test failed with exception.

1. Select “Close the program” to end the Tailspin Toys admin application.
2. File a bug using the Create Bug button in the MTM toolbar.

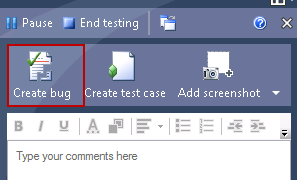


Figure - Create new bug in MTM.

1. Name the bug “Admin client crashes when browsing products” and assign it to Adam Barr. Make sure an iTrace file is added to the bug.

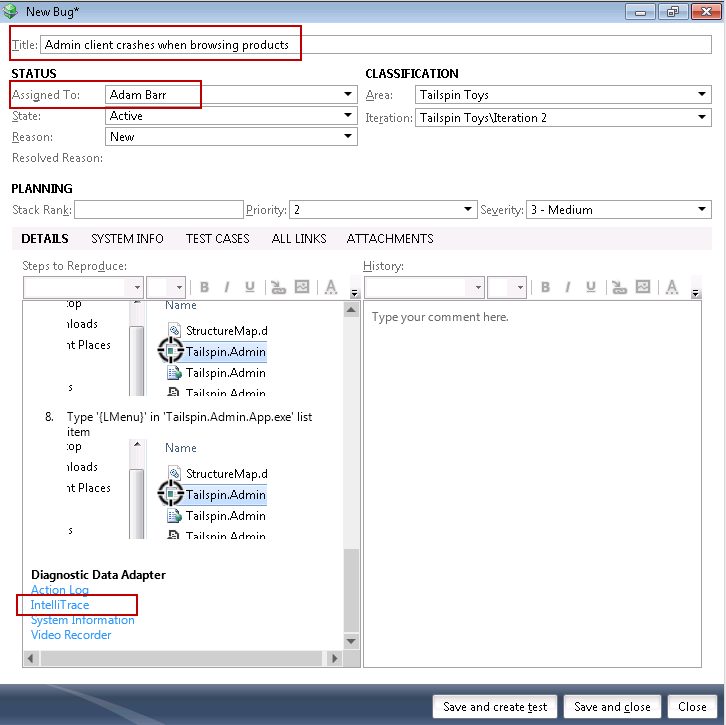


Figure - Fill out the bug report and make sure iTrace file is attached.

1. Save and close the bug report.
2. End the exploratory test session.

|  |  |
| --- | --- |
| Review | This exercise you have configured and run a test session what will collect IntelliTrace data. You have learnt how to create a new bug and have MTM automatically attach the collected data to the bug report. |

# Exercise 2 – Run IntelliTrace session with missing symbols

|  |  |
| --- | --- |
| Objective | This exercise will show you how to debug an IntelliTrace log file sent to you from a remote environment. But without having matching source symbol files available IntelliTrace will not be able to open the matching source code location. |

1. Start Visual Studio, open the Team Explorer and connect to the Tailspin Toys team project.
2. Navigate to the Work Items view in Team Explorer and run the My Bugs.
3. Open the bug created in exercise 1, go to the All Links tab and double-click the .iTrace file to open the IntelliTrace log.

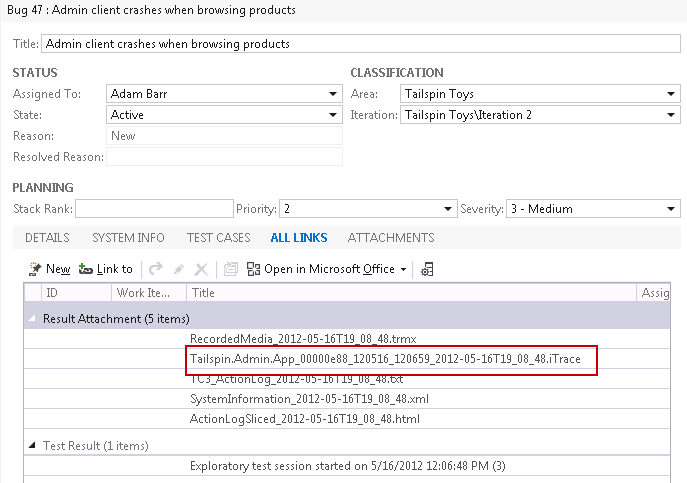


Figure - Locate the .iTrace file in Visual Studio.

1. Next double-click the exception to start debugging using IntelliTrace.

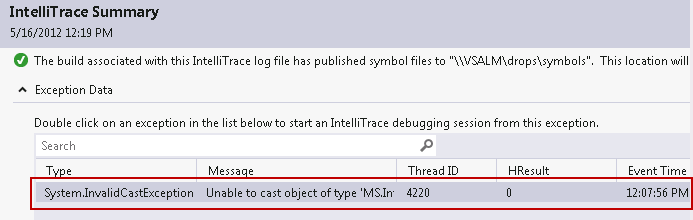


Figure - Start the IntelliTrace debugger from the .iTrace file.

1. A dialog is shown informing us that Visual Studio is unable to find the source symbols for the application being debugged.

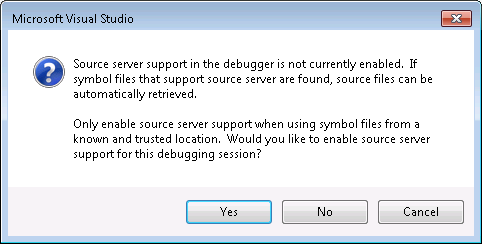


Figure - Source server support not enabled dialog.

1. To fix this you need to configure Visual Studio to locate the missing symbol files, which is the goal for the next exercise.

|  |  |
| --- | --- |
| Review | This exercise has shown you how Visual Studio presents IntelliTrace debug information if symbol data is not available. |

# Exercise 3 – Configure Visual Studio to locate the symbols

|  |  |
| --- | --- |
| Objective | In this exercise you will configure Visual Studio to locate source symbols. To do this you need to do two things, first register the location of the symbol files and next enable source server support in Visual Studio. |

1. First share the location of the symbol files so we can configure Visual Studio to load from the symbol location.
   1. Open a Windows Explorer, select the **c:\drops** folder and share the folder as **Drops**.

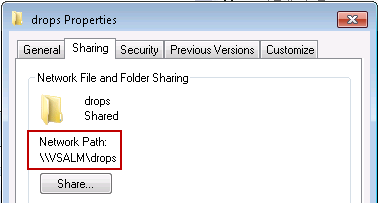


Figure - Create file share for symbols.

1. Open the debug symbols settings from Tools🡪Debugging🡪Symbols dialog in Visual Studio.
2. Add the **\\vsalm\drops\symbols** folder to the list of symbol file locations.

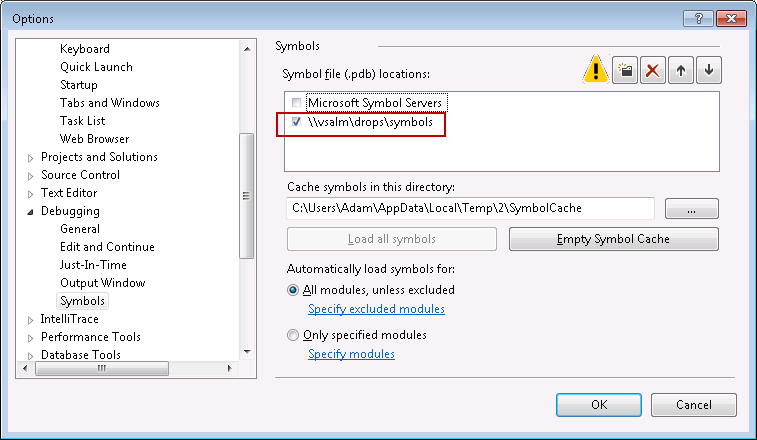


Figure - Add symbol share to debugging settings.

1. Go to the Tools🡪Debugging🡪General dialog in Visual Studio and check the “Enable source server support” and “Always run untrusted source server commands without prompting” settings.

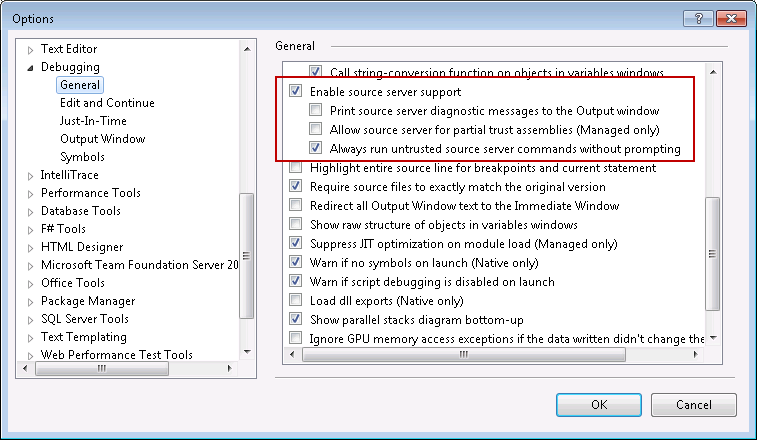


Figure - Enable source server support in debugging settings.

1. Close the dialog.

|  |  |
| --- | --- |
| Review | You have now seen what you need to do to configure Visual Studio to locate symbol files and to enable source server support. |

# Exercise 4 – Run IntelliTrace again and have Visual Studio locate the symbols

|  |  |
| --- | --- |
| Objective | In this exercise you will repeat exercise 2 and now Visual Studio will locate source symbols correctly. |

1. Switch back to the bug in Visual Studio and double click on the .iTrace file.
2. Double-click the exception to start the IntelliTrace debugger.
3. Notice how the symbols got loaded and the debugger breaks at the line of code causing the problem. Thanks to the Source server integration (i.e. TFS) it is also the matching version of the file that was used to build the version of the app being debugged that is displayed.

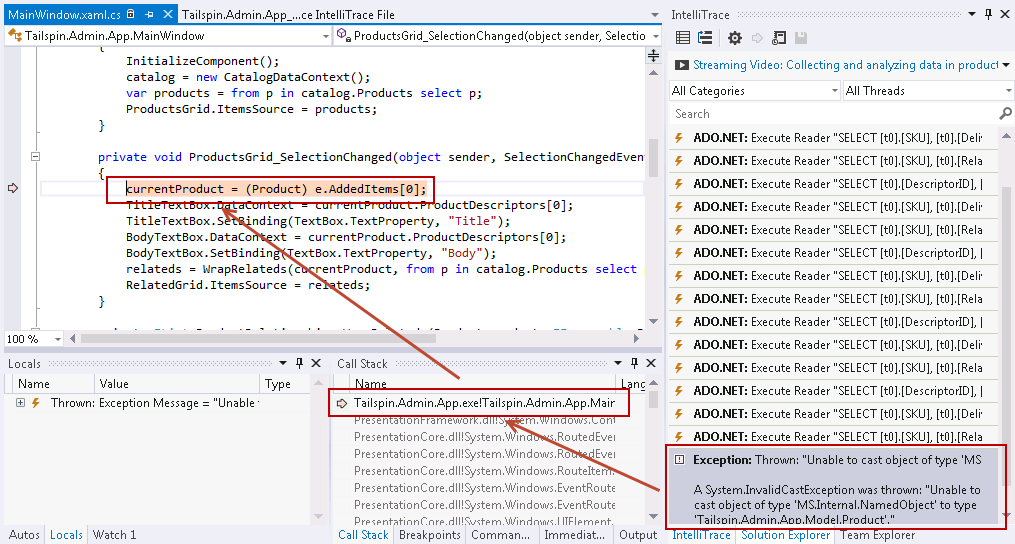


Figure - IntelliTrace debugger successfully loaded symbols.

|  |  |
| --- | --- |
| Review | With symbol location and source service configured correctly Visual Studio will now be able to locate and load the symbol information as well as to find and load source code files. |

# Appendix

## Other ALM Rangers Resources

Understanding the ALM Rangers – <http://aka.ms/vsarunderstand>

Visual Studio ALM Rangers Home Page – <http://aka.ms/vsarmsdn>

Visual Studio ALM Ranger Solutions – <http://aka.ms/vsarsolutions>

## Code Reference

**No table of figures entries found.**

## Figures Reference

[Figure 1 - Connect MTM to the Tailspin Toys team project. 5](#_Toc353274726)

[Figure 2 - Start exploratory testing. 5](#_Toc353274727)

[Figure 3 - Explore with full diagnostics. 6](#_Toc353274728)

[Figure 4 - Application under test failed with exception. 6](#_Toc353274729)

[Figure 5 - Create new bug in MTM. 7](#_Toc353274730)

[Figure 6 - Fill out the bug report and make sure iTrace file is attached. 7](#_Toc353274731)

[Figure 7 - Locate the .iTrace file in Visual Studio. 8](#_Toc353274732)

[Figure 8 - Start the IntelliTrace debugger from the .iTrace file. 8](#_Toc353274733)

[Figure 9 - Source server support not enabled dialog. 9](#_Toc353274734)

[Figure 10 - Create file share for symbols. 10](#_Toc353274735)

[Figure 11 - Add symbol share to debugging settings. 10](#_Toc353274736)

[Figure 12 - Enable source server support in debugging settings. 11](#_Toc353274737)

[Figure 13 - IntelliTrace debugger successfully loaded symbols. 12](#_Toc353274738)

## Tables Reference

**No table of figures entries found.**

1. <http://aka.ms/vsarindex> [↑](#footnote-ref-2)
2. <http://blogs.msdn.com/b/briankel/archive/2011/09/16/visual-studio-11-application-lifecycle-management-virtual-machine-and-hands-on-labs-demo-scripts.aspx> [↑](#footnote-ref-3)