PRE*f*ast for Drivers: WHDC Lab

December 20, 2007 — WHDC Preview Draft Version 0.9

Abstract: This lab provides an introduction to PREfast for Drivers for people who have not used this tool before. The lab help you become familiar with running PREfast for Drivers and analyzing some of the warnings it produces.

Feedback: Please tell us whether this preview lab is useful to you. Give us your comments at <http://go.microsoft.com/fwlink/?LinkId=101534>.

Contents

[Lab Overview 3](#_Toc180509978)

[Time to Complete This Lab 3](#_Toc180509979)

[Lab Objectives 3](#_Toc180509980)

[Prerequisites 3](#_Toc180509981)

[Feedback for This Lab 3](#_Toc180509982)

[Lab Setup 3](#_Toc180509983)

[Required Files for the Lab 4](#_Toc180509984)

[Tips for This Lab 4](#_Toc180509985)

[Exercise 1: Running PRE*f*ast 5](#_Toc180509986)

[Task 1: Build the PRE*f*ast Examples 5](#_Toc180509987)

[Exercise 2: Displaying PRE*f*ast Results 7](#_Toc180509988)

[Task 1: Use the PRE*f*ast Defect Log Viewer 7](#_Toc180509989)

[Task 2: Use the Message List Window 7](#_Toc180509990)

[Task 3: Use the View Annotated Source Window 8](#_Toc180509991)

[Task 4: Use the Message List Window in Filter View 9](#_Toc180509992)

[Task 5: Filter PRE*f*ast Results 10](#_Toc180509993)

[Task 6: Hide Individual Messages 10](#_Toc180509994)

[Task 7: Review PRE*f*ast Defect Log Text Output 11](#_Toc180509995)

[Learn More 12](#_Toc180509996)

[Try More PRE*f*ast Examples 12](#_Toc180509997)

[Understanding the PRE*f*ast Analysis Mode 12](#_Toc180509998)

[Resources 13](#_Toc180509999)

Disclaimer

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

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

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2007 Microsoft Corporation. All rights reserved.

Microsoft, MSDN, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

# Lab Overview

This lab provides an introduction to PREfast for Drivers (PFD) for people who have not used this tool before.

In this lab, you will use PREfast for Drivers on intentionally incorrect sample code to learn how to run PREfast and to examine the issues that it finds.

## Time to Complete This Lab

This lab has been designed to be completed in 20 to 30 minutes.

## Lab Objectives

After completing this lab, you will be able to:

* Run PREfast for Drivers.
* Examine the results and determine what should be fixed.

## Prerequisites

* Know how to use a Windows® Driver Kit (WDK) build environment window.
* Be familiar with the basics of static verification and PREfast as described in:

"Static Analysis and Verification of Drivers" on the WHDC Web site.

"Static Analysis Tools: PREfast for Drivers" on the WHDC Web site.

Chapter 23: "PREfast for Drivers," in Developing Drivers with the Windows Driver Foundation from Microsoft Press.

## Feedback for This Lab

To provide feedback about this lab:
 <http://go.microsoft.com/fwlink/?LinkId=101534>

# Lab Setup

PREfast runs on Windows XP and later versions of the operating system. The tool can be used in free and checked build environments for x86 and x64 architectures. This lab requires the following software and settings:

* WDK version 6001 or later.

Be sure to install the full WDK on your machine. For information, see "Windows Driver Installation Guide" Install.htm on the WDK installation media.

At the time of this publication, the Windows Server® 2008 WDK build 6001 was in beta, but is available from MS Connect. The final release of Windows Server 2008 WDK is provided through MS Connect and MSDN®. For information about obtaining current builds of the WDK, see "How to Get the Windows Driver Kit and the Windows Logo Kit."

* Permissions to create, edit, and delete files in the %wdk%\tools\pfd\samples directory, where %wdk% is the root installation directory for the WDK—typically C:\WinDDK\6001.

## Required Files for the Lab

This lab requires the following files and folders.

|  |  |
| --- | --- |
| File | Location |
| Sample source code  | %wdk%\tools\pfd\samples |
| Fail\_drivers samples | %wdk%\tools\pfd\samples\fail\_drivers |

## Tips for This Lab

* If you copy code from the online version of this manual, paste it into Notepad and then copy it from Notepad into your driver.
* The warning numbers in blue in the viewer window are hotlinks to the documentation that describes each warning.

To open a build environment window

1. Click Start, and then click All Programs.

2. Click Windows Driver Kits, click the latest WDK version (which is 6001 in this lab), and then click Build Environments.

3. Click the appropriate CPU architecture, and then open a checked or free build environment window for the appropriate Windows version.

The build environment window for a specified version of Windows works for that version and all later versions.

For this exercise, use Windows Vista and Windows Server 2008 x86 Free Build.

To do this, navigate to:
Start > All Programs > Windows Driver Kits > WDK 6001 > Build Environments > Windows Vista and Window Server 2008 > Windows Vista and Window Server 2008 x86 Free Build Environment

To get help with PREfast

* If you are working in a WDK build environment window and want to view a summary of available command-line commands, type:
prefast /?
—or—
prefast /help
* To view the PREfast for Drivers help file, type:
prefast help

In the documentation window, navigate to:
PRE***f***ast for Drivers > PRE***f***ast For Drivers Overview > Running PRE***f***ast for Drivers

# Exercise 1: Running PRE*f*ast

You can use PREfast to analyze kernel-mode drivers, other kernel-mode components, and user-mode drivers. PREfast is installed with the WDK. You do not need to take any additional steps to install PREfast. By default, PREfast analyzes code according to rules for kernel-mode components.

This section provides a brief introduction to building a driver with PREfast.

## Task 1: Build the PREfast Examples

About the PRE*f*ast Examples. PREfast is installed with the pfd\samples directory, which contains intentionally incorrect source code examples that trigger various PREfast warnings. You can use the PREfast examples to validate your PREfast installation and to experiment with the PREfast defect log viewer. The pfd\samples\fail\_drivers subdirectory contains additional driver source code that illustrates driver-specific rules in more depth.

When you run PRE*f*ast, you:

* Specify the appropriate version of PREfast by choosing the related WDK build environment.
* Use the cd command to set the default directory where you build your source code.

For example, if you are building a driver, you would set the default directory to one that contains a sources file or a dirs file.

* Use the prefast command-line command with the same Build utility parameters that are required to build your code.

To build the PREfast examples

1. Open the build environment window for Windows Vista® and Windows Server 2008 x86 Free Build Environment.

2. Make the tools\pfd\samples directory the default directory by typing the following, replacing %wdk% with the root WDK installation directory:

cd %wdk%\tools\pfd\samples

3. At the command prompt, type the following command to build the examples:

prefast build -cZ

The resulting object files are the same as those produced by the usual build -cZ command.

The command window output in the following listing shows the results of building the PREfast samples. The errors reflect deliberate errors in the examples:

Building PREfast samples—Command window output

C:\WINDDK\tools\pfd\samples>prefast build -cZ

-------------------------------------------------------------

Microsoft (R) PREfast Version 8.0.xxxxx.

Copyright (C) Microsoft Corporation. All rights reserved.

-------------------------------------------------------------

BUILD: Compile and Link for x86

BUILD: Start time: Mon Dec 04 14:37:10 2006

BUILD: Examining c:\winddk\tools\pfd\samples directory for files to compile

 c:\winddk\tools\pfd\samples

BUILD: Compiling c:\winddk\tools\pfd\samples directory

\_NT\_TARGET\_VERSION SET TO WINXP

Compiling - bounds-examples.cpp

Compiling - pft-example1.cpp

Compiling - pft-example2.cpp

Compiling - pft-example3.cpp

Compiling - precedence-examples.cpp

Compiling - hresult-examples.cpp

Compiling - drivers-examples.cpp

Compiling - bounds-examples.cpp

Compiling - pft-example1.cpp

Compiling - pft-example2.cpp

Compiling - pft-example3.cpp

Compiling - precedence-examples.cpp

Compiling - hresult-examples.cpp

Compiling - drivers-examples.cpp

Compiling - generating code...

Building Library - objchk\_wxp\_x86\i386\prefastexamples.lib

BUILD: Finish time: Mon Dec 04 14:37:19 2006

BUILD: Done

 16 files compiled

 1 library built

-------------------------------------------------------------

Removing duplicate defects from the log...

-------------------------------------------------------------

PREfast reported 31 defects during execution of the command.

-------------------------------------------------------------

Enter PREFAST LIST to list the defect log as text within the console.

Enter PREFAST VIEW to display the defect log user interface

.

**About Running PRE*f*ast.** When you build a component with the **prefast** command, PREfast:

* Intercepts the call to the compiler.
* Analyzes the code to be compiled.
* Writes results of the analysis to a log file, which is stored as XML.

The Defects.xml default log file is written to %wdk%\tools\pfd. To write the log file to another location, use the /LOG= switch with the prefast command.

PREfast operates separately on each function in the source code. It:

* Produces a single combined log for all of the files that are checked in a single run.
* Eliminates duplicate errors and warnings that header files generate.
* Calls the regular compiler to produce the usual build output.

Tip: For comparison to the sample code that triggers PREfast warnings, the Bounds‑examples.cpp file contains several functions that do not have errors and so do not trigger any PREfast warnings. Look in the source code for functions with "\_ok" in the function name.

# Exercise 2: Displaying PRE*f*ast Results

This exercise uses the PREfast output that you created in Exercise 1.

You can display the log file results of the PREfast analysis in two ways:

* In the PREfast defect log viewer.
* As text output.

In this exercise, you will practice viewing and filtering PREfast output results.

## Task 1: Use the PRE*f*ast Defect Log Viewer

The PREfast defect log viewer provides a graphical user interface that you can use to review PREfast output, to filter output so you can show or hide particular messages, and to view annotated source code so you can see the analysis path that produced a given warning.

To display results in the PREfast defect log viewer

1. Run PREfast on your source code, as described in Exercise 1.

2. In the build environment window, type the following:

prefast view

The PREfast defect log is displayed in a **Message List** window.

Troubleshooting Tip: If a User Account Control message appears, restart the build environment with elevated privileges.

## Task 2: Use the Message List Window

Figure 1 shows the Message List window with all of the PREfast output from building the examples in Task 1. The version number at the top of the window indicates the version of PREfast that displays the log.



Figure 1. PREfast Message List window

To experiment with the Message List window

1. To sort the messages in alphabetic or numeric order, click a column heading:
Description
Warning
Source Path
Source Location
In Function

2. To display the code that triggered a particular message, double-click the message to open the View Annotated Source window.

3. To filter messages, click the Filter button. Filtering removes specified types of messages from the message list.

You can choose from a list of predefined filters or show and hide individual messages.

## Task 3: Use the View Annotated Source Window

If you double-click a message in the Message List window, the View Annotated Source window appears, as shown in Figure 2.

The View Annotated Source window displays annotated source code for the error that triggered that message, with a few lines of code before and after, for context.



Figure 2. PREfast View Annotated Source window

To experiment with the View Annotated Source window

1. To display annotated source code for other messages, click Prev or Next.

Click Msg List to return to the Message List window.

2. To display annotated source code for the entire file that contains the error, on the View menu, click Show Entire File.

3. Click the warning number to display PREfast documentation that describes the problem in detail.

4. Under Go to:

Click Start of Function to view the beginning of the function in the Annotated Source window.

Click Start of Path to go to the beginning of the PREfast analysis path.

Click Warning Line to go to the line that triggered the warning.

To view detailed information about warnings

* Click the warning number.

In the PREfast viewer, the text "warning nnnn" is a hyperlink to the PREfast for Drivers documentation in the WDK. For many warnings, the documentation provides significant insights into the precise nature of the warning and often suggests how to fix the problem. If you are unfamiliar with a particular warning number, read the documentation—it can save you a lot of time.

## Task 4: Use the Message List Window in Filter View

When you click Filter, a list of messages that can be filtered appears above the list of messages that was generated by building your code, as shown in Figure 3.

To view a filtered list of messages

* In the Message List window, click Filter.



Figure 3. PREfast Message List window in filter view

To experiment with the Message List window in filter view

1. Choose from a list of predefined filters in Presets to show only the messages selected by that filter.

2. To hide a message in the message list:

In the message filter pane, clear the check box next to the message.

—Or—

Select the message and click Invert.

3. To update the message list, click Apply.

The list then shows only messages selected in the message filter pane.

4. To hide filters, click the Filter button again.

The message list continues to show only messages that are selected in the message filter pane. Filters are "sticky" from one PREfast run to the next.

To display the View Annotated Code window for a specific message

* Double-click the message.

This works the same as when filters are not visible.

## Task 5: Filter PRE*f*ast Results

Filtering results does not prevent PREfast from finding errors. Rather, it simplifies the list of results in the PREfast viewer so you can work with them more effectively. If you have a limited amount of time to fix errors that PREfast detects in your driver, use the predefined drivers\_recommended or drivers\_only filter and concentrate on fixing the errors that it displays.

To experiment with PREfast viewer in filter view

1. In the Message List window, click Filter.

2. In the list of predefined filters, click drivers\_recommended and then Apply.

This filter displays messages for serious errors in both general-purpose code and driver code.

These messages identify errors that tend to be genuine rather than messages that might not represent actual errors in code, which are often referred to as false positives or "noise."

3. In the list of predefined filters, click drivers\_only and then click Apply.

This filter displays messages only for errors that apply specifically to drivers.

Tip: After you fix the errors shown in the filtered results, you should always run PREfast again and change the way in which results are filtered, so that you can see and fix other less critical errors.

## Task 6: Hide Individual Messages

You might want to hide individual messages for several reasons, such as:

* Your development team might think that the risk associated with a message is acceptably low or the noise is unacceptably high.
* Your product ship cycle might allow fixing only the most critical errors.
* The messages simply might be irrelevant to your project.

For example, certain PREfast warnings that apply to kernel-mode drivers are also triggered by user-mode drivers. If you are testing a user-mode driver, you might want to hide kernel-mode driver messages such as the following:

Warning 28110: Drivers must protect floating point hardware state.

See use of float <expression>

Warning 28111: The IRQL where the floating point state was saved

does not match the current IRQL (for this restore operation)

Warning 28146: Kernel mode drivers should use ntstrsafe.h, not strsafe.h

To hide an individual message

* Clear its check box in the message filter pane as shown in Figure 3.

## Task 7: Review PREfast Defect Log Text Output

You can use the prefast list command to display the contents of the PREfast defect log as text output in the build environment command window.

This command is useful if you need only a short list of errors and do not need access to annotated source code—for example, to see the effect of fixing errors that PREfast found in a previous run.

The prefast list output shows the same information as the PREfast defect log viewer, but in a form suitable for pasting into files or bug reports. If you have previously filtered the results, the prefast list output shows the same filtered messages.

To display PREfast results as text output

1. Run PREfast on your source code, as described earlier.

2. At the command prompt, type the following:

prefast list

PREfast displays the filtered message list in the command window.

The example in the following listing shows text output for the first few messages from building the PREfast examples.

PREfast examples: prefast list command output

C:\WINDDK\tools\pfd\samples>prefast list

--------------------------------------------------------------------

Microsoft (R) PREfast Version 8.0.100006.

Copyright (C) Microsoft Corporation. All rights reserved.

--------------------------------------------------------------------

Contents of defect log:

C:\Documents and Settings\<username>\ApplicationData\Microsoft\PFD\defects.xml

--------------------------------------------------------------------

c:\winddk\tools\pfd\samples\bounds-examples.cpp

(45): warning 6029: Possible buffer overrun in call to 'fgets':

use of unchecked value 'line\_length'

 FUNCTION: read\_size (40)

c:\winddk\tools\pfd\samples\bounds-examples.cpp (54): warning 6057:

Buffer overrun due to number of characters/number of bytes mismatch

in call to 'wcsncpy'

 FUNCTION: unicode\_misuse (51)

c:\winddk\tools\pfd\samples\bounds-examples.cpp (62): warning 6201:

Index '10' is out of valid index range '0' to '9' for possibly

stack allocated buffer 'arr'

 FUNCTION: constant\_index (59)

c:\winddk\tools\pfd\samples\drivers-examples.cpp (23): warning 28125:

The function 'ProbeForRead' must be called from within a try/except

block: The requirement might be conditional.

 FUNCTION: drivers\_test2 (21)

# Learn More

## Try More PRE*f*ast Examples

You can try PREfast on the WDK Failed Driver samples to see more results from running PREfast.

1. In the build environment window, navigate to the WDK "Failed Driver" sample source files. These can be found at:

%wdk%\tools\pdf\samples\fail\_drivers
%wdk%\tools\sdv\samples\fail\_drivers

2. At the command prompt, type:

prefast build -cZ

Use the Defect Viewer to review the results.

## Understanding the PREfast Analysis Mode

The PREfast analysis mode determines which set of rules PREfast uses when it analyzes code. The analysis-mode annotation—defined in %wdk%\inc\ddk\driverspecs.h—informs PREfast whether a particular body of code is user-mode or kernel-mode code and whether the code is actually a driver. This annotation applies to an entire source file.

The analysis mode annotation can be one of the following:

\_\_kernel\_driver

For kernel-mode driver code.
This is the default mode, which is used in this lab.

\_\_kernel\_code

For nondriver kernel-mode code.

\_\_user\_driver

For user-mode driver code.

\_\_user\_code

For nondriver user-mode code.

By default, PREfast uses the \_\_kernel\_driver analysis mode. If this mode is inappropriate for your project, you should insert the appropriate analysis mode annotation in the source file or appropriate header file. For more information, see the PREfast documentation.

# Resources

For questions about PREfast for Drivers, send e-mail to: pfdfdbk@microsoft.com

To find out more about PREfast for Drivers, see the following resources.

### Tools and Files:

PRE*f*ast in the WDK

*%wdk%*\tools\pfd—Build 6001 or later version

### WinHEC Presentations:

Static Analysis Tools: PRE*f*ast for Drivers

This is an advanced session on the use of annotations

### References:

Chapter 23: "PRE*f*ast for Drivers," in Developing Drivers for the Windows Driver Foundation

<http://www.microsoft.com/whdc/driver/wdf/wdfbook.mspx>

How to Get the Windows Driver Kit and the Windows Logo Kit

<http://www.microsoft.com/whdc/DevTools/WDK/WDKpkg.mspx>

PRE*f*ast for Drivers on the WHDC Web site

<http://www.microsoft.com/whdc/DevTools/tools/PREfast.mspx>
Including:

PREfast Step-by-Step
English: <http://www.microsoft.com/whdc/DevTools/tools/PREfast_steps.mspx>

Links to Chinese and Japanese translations are also available on this page

PREfast Annotations
<http://www.microsoft.com/whdc/DevTools/tools/annotations.mspx>