Static Driver Verifier for KMDF Drivers: WHDC Lab

December 20, 2007 — WHDC Preview Draft Version 0.9

Abstract: This lab introduces the basic features of Static Driver Verifier (SDV) and demonstrates how to use SDV to examine a kernel-mode driver framework (KMDF) driver.

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](#_Toc180508185)

[Time to Complete This Lab 3](#_Toc180508186)

[Lab Objectives 3](#_Toc180508187)

[Prerequisites 3](#_Toc180508188)

[Feedback for This Lab 3](#_Toc180508189)

[Lab Setup 4](#_Toc180508190)

[Tips for This Lab 4](#_Toc180508191)

[Exercise 1: Preparing to Use SDV to Verify Fail\_Driver3 5](#_Toc180508192)

[Task 1: Confirm Role-Type Definitions 5](#_Toc180508193)

[Task 2: Process the Library 6](#_Toc180508194)

[Summary of Exercise 1 7](#_Toc180508195)

[Exercise 2: Using SDV to Verify KMDF Rules 7](#_Toc180508196)

[Task 1: Verify KMDF Rule Usage 7](#_Toc180508197)

[Summary of Exercise 2 9](#_Toc180508198)

[Exercise 3: Examining the SDV Results 9](#_Toc180508199)

[Task 1: Examine the Static Driver Defect Report Page 10](#_Toc180508200)

[Summary of Exercise 3 11](#_Toc180508201)

[Resources 11](#_Toc180508202)

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 introduces Static Driver Verifier (SDV), a compile-time testing tool that systematically tests all code paths through a driver to detect violations of kernel-mode driver framework (KMDF) rules. In completing the three exercises, you will:

* Prepare files for running SDV verification.
* Scan a sample driver that has intentional coding errors.
* Use the SDV Defect Viewer to review the results of the verification.

Each exercise builds upon the work that was completed in the previous exercise.

About the KMDF Fail\_Driver3 sample driver. This sample driver:

* Contains coding errors that SDV detects.
* Has one source file and one header file.
* Is installed with the Windows® Driver Kit (WDK) at:

%wdk%\tools\sdv\samples\fail\_drivers\kmdf\fail\_driver3

where %wdk% represents the root installation directory for the WDK.

## Time to Complete This Lab

This lab is designed to be completed in 45 minutes.

## Lab Objectives

After completing this lab, you will be able to:

* Prepare the build environment to use SDV to detect KMDF rule violations in the Fail\_Driver3 sample driver.
* Verify two KMDF rules: one that passes and one that fails.
* Use the results to trace defects in the driver code.
* Identify and correct common errors in KMDF drivers.

## Prerequisites

To get the most benefit from this lab, you should:

* Know how to use a WDK build environment window.
* Be familiar with SDV as described in the following resources:

Developing Drivers with the Windows Driver Foundation from Microsoft Press.

White papers about SDV on the WHDC Web site.

WDK documentation for SDV.

You can find links to the resources at the end of this lab manual.

## Feedback for This Lab

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

# Lab Setup

SDV runs on Windows XP and later versions of Windows. The tool supports free and checked build environments for both 32-bit and 64-bit architectures.

This lab requires the following software and settings:

* WDK version 6001 or later, which includes:

Static Driver Verifier in:  
%wdk%\tools\sdv

Fail\_Driver3 sample in:  
 %wdk%\tools\sdv\samples\fail\_drivers\kmdf\fail\_driver3

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 the 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."

* Permission to create, edit, and delete files in the driver's sources directory and in the \tools\sdv subdirectories of the WDK.

## Tips for This Lab

* If you copy code from the online version of this manual, paste it into an instance of Notepad and then copy it from Notepad into your driver.
* If you want to begin the lab in the middle or return to a recoverable state, you can start each exercise from where you last concluded.
* Be sure to install the full WDK on your machine.

See "Windows Driver Installation Guide" in Install.htm on the WDK installation media.

* If you are working in a build environment window, you can view a summary of available SDV commands by typing the following command:

staticdv /?

You can view the Static Driver Verifier Help file by typing the following command:

staticdv /help

* If your host computer is running Windows Vista®, the command window that you use for SDV must run with elevated privileges. To run an application or command window with elevated privileges:

On the taskbar, click Start, right-click the application, and then click Run As Administrator.

If you already have administrative privileges, Windows Vista displays a User Account Control dialog box asking for permission to proceed. To run the application, click Continue.

If you do not have administrative privileges, Windows Vista asks for administrative credentials.

To open a build environment window

1. Click Start, and then click All Programs.

2. Click Windows Driver Kits, click the latest WDK version (6001 or later), 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 by navigating to the following:

Start > All Programs > Windows Driver Kits > WDK 6001 >   
Build Environments > Windows Vista and Window Server 2008 >   
Windows Vista and Windows Server 2008 x86 Free Build Environment

All of the exercises for this lab are in the following directory:

%wdk%\tools\sdv\samples\fail\_drivers\kmdf\fail\_driver3

# Exercise 1: Preparing to Use SDV to Verify Fail\_Driver3

To prepare to use SDV on the Fail\_driver3 sample KMDF driver:

* Confirm that role types are defined for the driver’s entry points.
* Process the library.

## Task 1: Confirm Role-Type Definitions

To take advantage of SDV capabilities, the role type for each driver-supplied routine in a KMDF driver must be declared in one of the driver’s header files. Role types provide SDV with information about the intended use of a function, so that SDV can better determine whether a particular bug exists.

The callback function role types for KMDF drivers:

* Are defined in the KMDF header files at %wdk%\inc\wdf\kmdf\10.
* Follow a naming convention that is similar to placeholder function names in the WDK.

For example, the EVT\_WDF\_DEVICE\_D0\_EXIT role type corresponds to the placeholder function named EvtDeviceD0Exit.

To find documentation about a callback function, delete the characters "WDF" and "\_" (underscore) from the role type name and then search the WDK documentation for the resulting string.

For a detailed explanation of role-type declarations, see the StaticDV help file by typing staticdv /help and also see Chapter 24, "Static Driver Verifier," in Developing Drivers with the Windows Driver Foundation from Microsoft Press.

Note: Role types have been predefined for Fail\_driver3, so you do not need to create them for this lab, but you can view them to see how they are defined in a driver’s header file, as shown in the following steps.

To view role-types definitions for Fail\_driver3’s entry points

1. In the build environment window, navigate to the Fail\_Driver3 directory at:

%wdk% \tools\sdv\samples\fail\_drivers\kmdf\fail\_driver3\driver

2. Use Notepad to open the Fail\_driver3.h file.

3. Scroll through the file to view the role declarations, which are shaded in the following example:

#include <NTDDK.h>

#include <wdf.h>

#include "fail\_library3.h"

NTSTATUS

DriverEntry(

IN PDRIVER\_OBJECT DriverObject,

IN PUNICODE\_STRING RegistryPath

);

**EVT\_WDF\_DRIVER\_DEVICE\_ADD EvtDriverDeviceAdd;**

**EVT\_WDF\_IO\_QUEUE\_IO\_READ EvtIoRead;**

**EVT\_WDF\_IO\_QUEUE\_IO\_WRITE EvtIoWrite;**

**EVT\_WDF\_IO\_QUEUE\_IO\_DEVICE\_CONTROL EvtIoDeviceControl;**

**EVT\_WDF\_DRIVER\_UNLOAD EvtDriverUnload;**

**EVT\_WDF\_TIMER EvtTimerFunc;**

**EVT\_WDF\_REQUEST\_CANCEL EvtRequestCancel;**

## Task 2: Process the Library

You should verify the driver libraries and the driver code. Library verification is essential for determining whether a driver complies with SDV rules. For example, if verification does not include a driver's library code, a driver might appear to omit a required call that is part of the library, or the library might contain a call that the driver duplicates, thus causing an error such as releasing a lock twice.

Before you can include a library in the verification of a driver, SDV must process the library. Processing a library prepares the internal representation of the library for use in verifying the driver.

You should process all libraries that you create for use in your drivers. You do not need to process the Windows libraries that provide device driver interfaces (DDIs)—such as Wdm.lib, Wdf.lib, Bufferoverflow.lib, Hal.lib, Ntoskrnl.lib, Wmi.lib, and so on—because the SDV operating system model supplies verification stubs for functions in the Windows libraries.

To process the library

1. In the build environment window, navigate to the library for the KMDF Fail\_Driver3:

%wdk%\tools\sdv\samples\fail\_drivers\kmdf\fail\_driver3\library

2. To process the library, type the following SDV command:

staticdv /lib

The command output will be similar to the following example:

C:\WinDDK\6001\tools\sdv\samples\fail\_drivers\kmdf\fail\_driver3\library>

staticdv /lib

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

Microsoft (R) Windows (R) Static Driver Verifier Version 1.5.315.0

Copyright (C) Microsoft Corporation. All rights reserved.

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

Build 'library' ...Done

Publishing :fail\_library3.lib.li to c:\winddk\6001\tools\sdv\samples\fail\_driver

s\kmdf\fail\_driver3\library\objfre\_wlh\_x86\i386\fail\_library3.lib.li

The cache contains 1 library.

Compiled 'fail\_library3.lib' ...Done

For more information, see "Library Processing in Static Driver Verifier" in the WDK documentation.

## Summary of Exercise 1

Exercise 1 prepared for SDV verification of the driver through the following steps:

* Confirmed that the Fail\_driver3 header file contains role-type definitions, which SDV uses to identify the entry points in the driver.
* Prepared the driver’s library for SDV verification.

# Exercise 2: Using SDV to Verify KMDF Rules

After preparing for SDV, the next step is to identify the rules that SDV verifies in the driver’s code.

## Task 1: Verify KMDF Rule Usage

The Fail\_Driver3 sample is a function driver, but it does not correctly call the WdfDeviceInitSetPowerPolicyEventCallbacks power management function. Therefore, the Fail\_Driver3 sample passes SDV verification for the KMDF rule named FDODriver but violates the FDOPowerPolicyOwnerAPI rule.

Important: Be sure to type the staticdv commands exactly as shown in these steps. Do not add extra spaces.

To verify KMDF rule usage

1. In the build environment window, navigate to the following folder:

%wdk%\tools\sdv\samples\fail\_drivers\kmdf\fail\_driver3\driver

2. To verify the precondition that the driver creates a functional driver object (FDO), type the following command:

staticdv /rule:FDODriver

3. Examine the results that appear in the build environment window, and verify that "1 Rule Passes" appears as shown in the following example:

\C:\WINDDK\6001\tools\sdv\samples\fail\_drivers\kmdf\fail\_driver3\driver>

staticdv /rule:FDODriver

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

Microsoft (R) Windows (R) Static Driver Verifier Version 1.5.315.0

Copyright (C) Microsoft Corporation. All rights reserved.

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

Build 'driver' ...Done

Link 'driver' for [fail\_library3.lib] ...Done

Scan 'driver' ...Done

Warning 'driver' No entry points have been approved in SDV-map.h for this driver

Compile 'driver' for [sdv\_harness\_pnp\_io\_requests] ...Done

Link 'driver' for [fail\_library3.lib] ...Done

Check 'driver' for 'fdodriver' ...Running

Check 'driver' for 'fdodriver' ...Done

**Static Driver Verifier performed 1 check(s) with:**

**1 Rule Passes**

Start Time :1/25/2007 3:34:06 PM and End Time :1/25/2007 3:34:16 PM

Although a warning appears in the output, you have already verified that role types are declared for the driver callback. Therefore, you can safely ignore the warning.

4. Clean the driver directory by typing the following command:

staticdv /clean

5. To verify that the driver correctly calls the KMDF power-policy owner function, type the following command:

staticdv /rule:FDOPowerPolicyOwnerAPI

6. Examine the results and note that SDV reports a defect, as shown in the following example:

C:\WINDDK\6001\tools\sdv\samples\fail\_drivers\kmdf\fail\_driver3\driver>

staticdv /rule: FDOPowerPolicyOwnerAPI

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

Microsoft (R) Windows (R) Static Driver Verifier Version 1.5.315.0

Copyright (C) Microsoft Corporation. All rights reserved.

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

Build 'driver' ...Done

Link 'driver' for [fail\_library3.lib] ...Done

Scan 'driver' ...Done

Warning 'driver' No entry points have been approved in SDV-map.h for this driver

Compile 'driver' for [sdv\_harness\_pnp\_io\_requests] ...Done

Link 'driver' for [fail\_library3.lib] ...Done

Check 'driver' for 'fdopowerpolicyownerapi' ...Running

Check 'driver' for 'fdopowerpolicyownerapi' ...Done

**Static Driver Verifier performed 1 check(s) with:**

**1 Defect(s)**

Start Time :1/25/2007 3:45:43 PM and End Time :1/25/2007 3:45:54 PM

Note:  The FDOPowerPolicyOwnerAPI rule displays a warning about a possible defect. Some drivers establish power policy ownership based on external information that is outside the scope of SDV. In actual drivers, you must check the external information to verify whether the driver owns power policy for its device regardless of the results that SDV reports.

For more information, see the Serial sample driver at %wdk%\src\kmdf\serial. In the Serial sample, a registry setting determines whether the driver is a power policy owner, as defined in the driver’s SerialEvtDeviceAdd callback function.

## Summary of Exercise 2

The steps in Exercise 2 specify the rules that SDV verifies and provide preliminary results of the verification. To examine the detailed results, complete the next exercise.

# Exercise 3: Examining the SDV Results

If the preliminary results indicate defects in the driver, you can generate and examine a report in the Static Driver Defect Report Page, as shown in Figure 1.



Figure 1. Graphical User Interface for Static Driver Verifier

The interface contains the following elements:

**➀** The Trace Tree pane on the top left displays the sequence of the critical elements of the source code that are executed in the path to the rule violation.

**➁** The Source Code pane in the center highlights the corresponding line of code as you step through the source code in the Trace Tree pane. Trace fragments appear in red.

Each tab on the Source Code pane displays a step in the trace through the particular source code file.

**➂** The State pane at the bottom left lets you examine the values of variables in the driver, the SDV operating system model, and the rule. The values appear as Boolean expressions.

**➃** The status bar at the bottom of the window describes the defect.

**➄** The Results pane is shown in detail in Figure 2.

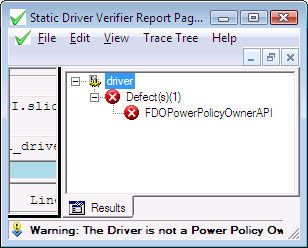


Figure 2. Results Pane in an SDV report

The Results pane shows the results of the verification, including a list of the defects that SDV discovered.

## Task 1: Examine the Static Driver Defect Report Page

For details about how to use Defect Viewer to filter and manage SDV results, see "Defect Viewer" in the WDK.

To examine the Static Driver Defect Report Page for Fail\_Driver3

1. In the build environment window, navigate to the following directory:

%wdk%\tools\sdv\samples\fail\_drivers\wdm\fail\_driver3\driver

2. To open the Static Driver Verifier Report Page, type the following command:

staticdv /view

3. To display the trace of the code path to the rule violation, double-click FDOPowerPolicyOwnerAPI in the Results pane on the right side of the Static Driver Verifier Report Page.

This is area **➄** in Figure 1.

4. To review the KMDF rule that the driver violated, examine the FDOPowerPolicyOwnerAPI.slic file by clicking the tab in the Source Code pane in Defect Viewer.

This is area **➁** in Figure 1.

5. In the Trace Tree pane, step through the execution path by clicking a line and examining the source code in the Source Code pane.

This is area **➀** in Figure 1.

6. Look for the following when you view the violation of the FDOPowerPolicyOwnerAPI rule for Fail\_Driver3:

Inside the EvtDriverDeviceAdd callback of the Fail\_Driver3 sample, the driver calls the SDVTest\_wdf\_FDOPowerPolicyOwnerAPI library function.

The SDVTest\_wdf\_FDOPowerPolicyOwnerAPI library function calls the WdfDeviceInitSetPowerPolicyOwnership function with FALSE as the second parameter, which specifies whether the driver is the power policy owner for its device stack.

Therefore, the driver is no longer the power policy owner, and SDV marks this information.

Inside the same library function, the WdfDeviceInitSetPowerPolicyEventCallbacks function is also called.

SDV reports a defect because only the power policy owner can call this function.

## Summary of Exercise 3

The steps of Exercise 3 provided:

* An introduction to the Static Driver Defect Report Page.
* An examination of code execution.
* Tips for identifying errors that SDV finds in source code.

# Resources

To find out more about SDV and KMDF rules, send e‑mail to:  
 sdvfdbk@microsoft.com

### Tools and Files:

SDV in the WDK

%wdk%\tools\sdv (Version 6001 or later)

Wdf.h header file

%wdk%\inc\wdf\kmdf

### WDK Sample Drivers Annotated for SDV:

OsrUsbFx2

%wdk%\src\kmdf\osrusbfx2

KMDF fail drivers

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

WDM fail drivers

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

### WDK Documentation:

Static Driver Verifier

<http://go.microsoft.com/fwlink/?LinkId=80084>

Library Processing in Static Driver Verifier

<http://go.microsoft.com/fwlink/?LinkId=80077>

Defect Viewer

<http://go.microsoft.com/fwlink/?LinkId=80066>

### Other Resources:

Static Driver Verifier - Finding Driver Bugs at Compile-Time

<http://go.microsoft.com/fwlink/?LinkId=80082>

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

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

Microsoft Research: SLAM

[http://research.microsoft.com/slam](http://research.microsoft.com/slam%20)

Chapter 24: "Static Driver Verifier," in Developing Drivers with the Windows Driver Foundation

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