

# **Developing Drivers with Visual Studio 2012**

**Advanced Developers Conference (adcpp.de)  
Bad Aibling, Germany  
May, 2013**

**Developed and Presented by :**

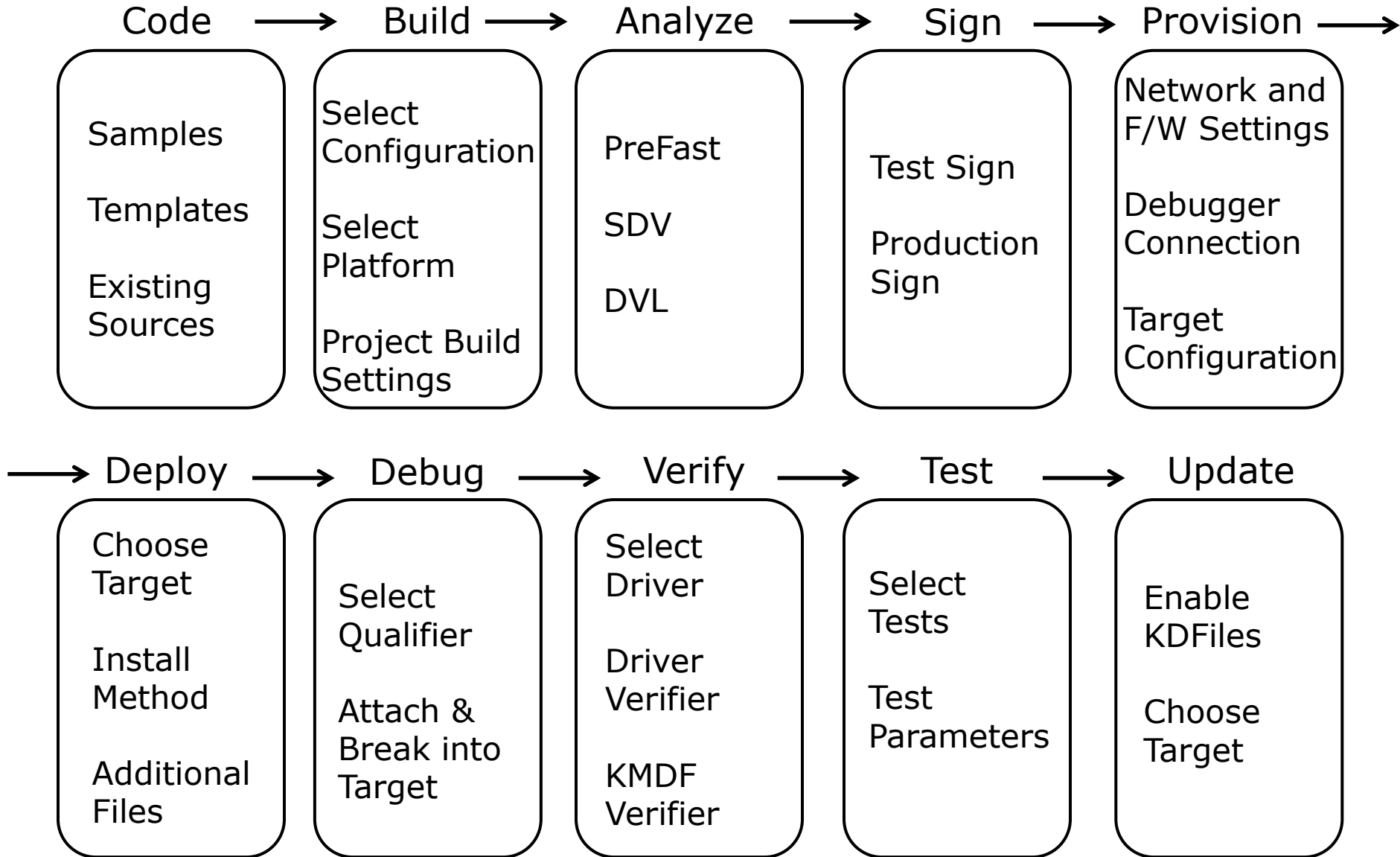
**CodeMachine**

**[www.codemachine.com](http://www.codemachine.com)**

# Introduction

- T.Roy
  - Masters Degree in Computer Engineering
  - 20 years experience in system software development
  - Author, instructor, consultant & founder of CodeMachine
- CodeMachine
  - Consulting and Training Company
    - [codemachine.com](http://codemachine.com), [twitter.com/codemachineinc](https://twitter.com/codemachineinc), [facebook.com/codemachineinc](https://facebook.com/codemachineinc)
  - Custom software development and debugging services
    - WDM, KMDF, Networking, Storage, File System & Security
  - Onsite and online training courses
    - Windows Internals for Application Developers
    - Windows Application Debugging & Memory Dump Analysis
    - Windows Internals for Driver Developers
    - Windows Driver Debugging & Memory Dump Analysis

# Agenda



# Demo Setup

- Host System Windows 8 Enterprise X64
- Visual Studio Professional 2012 (11.0.60315.01)
  - With Visual Studio 2012 Update 2
- .NET Framework 4.5.50709
- Windows Driver Kit 8.0 (6.2.9200.16384)
- Debugging Tools for Windows (6.2.9200.16384)
  - Part of Windows Driver Kit
- Virtualization Software is Virtual Box 4.2.0.r80737
- Target System VM is Windows 8 Basic X86

# VS 2012 Features

- Entire driver development workflow integrated in IDE
- Development system must be Win7 or above
  - Requires .NET Framework 4.5
- Compiler & Linker are kernel mode code aware
- Windows Driver Kit 8.0
  - No longer a self contained driver development package
    - Does not contain build toolset i.e. Compiler, Linker, etc.
  - Fully integrates into VS IDE
    - Must be installed "on top of" VS 2012
  - WDK Help is a part of VS 2012 documentation set
- Debugging Tools for Windows
  - Fully integrates into VS IDE
    - But also works as standalone debugging tool
  - Entire WinDBG command set supported by VS
  - Kernel memory dumps can be analyzed directly from VS

# Coding

- Driver Templates
  - VS2012 has the following built in driver templates
    - WDM Driver, File System Mini-Filter, NDIS Filter, KMDF Driver, KMDF USB Driver, UMDF Driver, UMDF USB Driver
- Driver Samples
  - No longer a part of the WDK
  - Downloaded separately from MS Download
    - Each sample can be downloaded individually
    - Package Containing all Samples
  - VS can enumerate & download samples within the IDE
  - Each sample has its own .SLN and .VCXPROJ files
    - Does not include "sources" and "dirs" files
- Existing Source Base
  - nmake2msbuild.exe converts old WDK "sources" & "dirs" files can be to new MSBuild format .vcxproj and .sln files

# Building

- Driver solutions have 2 projects in a single .sln file
  - Driver – Build involves stamping .INF, running ETW pre-processor, compiling, linking & test signing .SYS file
  - Driver Package – Build involves gathering driver package files, creating CAT file, signing package and deploying package
- Drivers can be build
  - Using VS 2012 IDE
  - Using Command Line Build Environment (MSBuild.exe)
    - Old build.exe and nmake.exe are no longer supported
- VS -> Build -> Configuration Manager
  - Platform = Win32 | x64
  - Configuration = Vista | Win7 | Win8 & Release | Debug
- If deployment is enabled for “Driver Package” build implies build & deploy to target system
  - WPP Tracing is recommended but optional

# Command Line Building

- Command line build environment fully supported
  - Uses the same engine as VS 2012 IDE
- VS 2012 Developer Command Prompt
  - `%comspec% /k ""C:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\Tools\VsDevCmd.bat""`
- In the project directory containing the
  - .sln file run `"MSBuild <Driver>.sln"`
  - .vcxproj file run `"MSBuild <Driver>.vcxproj"`
- Configuration and Platform specification
  - For Windows 8 Free Build X64
    - `msbuild /p:configuration="Win8 Release" /p:Platform=X64`
  - For Windows 7 Checked Build X86
    - `msbuild /p:configuration="Win7 Debug" /p:Platform=Win32`



# Code Analysis (PFD)

- Static Code Analysis Tool (aka OACR)
  - Finds common coding errors missed by compiler
  - Uses kernel mode driver specific rules
- Configuration Properties -> Code Analysis -> General
  - Run this rule set = Microsoft Driver Recommended Rules
  - Enabling Code Analysis on Build
    - Automatically runs code analysis every time driver is built
- Static analysis tools are run as VC Compiler plugins
  - /analyze:plugin "WindowsPrefast.dll"
  - /analyze:plugin "Drivers.dll"
- To suppress a particular code analysis message
  - Actions -> Suppress Message -> In Source
  - Adds "#pragma warning (supress:xxxx)" to source code
- Adding SAL 2.0 annotations in code helps SDV to make decisions about code defects

# Static Driver Verifier (SDV)

- Static Code Analysis Tool
  - Checks for adherence to DDI compliance rules
  - Performs code path analysis and attempting to reach paths not exercised under normal execution and testing
- Run SDV
  - VS -> Driver -> Launch Static Driver Verifier
    - Can be run only on release builds
  - From command line -> MSBuild /t:sdv
- Generate Driver Verification Log for HCT Submission
  - Contains a summary of the results from the Code Analysis and Static Driver Verifier log files
  - VS -> Driver -> Create Driver Verification Log
    - Driver project must be configured to build for X64 platform
  - From command line -> MSBuild /t:dvl

# Signing

- VS will sign driver package as a part of build process
  - Configuration Properties -> Driver Signing -> General
- Test Signing
  - Code Signing Certificate is automatically added to certificate store on development and target system
    - On development system -> "Personal"
    - On target system -> "Trusted Root Certificate Authorities"
    - Test signing must be enabled on target system
- Production Signing
  - Code Signing Certificate from Certificate Vendor
  - Cross Signing Certificate from Microsoft
    - C:\Program Files (x86)\Windows Kits\8.0\CrossCertificates
  - Driver Binary needs to be signed separately for Windows Vista, Windows 7 and Windows 8

# Provisioning

- Target/Test system must be configured before VS 2012 can deploy the driver package
- Requires administrative access to target
- Steps to configure the target system
  - Enable the Administrator Account and create a password
    - `net user administrator <password> /active:YES`
  - Turn on "Network Discovery"
    - `netsh advfirewall firewall set rule group="Network Discovery" new enable=yes`
  - Turn on "File and Printer Sharing"
    - `netsh advfirewall firewall set rule group="File and Printer Sharing" new enable=Yes`
  - Logon to the target system as an Administrator
    - Different account in the Administrators group will not work
  - Disable BitLocker on the OS partition

# Configuring Target

- Automatically performs the following steps
  - Creates a new user account "WDKRemoteUser" and logs into that account
  - Installation files are copied to %SystemDrive%\DriverTest
  - Installs Debuggers, .NET 4.5 Framework, TAEF, & WDTF
  - Copies DevCon, TraceLog, TraceView and other related files to the target
  - Multiple target systems can be provisioned together
- System "Qualifier" is specified at the time of deployment and debugging
- If connecting to target fails, check for access from host system
  - ping <TargetSystem>
  - dir \\<TargetSystem>\C\$

# Deploying

- Driver Install Options
  - Driver Package – Install driver package on target system
    - Driver Package Deployment needs to be performed just once
  - Driver – Transfer just the driver binary to target system
    - KDFiles Deployment (Currently broken in VS 2012 IDE)
    - Manually using .kdfiles in VS debugger window works
- Deployment can be enabled for either “Driver” or “Driver package” at any one time
- Driver package is copied to target system at C:\DriverTest\Drivers
  - Contents of this folder cleaned before every deployment
- Any other file specified via “Additional Files:” option is also copied to target system

# Deployment Options

- Do Not Install
  - Driver package is added to driver store
    - C:\Windows\System32\DriverStore\FileRepository
    - Running "pnputil -e" lists driver package
  - Device is not installed on system
- Hardware ID Update
  - Used only for root enumerated devices
  - Devcon installs device using specified root ID
    - devcon /r install <FullPathToINF> <RootDeviceId>
- Custom Command Line
  - Specify a custom command line to install the device
    - Command process should return exit code 0 for test pass
- Install and Verify – Default Driver Package Installation
  - Device is installed and installation is verified

# Debugging

- Kernel debugger is integrated in VS 2012
- Kernel Debugging requires 2 system setup
- Connection between systems (Debug Transports)
  - Network(Win8), USB 3.0(Win8), 1394, USB 2.0, Serial
- VS runs NTKD.exe as named pipe debug server to connect to target system using debug transport
  - NTKD hosts the debugger engine (DBGENG.dll)
- Connect to target system
  - VS -> Debug -> Attach to process
  - Transport : Windows Kernel Mode Debugger
  - Qualifier: Previously provisioned system
- Break into target system : VS -> Debug -> Break All
- Driver package deployment requires debugger to be disconnected from the target



# Verifying

- VS 2012 can enable Driver & KMDF Verifier on target system (both verifiers can be enabled simultaneously)
- Driver Verifier
  - Driver Package -> Properties -> Configuration Properties -> Driver Install -> Driver Verification
  - Options : Driver Selection, Standard Flags Scenario, Specific Settings
  - Can be directly enabled on target system using verifier.exe
- KMDF Verifier
  - Driver Package -> Properties -> Configuration Properties -> Driver Install -> KMDF Verification
  - Can be directly enabled on target system using wdfverifier.exe
  - Default setting for "Fail Memory Allocations" is 0
    - Causing every memory allocation in selected driver to FAIL!!!
  - Incorrect registry setting for "Track KMDF Object Handles"
    - For tracking "All WDF Handles" set "TrackHandles" value to "\*"

# Testing

- VS can execute “Device Fundamentals” and “Certification” tests on target system
  - To select individual tests : Driver -> Test -> Driver Test Group Explorer -> New Test Group -> Add or Remove Driver Tests
    - Parameters to individual tests within the test group can be specified via the UI
- In “Driver Test Group Explorer”, each “Driver Test Group NNNNN” shows the result from prior test runs
  - Results contain HTML report and TXT log
  - Log files are located in  
C:\Users\\AppData\Roaming\Microsoft\DriverTest\TestRepository\Results
- Tests can be run manually on target using TAEF
  - Useful for debugging individual test failures
  - C:\Program Files\Windows Kits\8.0\Testing\Runtimes\TAEF\TE.exe

# Updating

- Updating a driver binary already installed on target
  - Part of the Edit + Build + Test + Debug cycle
- Driver Package must not be enabled for deployment
- Configure Driver for KDFiles deployment
  - Driver binary must be present on target system
  - Driver must be forced to unload and then reload
  - Debugger transfers driver binary over debugger transport to target system
  - Replaces the driver binary in System32\Drivers
- Command to setup a new replacement map
  - `.kdfiles -m C:\Projects\TestDriver\Win8Debug\TestDriver.sys \Systemroot\system32\drivers\TestDriver.sys`
- Command to clear any previous replacement maps
  - `.kdfiles -c`

# VS 2012 Limitations

- Express (Free) version does not support driver development
  - Professional and Express versions can co-exist
- Building for XP/2003 is not supported with latest WDK
  - Use older WDK - Windows 7 SP1 WDK (7600.16385.1)
- Windows RT (ARM) driver development is not supported by the public release of WDK
  - Only available to OEMs building Windows RT systems
- Cannot attach debugger to Windows RT systems
  - Kernel Debugging disabled by OEMs via UEFI BIOS
- No support for headless VS build server
- Driver development features are relatively new and need some time to mature

## **COPYRIGHT © 2000-2013 CodeMachine Incorporated, USA**

Made in the United States of America. All rights reserved, worldwide.

### LIMITED WARRANTY

While every precaution has been taken in the preparation of this material, CodeMachine Inc. assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

### RESTRICTED RIGHTS

**This material may not be duplicated or distributed in any form or by any means without the express written permission of CodeMachine Inc.**

### TRADEMARKS

Windows 8, Windows Server 2012, Windows 7, Windows Vista, Windows Server 2008, Windows Server 2003, Windows XP, Windows 2000, Windows NT, Win32, Windows are registered trademarks of Microsoft Corporation. Other products and company names mentioned herein may be trademarks of their respective owners.

<http://www.codemachine.com>