

Beta, Beta Everywhere

Bill Sheldon

Blog: NerdNotes.net/blog

Twitter: NerdNotes

- Microsoft MVP - Visual Basic
- Author SQL Server Magazine
- Author Office Business Application Development (WROX)
- Instructor UCSD Extension
- Author Professional Visual Basic (WROX)
- Senior Technical Analyst-Developer Rubio's Restaurants

Session Focus

- ▣ This session is going to look at Visual Studio 2010.
 - Currently in Beta 1, but Beta 2 is coming fast
- ▣ How can you leverage the Visual Studio 2010 Beta(s)
 - VPCs are limited
 - There are many Beta's not just Visual Studio
- ▣ Focus will be on you feeling like you can create a system on your machine to host this and other Beta software

Where are we?

- ▣ We are in a Microsoft Release Wave
 - The wave will peak and come to shore in early 2010
 - Windows 7/ Windows Server 2008
 - Visual Studio 2010
 - Office 2010 (including SharePoint)
 - Silverlight 3 and possibly 4
 - updated Expression tools
 - ...
- ▣ The list continues, will this wave crash on you?
or will you ride the wave?

Managing Beta Software

- ▣ You don't want to install Beta software on your main 'machine'
 - It can break things that currently work
 - It never upgrades correctly and rarely uninstalls correctly
 - New releases show up more frequently than most of our development schedules so it expires quickly
 - Most of us don't have a spare physical machine we can rebuild to install next update
- ▣ Common answer in the past: VPC (or VM)

Virtual Machines

- ▣ Both VM Ware and VPC provide a good solution (Desktop focus here not Hyper-V)
 - Build a virtual environment and then use that to host the beta software
 - Can run side by side with your main OS
 - Is portable between machines
 - Is a performance Dog... at times unusable
- ▣ Even in the best cases (as I'll show) a VPC just doesn't have enough power for real work

Boot to VHD

- ▣ Builds on the VPC implementation, but removes the underlying OS
 - Boot into the virtual machine
 - VHD's on your system (or external eSata)
 - Gives you close to full system performance
 - Limits portability – VHD is mapped to your hardware (part of performance advantage)
 - Allows for easy 'tear-down' replacement based on baseline image

Boot to VHD

- ▣ Came out during the Windows 7 Beta
 - Lots of posts out there have extra steps that were needed during Beta
 - Avoid these there is a new CScript that does the job
- ▣ Scott Hanselman has the complete solution but its split in two posts (Bing: Hanselman VHD Boot)
 - Newer post (Step by Step) illustrates new MSDN tool (demo tonight)
 - Older post has still valid BCDEdit instructions you'll need but does manual creation process
- ▣ MSDN has the Windows Image to Virtual Hard Disk Converter
(code.msdn.microsoft.com/WIM2VHD)

Setting up Boot to VHD

- ▣ Windows 7 allows for a VHD to be mapped as a drive on your system.
 - Right click on 'computer' select 'Manage' and go to the Disk Management in the Computer Management console.
 - On the far right pane select More Actions from the Actions pane and select either Create or Attach VHD.
 - You'll need to recognize the VHD and assign it a drive letter.
- ▣ Boot to VHD leverages this capability.

Setting up Boot to VHD

- ▣ To automate the process of creating a VHD uses 2 tools
 - WIM2VHD – this is a CScript which handles file creation
 - Microsoft Automated Installation Kit (AIK) for Windows 7 – 1.5 Gig download from Microsoft
- ▣ You only need to install the AIK on the machine where you want to run WIM2VHD
 - It's an admin tool and for standard installs we can bypass its details

Setting up Boot to VHD

- ▣ Once the tools are installed get your Windows 7 installation disk
 - Can put in DVD drive or mount files on the network
- ▣ The only tool you run to create your new VHD is WIM2VHD with command line parameters
 - There is a documentation file with the tool – get it for all the parameter details

```
Wim2vhd.wsf /wim:E:\Sources\install.wim /sku:4  
/vhd:[path] /size:[in MB] /disktype:[fixed | dyna]
```

Setting up Boot to VHD

- ▣ Once the .VHD file has been created you are ready to map it to the Boot Manager
 - Need register valid VHD with BCDEdit
 - Does not support a USB external drive
 - Does support an eSata external drive (like mine)
- ▣ BCDEdit will default to your main OS (unless you explicitly change it)
 - If settings are bad, you'll get a message but still get into your system.
 - Can use delete to clean up mistakes
`bcdedit /delete {guid}`

Setting up Boot to VHD

- ▣ Copy your current settings

```
Bcdedit /copy {current} /d "VHD Windows 7"
```

- ▣ Note {current} is literal.
- ▣ Next need to update the newly copied entry to point to your VHD
 - This line requires [] around the drive letter
 - {guid} is not literal but the GUID copied from the results of the copy operation

```
Bcdedit /set {guid} device  
vhd=[F:]\vhdboot\my.vhd
```

Setting up Boot to VHD

- ▣ Once the device is mapped repeat for the OSDevice setting
- ▣ Finally need to have the image detect the Hardware Abstraction Layer

Bcdedit /set {guid} detecthal on

- ▣ That's it, you are ready to go a new machine in under 30 minutes (or as few as 15 with copy)
 - ~15 minutes to build a new VHD from scratch
 - ~5 min to set up BCDEdit (can move VHD files once this is done once.)
 - ~10 minutes for first startup and settings

Demo

Boot from VHD

Application Migration

- ▣ Isn't this a Beta discussion... Yes but you should approach VS 2010 as phase 1 of the migration of Visual Studio to .NET
 - Visual Studio was originally written predominately in C++, this release puts a WPF front end on VS
 - VS 2010 will still be a core C++ application and be limited to running in a 32-bit environment.
 - Note VS 2010 can create 64-bit applications but it runs as a 32-bit application.
 - VNext will include new compilers written using .NET

Visual Studio 2010 & .NET 4.0

- ▣ A major revision to Visual Studio and the .NET Framework
 - VS 2010 ships with .NET 4.0
 - Supports backward compatibility to .NET 2.0
 - Redesigned Editor based on WPF
 - Support for Dynamic Language Runtime (DLR)
 - Iron Python, Iron Ruby and F# become 'mainstream' languages
- ▣ New Team Suite tools and features
 - Historical debugging (note turned on by default in the beta)

C# and VB

- ▣ This is the first release to start to reflect the change in architecture/design direction
 - In the past C# and VB had different 'target audiences'
 - features were prioritized differently and one language would get or have 'features' that the other didn't.
- ▣ Starting with VS 2010 goal is to have VB & C# feature parity termed: CoEvolution
 - This is reflected in the features each language is getting in .NET 4.0

C# 4.0

- ▣ C# features include several capabilities familiar to VB developers
- ▣ Named Parameters
- ▣ Optional Parameters
- ▣ Variance/CoVariance
- ▣ Better Interop (dynamic import)
- ▣ Dynamic Type (IDynamicObject)
- ▣ Compilation without PIA's

VB 10.0

- ▣ VB is getting several new features to improve productivity and some that C# devs recognize
- ▣ Implicit line continuation
- ▣ Multi-line lambdas and no return value lambdas
- ▣ Auto-Implemented Properties
- ▣ Variance/CoVariance
- ▣ Interop with Dynamic languages (IDynamicObject)
- ▣ Compilation without PIAs
- ▣ Collection Initializers
- ▣ Array Literals (like XML literals)
- ▣ Nullable optional parameters

Summary

- ▣ Microsoft has multiple different products coming out in the next 9 months
- ▣ For developers Boot to VHD allows them to try out these products without corrupting their core system
- ▣ Visual Studio 2010 is in Beta 1, Beta 2 will arrive soon – you should start working with it
- ▣ The two main VS languages are moved to shared feature sets
- ▣ ...

Summary <cont.>

- ▣ WF and WCF are getting major updates
- ▣ Entity Framework is getting a ton of new features.
- ▣ RIDE THE WAVE – It's a blast and will keep your skills marketable
- ▣ Questions?...