

“Windows Server 8” and SMB 2.2 - Advancements in Management

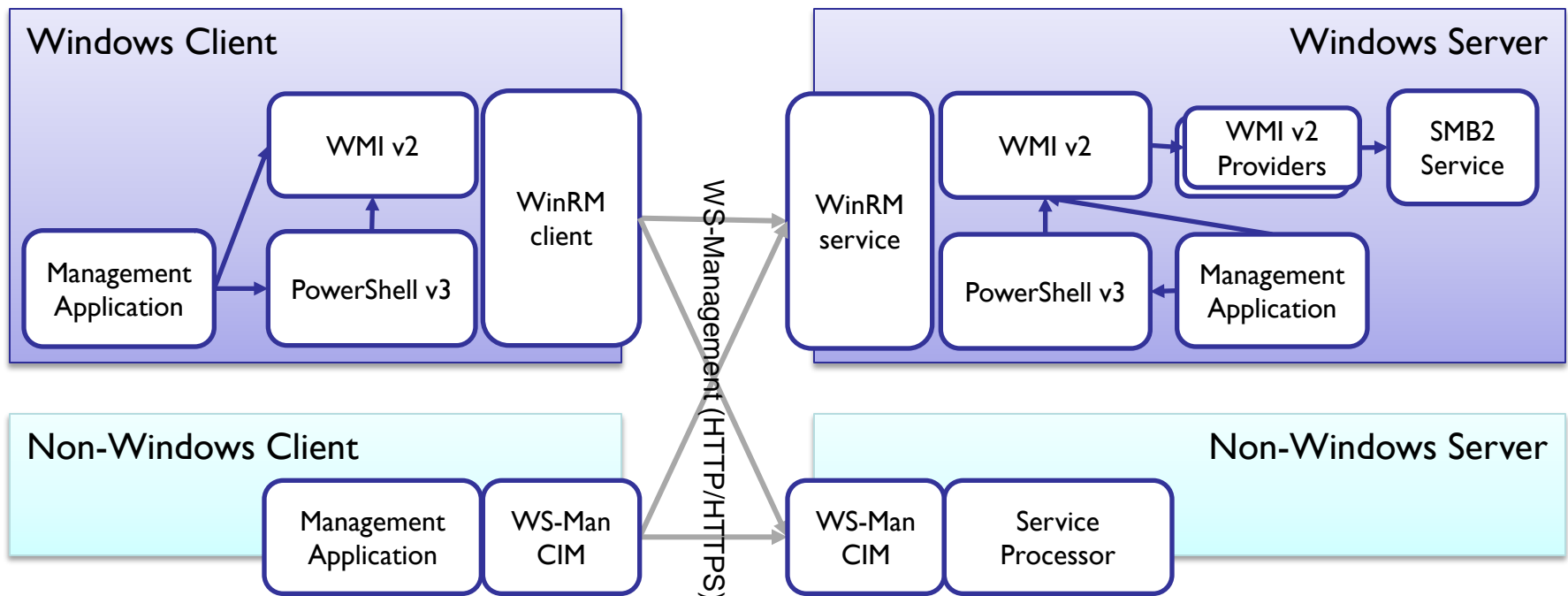
Jose Barreto
Principal Program Manager
Microsoft Corporation

Legal Disclaimers

- ❑ Products mentioned in this presentation are under development.
- ❑ Unless otherwise specified, the information included here refers to a pre-release version of the product known as “Windows Server 8” Developer Preview.
- ❑ Names in “quotes” are codenames used during development, not final product names.
- ❑ All information is subject to change.
- ❑ This presentation is for informational purposes only. Microsoft makes no warranties, express or implied, in this summary

SMB 2.2 Management – Short version

- ❑ “Windows Server 8” uses WMI v2 and PowerShell v3
- ❑ Manage from/into non-Windows via CIM/WS-Man
- ❑ New SMB 2.2 abilities **not** exposed via old services and APIs



- ❑ What we'll cover
 - ❑ The new WMI v2 classes and PowerShell v3 cmdlets for SMB 2.2 management, along with new a new Server Manager GUI, performance counters and events
 - ❑ The underlying changes in the “Windows Server 8” Windows Management Framework (WMF), including WMI v2, PowerShell v3, ODATA and POWWA
 - ❑ Overview of how you could plug into all of this
- ❑ Divided into two main parts
 - ❑ Specific “Windows Server 8” SMB 2.2 management advancements
 - ❑ General “Windows Server 8” management advancements

Note: This talk does not cover SMI-S. Jeff Goldner will be covering that topic in the “Microsoft SMI-S Roadmap Update” talk, right after this one.

Part I

“Windows Server 8” SMB 2.2 - Advancements in Management

Managing SMB in Windows 7

- ❑ Win32 API
 - ❑ Network Share Management API targeted at Developers.

- ❑ WMI v1 Provider
 - ❑ Works for both Developers and senior IT Administrators, with limitations.

- ❑ Registry Settings
 - ❑ Many SMB client and SMB server parameters are set directly in the Registry
 - ❑ HKLM\System\CurrentControlSet\Services\LanmanServer\Parameters
 - ❑ HKLM\System\CurrentControlSet\Services\LanmanWorkstation\Parameters.

- ❑ Command line: NET.EXE
 - ❑ Targeted at IT Administrators

- ❑ GUIs

SMB WMI v2 + PowerShell v3 - Overview

- Cmdlets and WMI classes to manage SMB File Server and SMB File Shares

- Allows an IT Administrator to
 - Use the PowerShell command line to manage and monitor File Servers and File Shares
 - Write scripts that automate common File Server administrative tasks
 - Leverage existing PowerShell skills
 - Integrate with related PowerShell-managed components (network, block storage, cluster)
 - Fully manage “Windows Server 8” SMB2 from non-Windows devices (POWWA)

- Allows a Developer to
 - Use a simpler API to manage and monitor File Servers and File Shares
 - Leverage existing skills to use WMI classes for management
 - Integrate with related WMI-managed components (network, block storage, cluster)
 - Fully manage “Windows Server 8” SMB2 from non-Windows devices (CIM, WS-Man)

SMB WMI v2 + PowerShell v3 – Goals

- ❑ **Administrator-friendly:** Consistent set of cmdlets, supersedes NET.EXE tool
- ❑ **Scripting-friendly:** Integrates with other PowerShell-managed components
- ❑ **Cluster-aware:** Manage both standalone and clustered file servers
- ❑ **Developer-friendly:** Enable developers, supersedes related Win32 Net API
- ❑ **GUI-friendly:** Predictable response time, if used by a GUI such as Server Manager
- ❑ **Remotable:** Able to manage a remote File Server by specifying its name
- ❑ **Properties-aware:** Enumerate all SMB share properties
- ❑ **Settings-aware:** Enumerate all SMB server and SMB client settings
- ❑ **Network Interface-aware:** Expose info on SMB2 Multichannel networks

SMB WMI Classes and associated Cmdlets

- SMB Share
 - Get-SmbShare
 - New-SmbShare
 - Set-SmbShare
 - Remove-SmbShare
- SMB Share Access
 - Get-SmbShareAccess
 - Grant-SmbShareAccess
 - Revoke-SmbShareAccess
 - Block-SmbShareAccess
 - Unblock-SmbShareAccess
- SMB Configuration
 - Get-SmbServerConfiguration
 - Set-SmbServerConfiguration
 - Get-SmbClientConfiguration
 - Set-SmbClientConfiguration
- SMB Session
 - Get-SmbSession
 - Close-SmbSession
- SMB Open File
 - Get-SmbOpenFile
 - Close-SmbOpenFile
- SMB Mapping*
 - Get-SmbMapping*
 - New-SmbMapping*
 - Remove-SmbMapping*
- SMB Connection*
 - Get-SmbConnection*
- SMB Network Interfaces
 - Get-SmbServerNetworkInterface
 - Get-SmbClientNetworkInterface
 - Get-SmbConnectionNetworkInterface
 - Update-SmbConnectionNetworkInterface
- Witness
 - Get-SmbWitnessCluster
 - Disconnect-SmbWitnessCluster
 - Get-SmbWitnessClusterClient
 - Get-SmbWitnessClusterResource
 - Move-SmbWitnessClusterClient

* Not in the Windows 8 Developer Preview

Comparing PowerShell and Net APIs

Net API call	SMB PowerShell
NetShareEnum NetShareGetInfo	Get-SmbShare
NetShareSetInfo	Set-SmbShare
NetShareAdd	New-SmbShare
NetShareDel NetShareDelEx	Remove-SmbShare
NetFileEnum NetFileGetInfo	Get-SmbOpenFile
NetFileClose	Close-SmbOpenFile
NetSessionEnum NetSessionGetInfo	Get-SmbSession
NetSessionDel	Close-SmbSession

Comparing PowerShell and NET.EXE

NET.EXE command	Equivalent SMB PowerShell
NET SHARE	Get-SmbShare New-SmbShare Set-SmbShare Remove-SmbShare
NET FILE	Get-SmbOpenFile Close-SmbOpenFile
NET SESSION	Get-SmbSession Close-SmbSession
NET USE	Get-SmbMapping* New-SmbMapping* Remove-SmbMapping*
NET CONFIG	Get-SmbServerConfiguration Set-SmbServerConfiguration Get-SmbClientConfiguration Set-SmbClientConfiguration

* Not in the Windows 8 Developer Preview

Sample SMB PowerShell cmdlets

```
Administrator: Windows PowerShell

PS C:\> Get-SmbShare

Name                ScopeName          Path                Description
-----
ADMIN$              *                  C:\Windows         Remote Admin
C$                  *                  C:\\               Default share
D$                  *                  D:\\               Default share
E$                  *                  E:\\               Default share
IOBW                *                  K:\IOBW            Remote Admin
IPC$                *                  K:\\               Remote IPC
K$                  *                  K:\\               Default share
VMS                  *                  K:\VMS             Default share

PS C:\> Get-SmbShareAccess VMS

Name                ScopeName          AccountName          AccessControlType  AccessRight
-----
VMS                  *                  FST2\Administrator  Allow              Full
VMS                  *                  FST2\FST2-HV1$     Allow              Full

PS C:\> Get-SmbOpenFile -SessionId 120259084305

FileId              SessionId          LocalPath            ShareRelativePath  ClientComputerName ClientUserName
-----
120259084441        120259084305     K:\IOBW\            VM2.vhd             \\192.168.150.3    FST2\Administrator
124017181857        120259084305     K:\VMS\VM2.vhd      VM2.vhd             \\192.168.150.3    FST2\Administrator
124017182193        120259084305     K:\VMS\base.VHD     base.VHD            \\192.168.150.3    FST2\Administrator
124017182241        120259084305     K:\VMS\base.VHD     base.VHD            \\192.168.150.3    FST2\Administrator
124017182245        120259084305     K:\VMS\VM2.vhd      VM2.vhd             \\192.168.150.3    FST2\Administrator
121332829441        120259084305     K:\VMS\            VM2.vhd             \\192.168.150.3    FST2\Administrator

PS C:\> Get-SmbServerNetworkInterface

Scope Name          Interface Index    RSS Capable         RDMA Capable        Speed                IPAddress
-----
*                   14                False               False               1000000000          192.168.150.2
*                   13                True                True                32000000000         192.168.151.2
*                   12                False               False               1000000000          10.121.26.65

PS C:\> _
```

“Windows Server 8” Server Manager

The screenshot shows the Windows Server Manager interface for File Services > Pools. The left-hand navigation pane includes SERVERS, POOLS (selected), VOLUMES, SHARES, and ISCSI VIRTUAL DISKS. The main area displays 'Storage Pools | 1 total'. A table lists the storage pool:

Health	Name	Server	Type	Capacity	Free Space	%Used
OK	Spaces and Pools 1.0 on FST2-FS1 (1)					
OK	Pool1	FST2-FS1	Storage Pool	447 GB	44.7 GB	10%

Below the pools table, there are sections for 'VIRTUAL DISKS' and 'PHYSICAL DISKS' for Pool1 on FST2-FS1. The virtual disks table shows:

Health	Name	Status	Redundancy	Provisioning Type	Volume
OK	FST2-FS1 (1)				
OK	Space1	Simple	Fixed		K:

The physical disks table shows:

Health	Name	Status	Capacity	Spindle Speed
OK	PhysicalDisk1	OK	149 GB	0
OK	PhysicalDisk2	OK	149 GB	0
OK	PhysicalDisk3	OK	149 GB	0

The 'New Share Wizard' dialog box is open, showing the 'Specify share name' step. The 'Share name' field contains 'SDC2011'. The 'Share description (Optional)' field contains 'Test share for the SDC 2011 Conference'. The 'Local path to share' field contains 'C:\Shares\SDC2011'. The 'Remote path to share' field contains '\\FST2-FS1\SDC2011'. The 'Share Name' step is selected in the left-hand navigation pane. At the bottom, there are buttons for '< Previous', 'Next >', 'Commit', and 'Cancel'.

The screenshot shows the Windows Server Manager interface for File Services > SHARES. The left-hand navigation pane includes SERVERS, POOLS, VOLUMES, SHARES (selected), and ISCSI VIRTUAL DISKS. The main area displays 'All shares | 2 total'. A table lists the shares:

Share	Local Path	Protocol	Cluster Type
FST2-FS1 (2)			
IOBW	K:\IOBW	SMB	None
VMS	K:\VMS	SMB	None

Below the shares table, there is a 'VOLUME' section for VMS on FST2-FS1. It shows a capacity of 400 GB, with a progress bar indicating 19.4% used (77.6 GB) and 323 GB free space. Data Optimization is set to OFF. At the bottom, there is a 'Go to Volumes >>' link.

SMB2 Performance Counters

SMB2 Server Shares		*\VMSonD		SMB Direct Connection {192.168.201.11:445 -> 192.168.201.21:49152}		SMB2 Client Shares \CONTOSO\HQ-F2\vmsond	
% Durable opens	0.000	% Durable opens	0.000	Bytes RDMA Read/sec	0.000	Avg. Bytes/Read	0.000
% Resilient opens	0.000	% Resilient opens	0.000	Bytes RDMA Written/sec	0.000	Avg. Bytes/Write	0.000
Avg. read response time	0.000	Avg. read response time	0.000	Bytes Received/sec	9,519.629	Avg. Data Bytes/Request	0.000
Avg. write response time		Avg. write response time		Bytes Sent/sec	6,439.749	Avg. Data Queue Length	0.000
Bytes read		Bytes read		Max Bytes/Receive	1,024.000	Avg. Read Queue Length	0.000
Bytes read/sec		Bytes read/sec		Max Bytes/Send	1,024.000	Avg. sec/Data Request	0.000
Bytes Received		Bytes Received		Protocol Version	0x00000100	Avg. sec/Data Request	0.000
Bytes Received/sec		Bytes Received/sec		Receives/sec	69.997	Avg. sec/Read	0.000
Bytes Sent		Bytes Received/sec		Send Credits Granted/sec	69.997	Avg. sec/Write	0.000
Bytes Sent/sec		Bytes Sent		Send Credits Received/sec	69.997	Avg. Write Queue Length	0.000
Bytes Transferred		Bytes Sent/sec		Sends/sec	69.997	Current Data Queue Length	0.000
Bytes Transferred/sec		Bytes Transferred		SMB2 Messages Received/sec	69.997	Data Bytes/sec	0.000
Bytes Written		Bytes Transferred/sec		SMB2 Messages Sent/sec	69.997	Data Requests/sec	0.000
Bytes written/sec		Bytes Written		Stalls (FRMR)/sec	0.000	Metadata Requests/sec	116.153
Files opened/sec		Bytes written/sec	44,172.000	Stalls (Read Queue)/sec		Read Bytes/sec	0.000
Open file count		Files opened/sec	0.000	Stalls (Send Credit)/sec		Read Requests/sec	0.000
Pending Requests		Open file count	6.000	Stalls (Send Queue)/sec		Total Metadata Requests	2,399,310.000
Read Requests Processed		Pending Requests	0.000			Total Read Bytes	41,201,879,168.0000
Read requests processed/sec		Read Requests Processed	16,522.000			Total Read Requests	1,311,574.000
Requests Received		Read requests processed/sec	0.000			Total Write Bytes	8,973,274,548
Requests Received/sec		Requests Received	1,703,277.000			Total Write Requests	282,030.000
Total durable handle reopen count		Requests Received/sec	92.996			Write Bytes/sec	0.000
Total failed durable handle reopen count		Total durable handle reopen count	0.000			Write Requests/sec	0.000
Total failed resilient handle reopen count		Total failed durable handle reopen count	0.000				
Total file open count		Total failed resilient handle reopen count	0.000				
Total resilient handle reopen count		Total file open count	14.000				
Tree Connect Count		Total resilient handle reopen count	1.000				
Write Requests Processed		Tree Connect Count	1.000				
Write requests processed/sec		Write Requests Processed	4,675.000				
		Write requests processed/sec	0.000				

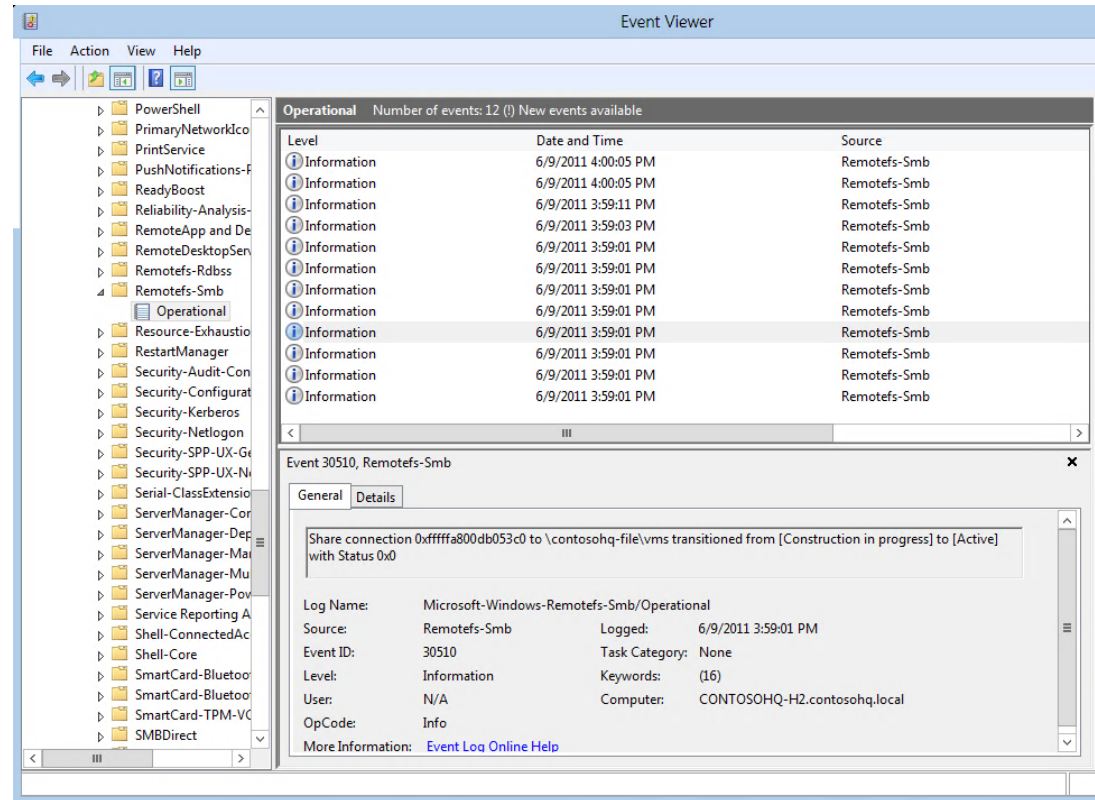
Mapping Local Storage Counters to SMB 2.2 counters

Type	Disk Object Counter	SMB2 Client Shares Counter
IOPS	Disk transfers / sec	Data Requests/sec
	Disk reads / sec	Read Requests/sec
	Disk writes / sec	Write Requests/sec
Latency	Avg disk sec / transfer	Avg. sec/Data Request
	Avg disk sec / read	Avg. sec/Read
	Avg disk sec / write	Avg. sec/Write
IO Size	Avg disk bytes / transfer	Avg. Bytes/Data Request
	Avg disk bytes / read	Avg. Bytes/Read
	Avg disk bytes / write	Avg. Bytes/Write
Throughput	Disk bytes / sec	Data Bytes/sec
	Disk read bytes / sec	Read Bytes/sec
	Disk write bytes / sec	Write Bytes/sec
Queue Length	Avg. Disk Read Queue Length	Avg. Read Queue Length
	Avg. Disk Write Queue Length	Avg. Write Queue Length
	Avg Disk Queue Length	Avg. Data Queue Length

SMB2 Client Shares
 Avg. Bytes/Read
 Avg. Bytes/Write
 Avg. Data Bytes/Request
 Avg. Data Queue Length
 Avg. Read Queue Length
 Avg. sec/Data Request
 Avg. sec/Read
 Avg. sec/Write
 Avg. Write Queue Length
 Current Data Queue Length
 Data Bytes/sec
 Data Requests/sec
 Metadata Requests/sec
 Read Bytes/sec
 Read Requests/sec
 Total Metadata Requests
 Total Read Bytes
 Total Read Requests
 Total Write Bytes
 Total Write Requests
 Write Bytes/sec
 Write Requests/sec

SMB2 Improved Eventing

- ❑ SMB2 Transparent Failover, Witness and SMB2 Multichannel and SMB2 Direct provide detailed events
- ❑ Facilitates troubleshooting by determining the source of errors using Event Viewer
- ❑ Allows performance analysis by understanding bottleneck using tools like XPerf



- ❑ SMB PowerShell cmdlets
- ❑ SMB2 Performance Counters
- ❑ SMB2 Events

Part II

General “Windows Server 8” Advancements in Management

Windows Management Framework (WMF)

- ❑ Robust multi-machine management platform and tools surfaced through PowerShell

- ❑ High value ecosystem now and future
 - ❑ Minimizes cost to develop solutions and maximizes value of manageability
 - ❑ Breaks the chicken/egg problem by delivering immediate value to ITPros
 - ❑ Transforms NxM into and NxI problem (everyone supplies and consumes PS)
 - ❑ Available down-level (Windows 7, Windows Server 2008 and 2008 R2)

- ❑ Investments Themes
 - ❑ Coverage [CEC, PSCIM, REST and Web Cmdlets]
 - ❑ Broad adoption [Simplification, better APIs, PSCIM, ODATA]
 - ❑ Robust, multi-machine [WF, WS-Man]
 - ❑ Standards [CIM, WS-Man, Community Promise PowerShell]
 - ❑ Services friendly [POWWA, ODATA, Automation]

Why PowerShell?

- ❑ “Automate or Drown!”
 - ❑ Going to an N machine world
 - ❑ Enables review and auditing of intent and actions
 - ❑ Reduces costs while increasing agility and quality
- ❑ Main attributes
 - ❑ High-level, task-oriented abstractions
 - ❑ Strong naming guidelines and production semantics
 - ❑ Learn once, manage everything
 - ❑ Produces a safe “think, type, get” world
 - ❑ (PowerShell –eq “No GUI”) | Very important for management of datacenter servers with minimal footprint
- ❑ Architected to maximize ecosystem value
 - ❑ Wide applicability: IT Pros, Developers, System Integrators
 - ❑ Investments amplify

Windows 7 Mgmt Client API

- ❑ Two management client APIs
 - ❑ WMI and WinRM
- ❑ Win7 WMI client API
 - ❑ Works only over DCOM, thus not heterogeneous friendly
 - ❑ Remains in “Windows Server 8” for compatibility
- ❑ WinRM
 - ❑ Windows 7 standards based client API is XML based.
 - ❑ Client needs to deal with XML documents and not with actual object instances.
 - ❑ Does not support all operations (Indications, create and delete)

New WMI MI API

- ❑ New Management Infrastructure MI API
- ❑ Simplified client and provider programming model
 - ❑ Simple session based operations
 - ❑ Improved Async programming.
 - ❑ Returns rich objects, not XML
 - ❑ Generic *-Cim* cmdlets
- ❑ Works with WinRM and DCOM
 - ❑ The default behavior is WinRM
 - ❑ Works down level over DCOM
 - ❑ Works with Windows and Non-Windows standard compliant CIMOMs
- ❑ Rich PowerShell semantics
 - ❑ Verbose, WhatIf, Confirm, etc...

Supported By Standards

- ❑ Latest CIM schema support
 - ❑ Support for CIM v2.25, CIM_Indications and CIM_Error
- ❑ Rich PowerShell Semantics
 - ❑ WMI method invocations
 - ❑ New CIM cmdlets leveraging the new rich API set
- ❑ Full WS-Man support
 - ❑ Integration with new MI client APIs
 - ❑ Subscription to WMI event indications
 - ❑ Retrieve CIM metadata (class schema & hierarchy)
 - ❑ Streaming output parameters for WMI methods

Windows Management Framework

Client

Server Manager,
PowerShell, ISE

Admin GUIs and tools,
IDEs

Remote Tools

Clients and
Servers

WSMAN

BITS

ODATA

HTTP/HTTPS

Protocols

WMI ↔ PowerShell ↔ Workflow

Engines

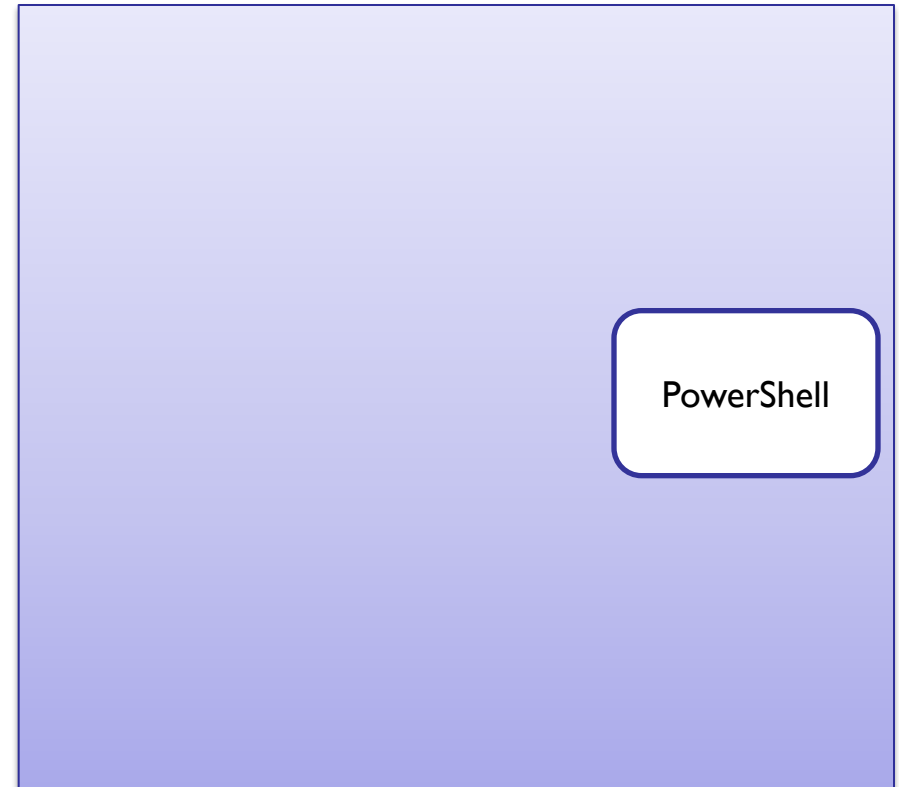
SMI-S, COM, .NET, RPC, DCOM, HTTP ...

Access API
(Local/Remote)

Process, Disk, Pool, Volume, Share, ...

Managed Elements
(Local/Remote)

The WMF technology stack



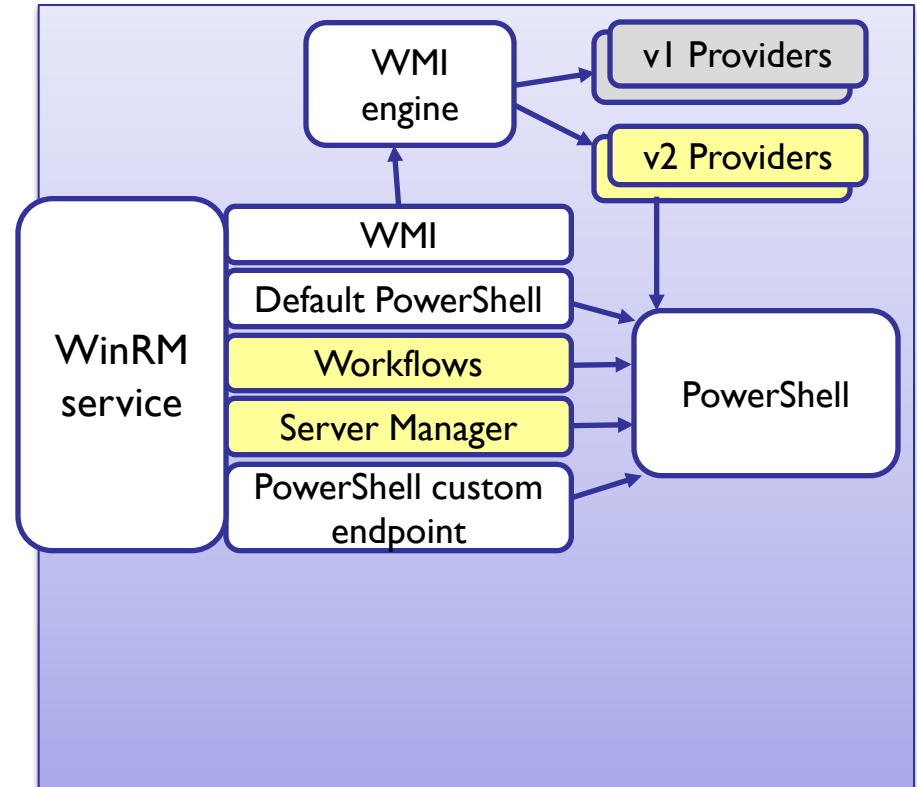
Unmodified

New in "Windows 8"

Updated in "Windows 8"

New for Non-Windows

The WMF technology stack



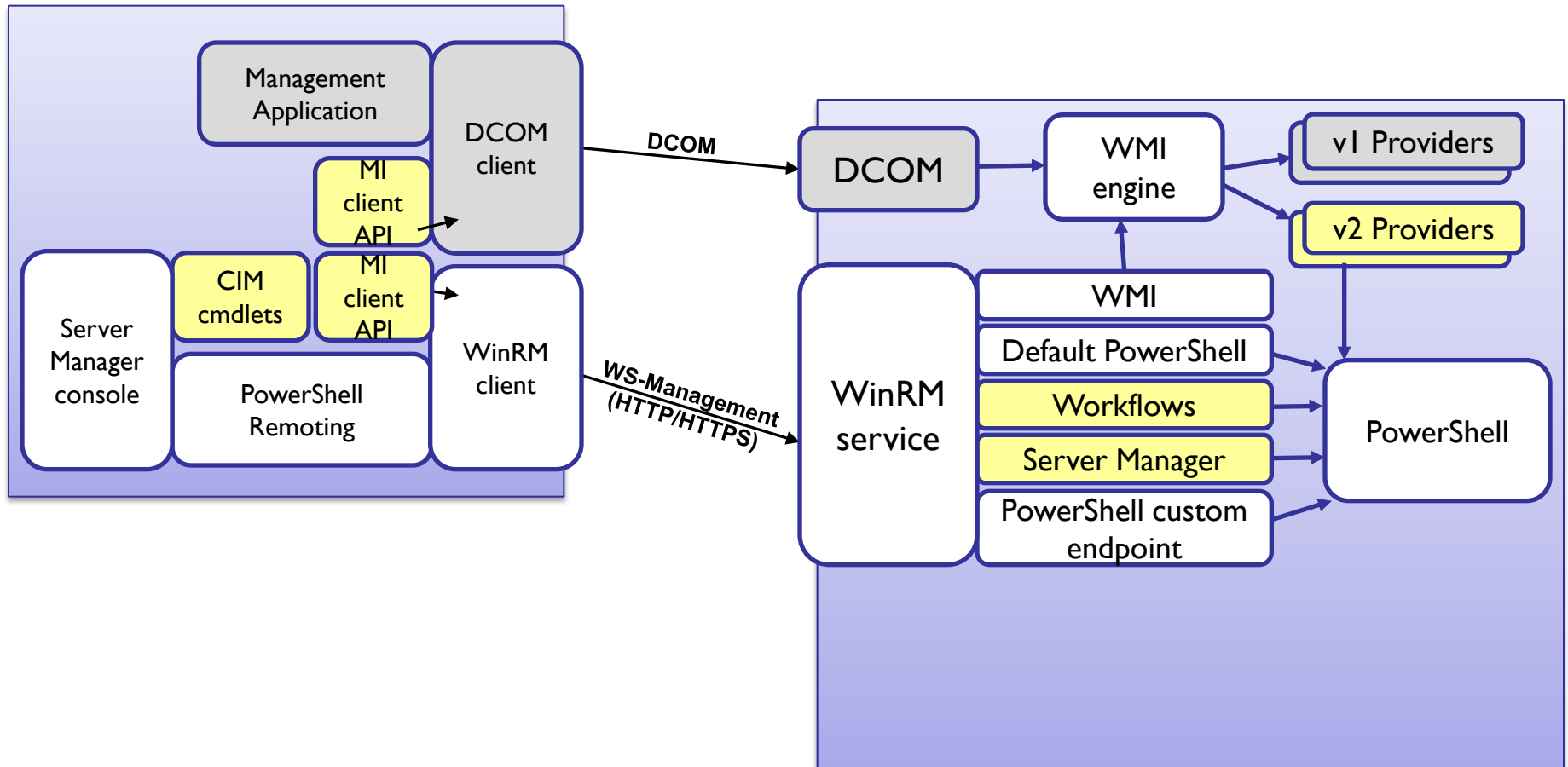
Unmodified

New in "Windows 8"

Updated in "Windows 8"

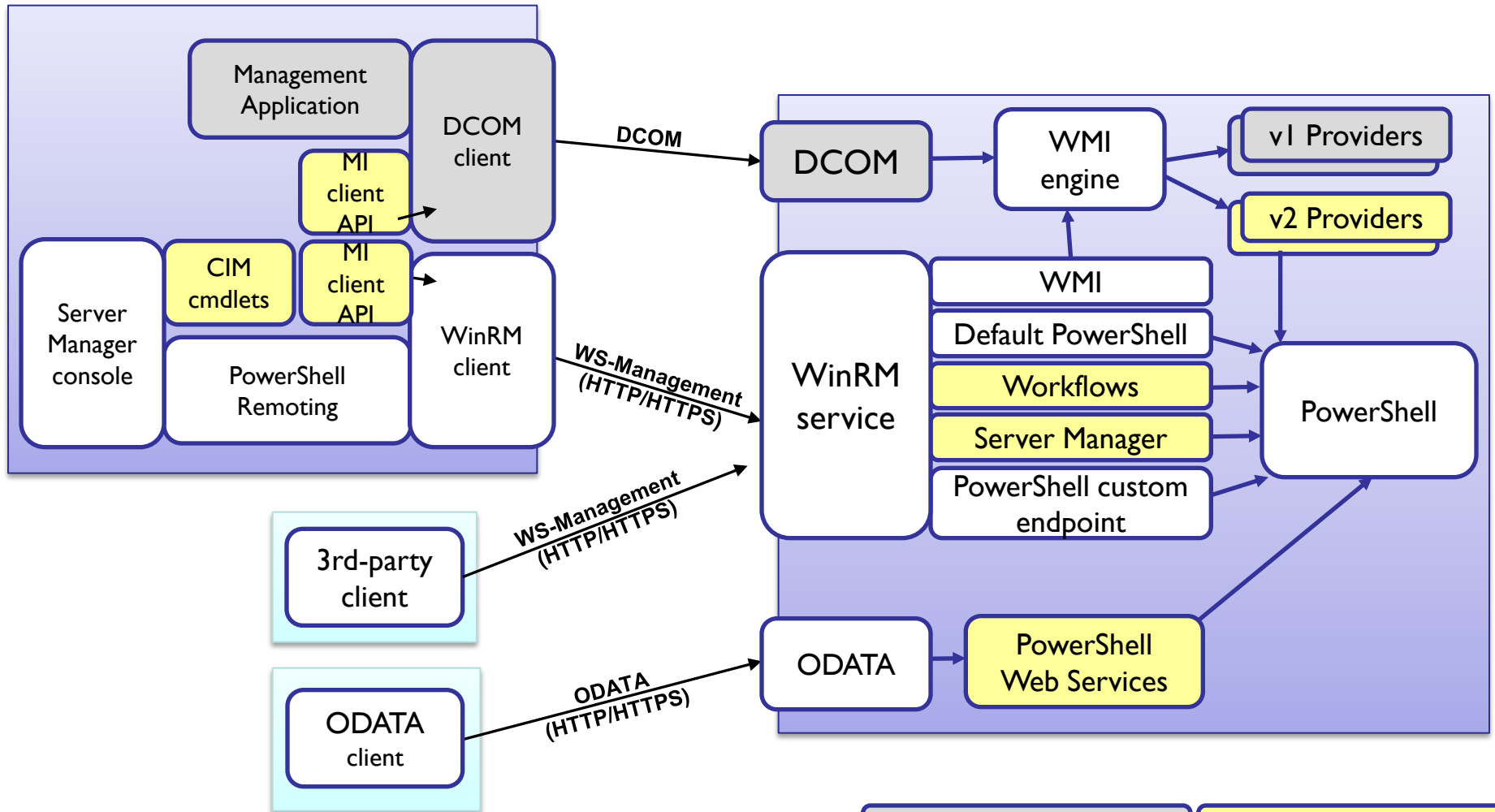
New for Non-Windows

The WMF technology stack



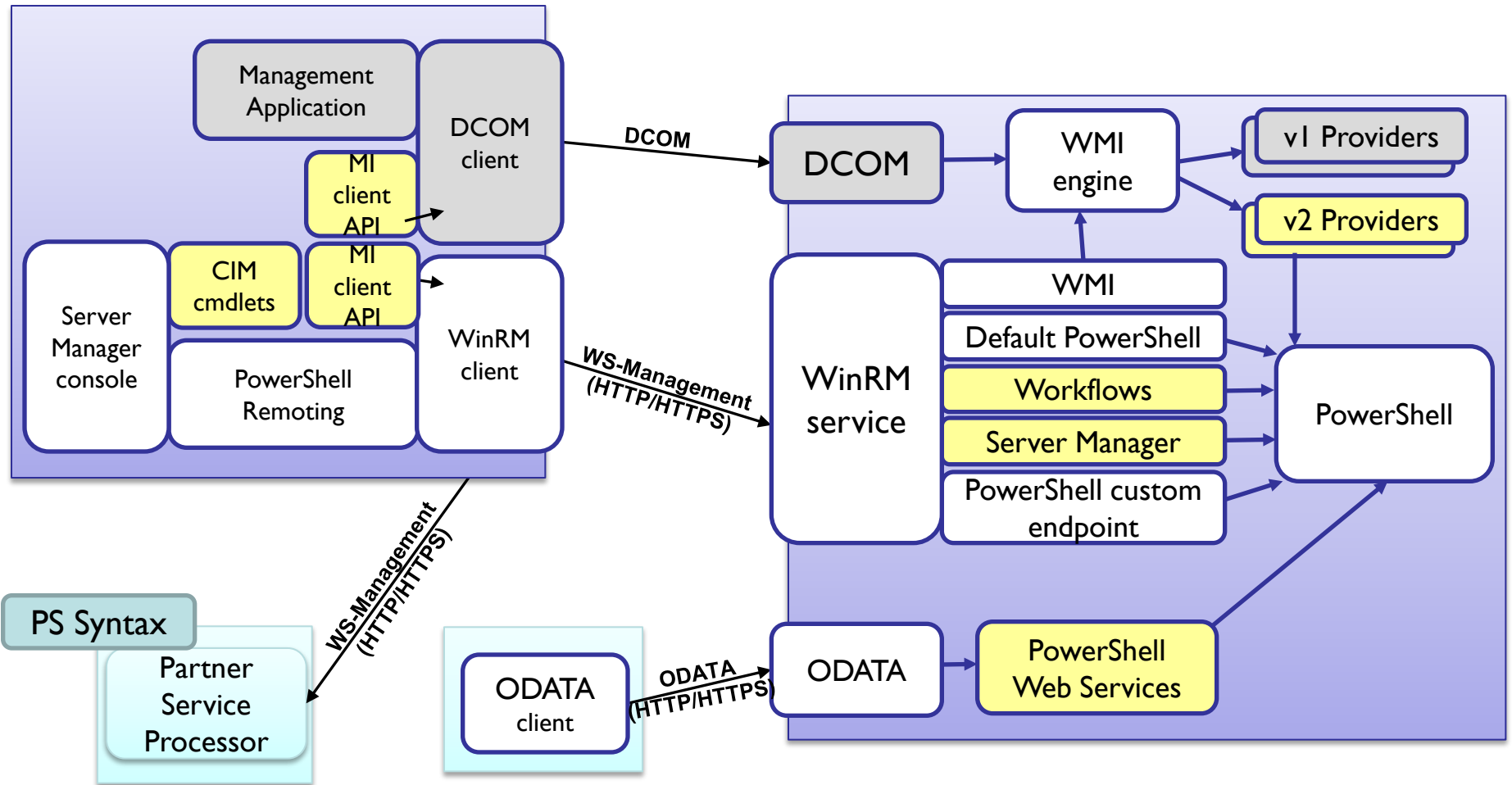
Unmodified	New in "Windows 8"
Updated in "Windows 8"	New for Non-Windows

The WMF technology stack



Unmodified	New in "Windows 8"
Updated in "Windows 8"	New for Non-Windows

The WMF technology stack



Unmodified	New in "Windows 8"
Updated in "Windows 8"	New for Non-Windows

- ❑ PowerShell v3 cmdlets
- ❑ “Windows Server 8” Server Manager

Part III

Call to Action

Manageability Call to Action

- ❑ Non-Windows Devices (i.e., SMB 2.2 implementer)
 - ❑ Support CIM and WS-MAN standards in your devices
 - ❑ New APIs and PowerShell Cmdlets allow developers and IT Pros to manage your device
 - ❑ Implement our CIM/WS-MAN extensions
 - ❑ PowerShell will generate high-level task-oriented PowerShell Cmdlets for free
 - ❑ Make use of NanoWBEM (small foot print cross platform CIMOM + WsMan toolkit)
 - ❑ Remember: SMB SRV service/Net APIs do not expose the new SMB 2.2 functionality

- ❑ Windows Devices
 - ❑ Implement CIM standard schema
 - ❑ Use simplified WMI V2 provider model and tools
 - ❑ Use our guidelines and our library APIs to get PowerShell Cmdlets for free [PSCIM]
 - ❑ We light you up with maximum manageability: existing and new tools and protocols

Management Tools Call to Action

- ❑ Install and run Management Tools on Client to manage remote Servers
- ❑ Layer GUI over PowerShell
 - ❑ Ensures everything is automatable and uses the GUI to teach the CLI
- ❑ Consider configuring your own PowerShell Endpoint
 - ❑ Allows delegated administration
 - ❑ Robust multi-machine management via workflow
- ❑ Consider configuring a PowerShell OData Endpoint
 - ❑ Easy management for Web GUIs or non-Windows tools
- ❑ Use new native/managed API to talk to remote CIM-only devices
 - ❑ Simplified access to WSMAN and DCOM remote CIM devices

QA

Thanks!

❑ NetFile Functions

- ❑ NetFileClose
- ❑ NetFileEnum
- ❑ NetFileGetInfo

❑ Session Functions

- ❑ NetSessionDel
- ❑ NetSessionEnum
- ❑ NetSessionGetInfo

❑ Statistics Function

- ❑ NetStatisticsGet

❑ Network Share Functions

- ❑ NetConnectionEnum
- ❑ NetShareAdd
- ❑ NetShareCheck
- ❑ NetShareDel
- ❑ NetShareDelEx
- ❑ NetShareEnum
- ❑ NetShareGetInfo
- ❑ NetShareSetInfo

[http://msdn.microsoft.com/en-us/library/bb525393\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb525393(VS.85).aspx)

```
class Win32_Share :  
CIM_LogicalElement  
{  
    uint32    AccessMask;  
    boolean   AllowMaximum;  
    string    Caption;  
    string    Description;  
    datetime  InstallDate;  
    uint32    MaximumAllowed;  
    string    Name;  
    string    Path;  
    string    Status;  
    uint32    Type;  
};
```

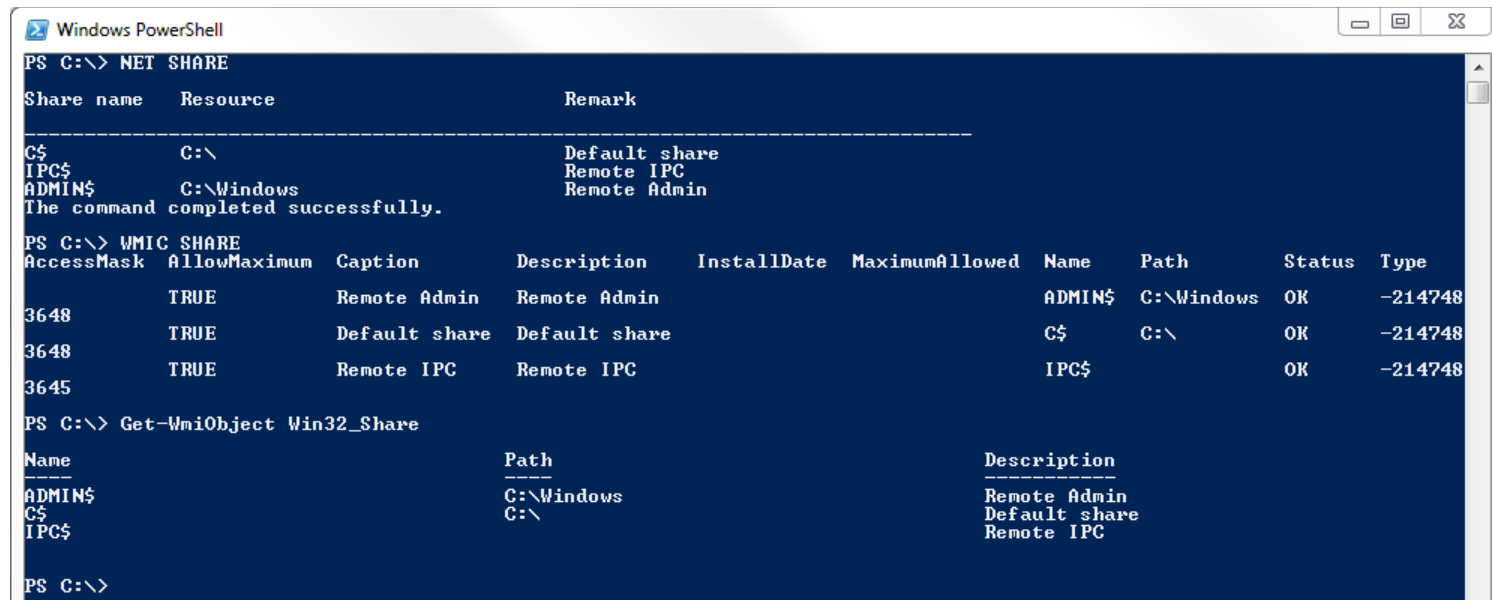
□ Win32_Share Methods

- Create
- Delete
- GetAccessMask
- SetShareInfo

[http://msdn.microsoft.com/en-us/library/aa394435\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa394435(VS.85).aspx)

WS 2008 R2 Command lines

- ❑ Command lines
 - ❑ NET SHARE
 - ❑ WMIC SHARE
 - ❑ PowerShell: Get-WMIObject Win32_Share



```
Windows PowerShell
PS C:\> NET SHARE
Share name      Resource          Remark
-----
C$              C:\              Default share
IPC$           C:\Windows       Remote IPC
ADMIN$         C:\Windows       Remote Admin
The command completed successfully.

PS C:\> WMIC SHARE
AccessMask  AllowMaximum  Caption      Description  InstallDate  MaximumAllowed  Name  Path      Status  Type
-----
3648        TRUE         Remote Admin Remote Admin          ADMIN$  C:\Windows  OK    -214748
3648        TRUE         Default share Default share          C$     C:\         OK    -214748
3645        TRUE         Remote IPC   Remote IPC            IPC$   C:\         OK    -214748

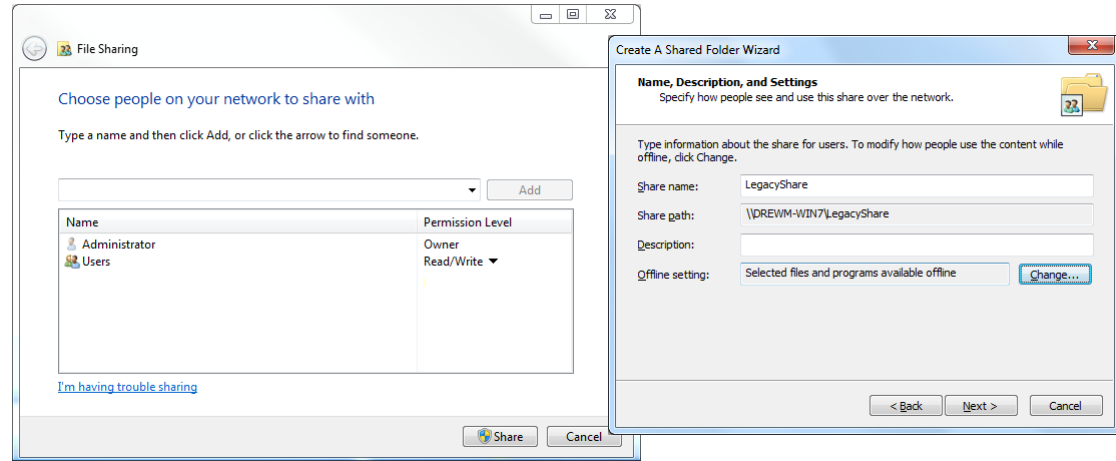
PS C:\> Get-WmiObject Win32_Share
Name              Path              Description
-----
ADMIN$           C:\Windows       Remote Admin
C$              C:\              Default share
IPC$           C:\              Remote IPC

PS C:\>
```

Windows 7 and WS 2008 R2 GUIs

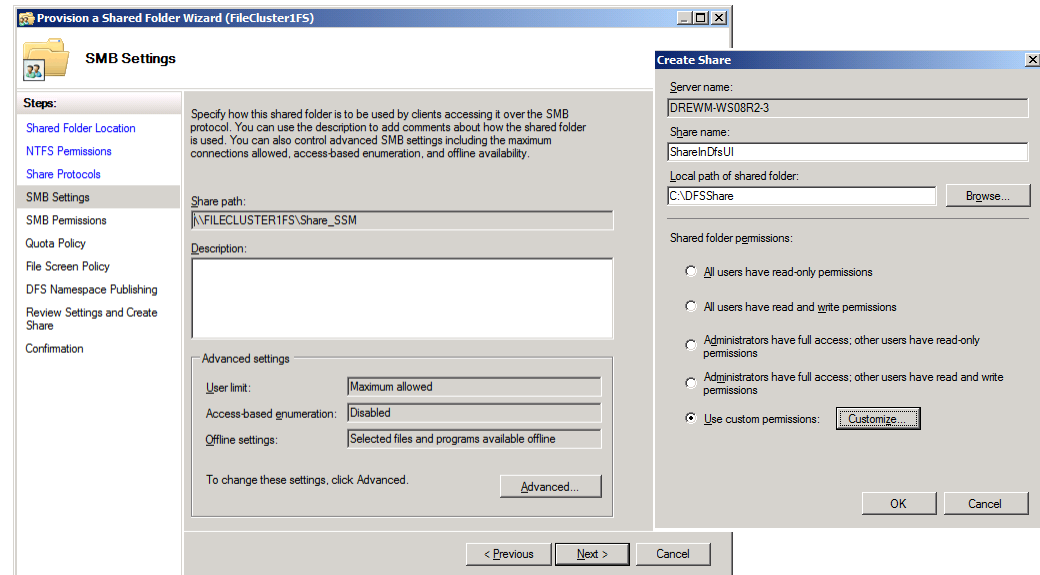
Focused on end users

- ❑ Windows Explorer
- ❑ Computer Management, File Shares

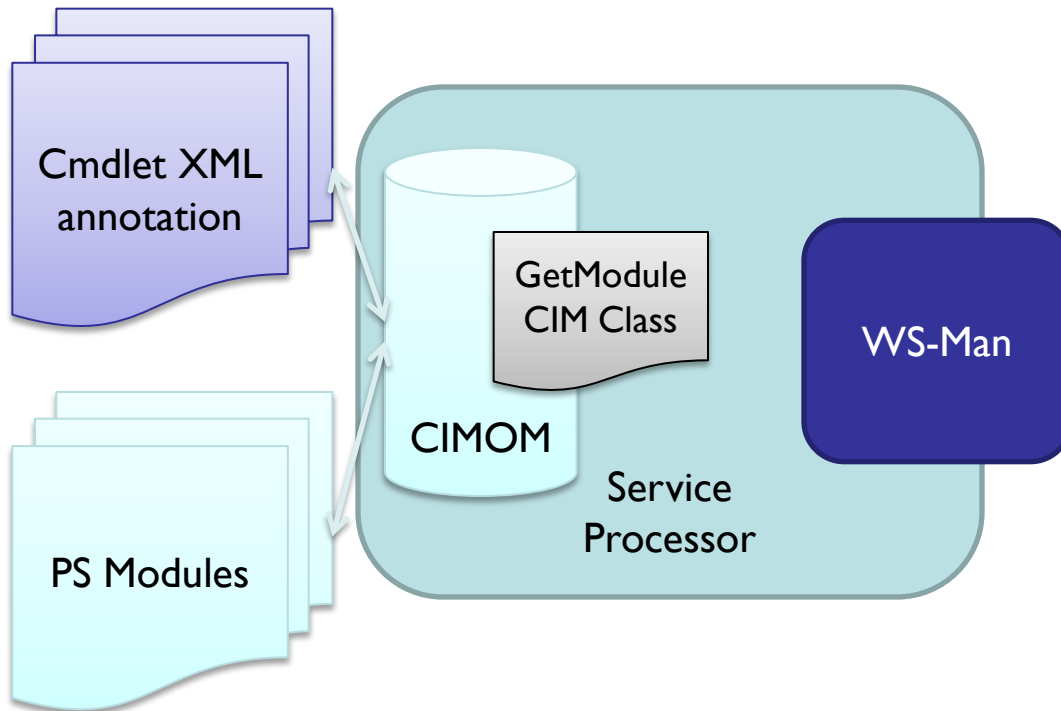


Focused on IT Admins

- ❑ Share and Storage Management
- ❑ Cluster Management
- ❑ DFS Management



Plug into the EcoSystem



- ❑ Annotate your CIM classes with the Cmdlet XML metadata
- ❑ Add a new CIM class in your service processor that enumerates and returns the registered Modules in your CIMOM.
- ❑ IT Pros start PowerShell on a “Windows 8” client.
- ❑ IT Pros Discover the modules on the service processor using Get-Module.
- ❑ IT Pro select which Module to import and import the Cmdlet XML metadata using Import-Module.
- ❑ IT Pro is now empowered to manage the Service processor

“Windows Server 8” Storage Management Architecture

