

Realtime
publishers

Tips and Tricks
Guide™ To

Windows
Administration

Don Jones and
Dan Sullivan

Tip, Trick, Technique 8: Remote Server Manager in R2	1
Tip, Trick, Technique 9: Leveraging Server Core in R2.....	1
Tip, Trick, Technique 10: Deleted AD Object Recovery in R2.....	4
Tip, Trick, Technique 11: Classifying Files in R2	7
Tip, Trick, Technique 12: Remote Command-Line Administration in R2.....	12
Download Additional eBooks from Realtime Nexus!.....	12

Copyright Statement

© 2009 Realtime Publishers. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers or its web site sponsors. In no event shall Realtime Publishers or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at info@realtimepublishers.com.

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library for IT Professionals. All leading technology eBooks and guides from Realtime Publishers can be found at <http://nexus.realtimepublishers.com>.]

Tip, Trick, Technique 8: Remote Server Manager in R2

Server Manager has proven to be a great way of administering Win2008's complex set of server roles and features. It offers a central means of adding, configuring, and removing roles and features, and provides central access to a number of security- and configuration-related features that would otherwise be scattered across the operating system (OS) and require a lot of digging. If Server Manager had one significant failing, though, it was its inability to work with remote computers. If you wanted to use Server Manager, you were stuck logging onto the server console directly—which is a real limitation and really breaks the “single-seat administration” model Microsoft has been slowly trying to implement.

In Windows Server 2008 R2 (“R2” for short), though, Server Manager has been improved to support remote management. As Figure 16 shows, this change is subtle and one that's easy to miss: You simply pick up a “Connect to Computer” menu option.

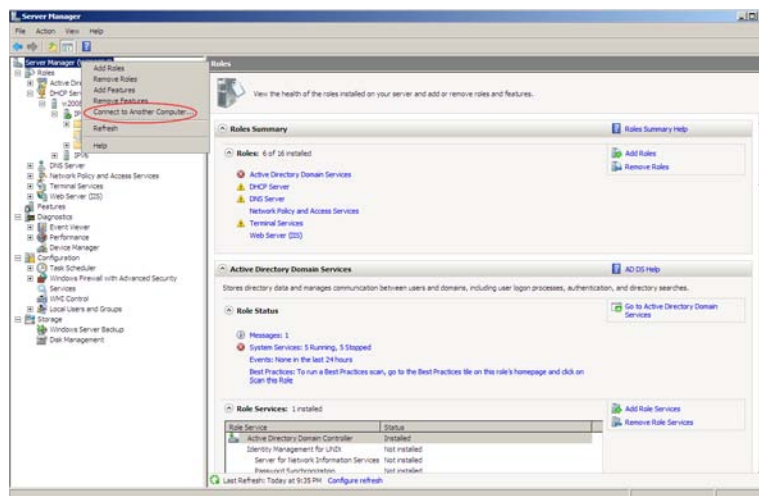


Figure 16: Connecting to a remote computer.

This feature means you can now use a local copy of Server Manager to manage features and roles on all your R2 servers—except those running Server Core; unfortunately, the Server Manager console can't install roles on the stripped-down Server Core version of the OS. Hopefully that capability will come in time, as it would go a long way toward making Server Core more approachable for a wider range of administrators.

Tip, Trick, Technique 9: Leveraging Server Core in R2

R2 offers an improved version of Server Core that makes up for a lot of the shortcomings of previous versions, albeit at a potentially higher level of maintenance overhead. One of the most important new features is the SConfig.exe utility (see Figure 17). This utility offers a text-based menu that helps administrators configure the core operating system (OS)

settings such as domain membership, computer name, Windows Update, network settings, and so forth. This is a welcome improvement, as many of these tasks in the past required complex, fairly arcane command-line tools. Those same tools are still in use; they're just called in the background by SConfig. Think of SConfig as a sort of lightweight "Server Manager" specifically for Server Core.

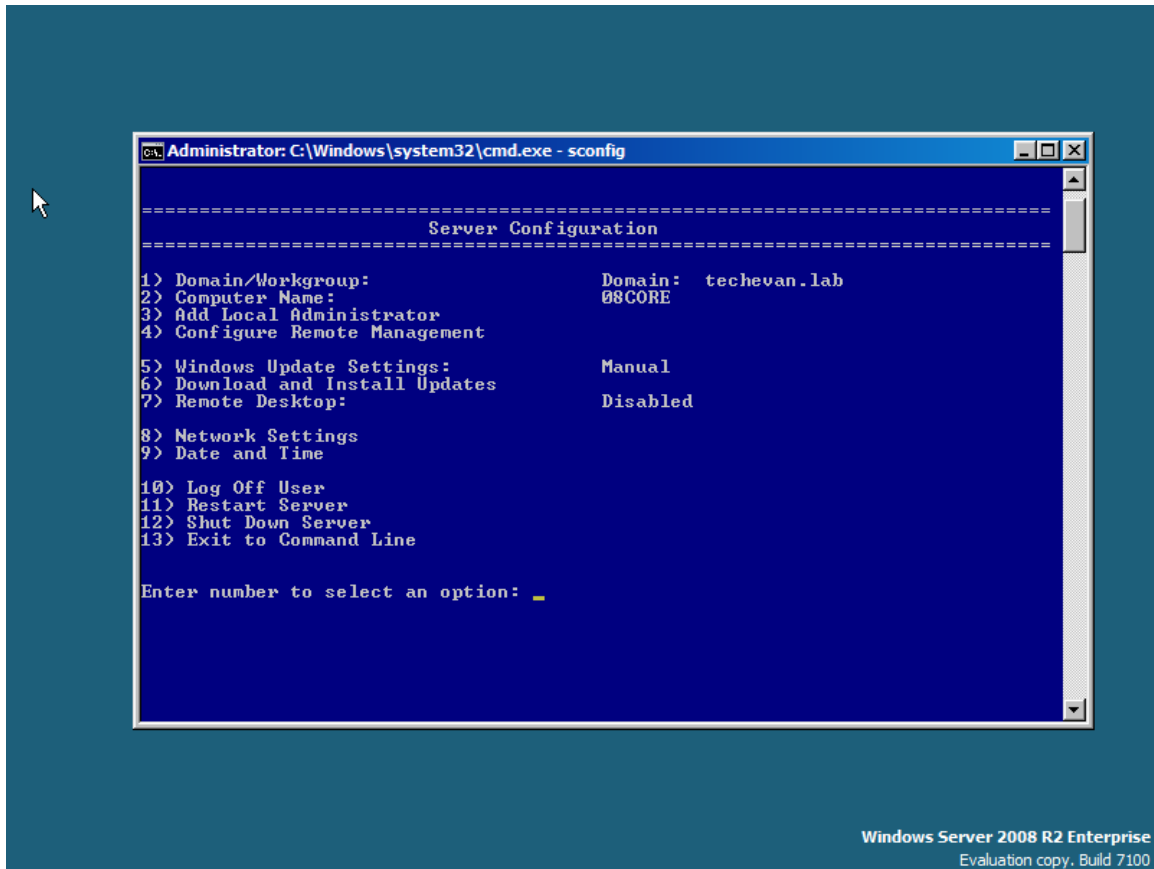


Figure 17: Using SConfig in Server Core R2.

Server Core also offers a subset of the .NET Framework. This subset includes portions of v2.0 and v3.0; it specifically excludes the Windows Forms classes and Windows Presentation Framework, which require graphical user interface (GUI) elements not present in Server Core. The inclusion of this Framework subset has a couple of really important, far-reaching consequences. One of those is the potential for additional patches, as the Framework is an additional set of “moving parts” that do come with their own potential problems and the resulting hotfixes and service packs. A major benefit of Server Core has always been that it requires fewer patches—historically, about a third of what the full Windows OS requires. The Framework isn’t historically a heavily-patched set of code but it *does* get patched.

The tradeoff, however, is significant: Server Core R2 now supports ASP.NET Web applications under IIS 7.5, which is a major improvement over the original Server Core release—which didn’t have any Framework and didn’t support ASP.NET at all. The inclusion of the Framework in Server Core R2 also permits remote management of IIS through the standard IIS management console—another major benefit for administrators (you have to enable the remote management service to make this happen).

Perhaps the biggest improvement offered by the Framework subset, however, is the inclusion of Windows PowerShell v2 as a pre-installed component of Server Core R2. This addition brings significant new administrative capability to Server Core, including the ability to remotely connect to Server Core’s PowerShell instances *from remote machines*, enabling remote command-line management of single and multiple servers.

Cross Reference

See Tip, Trick, Technique 12: Remote Command-Line Administration in R2 for more details on PowerShell v2’s remote management capabilities.

Active Directory Certificate Services (ADCS, formerly just Certificate Services) is also supported as a server role on Server Core R2. This means that yet another key infrastructure component—Public Key Infrastructure (PKI)—can now be migrated to this lower-maintenance, smaller-footprint OS.

Keeping in mind that R2 is only being made available in a 64-bit edition, Server Core R2 *optionally* supports a WoW64 layer that makes it possible to run 32-bit applications. I primarily see this as being used to support older management agents or anti-malware applications, although every effort should be made to acquire native 64-bit versions of these items as quickly as possible.

Finally, Server Core R2 also supports File Server Resource Manager (FSRM), which finally enables advanced file quotas and other FSRM-related functionality in Server Core.

Tip, Trick, Technique 10: Deleted AD Object Recovery in R2

Much has been made about the “Active Directory Recycle Bin” in Windows Server 2008 R2, but the reality falls somewhat short of the hype. Although this feature provides great capabilities, it also has some limitations that aren’t immediately obvious—and the term “Recycle Bin” actually implies a level of functionality and ease of access that simply isn’t present. But first, some background.

As you may know, deleted objects in Active Directory (AD) aren’t deleted immediately. Instead, they’re marked with a “tombstone” flag, which is replicated to all domain controllers in the domain. Tombstoned objects, as they’re called, continue to hang around in the directory for some time—180 days in the most recent versions of AD. Although they can’t be used to log on or for any other purposes, keeping the objects around in this tombstoned condition helps ensure that *every* domain controller knows about the deletion.

Some third-party Recycle Bin-like tools of the past simply take advantage of the situation, giving you a graphical user interface (GUI) for seeing tombstoned objects, and enabling you to remove the tombstone flag (and replicate that change), bringing the object back to life—*reanimating* it, to stick with the graveyard terminology. Some third-party recovery tools provide no other functionality, in fact, especially those of the shareware variety, and you don’t even *need* a tool if you’re comfortable using ADSIEdit or other free, low-level tools that enable you to change the tombstone attribute yourself.

There’s a downside, though: When an object is deleted, AD removes most of its attributes at the same time it applies the tombstone flag. That means many of the object’s attributes are no longer available, so the object isn’t “complete.” This is especially frustrating with user objects, as we tend to populate many of the users’ attributes. So simply reanimating an object often isn’t that “simple” at all because you may also need to re-populate the majority of its attributes to make it fully functional again.

Windows Server 2008 R2 makes one important change to the deleting process: It places deleted objects into a “recycled” state where their attributes are left intact. Thus, reanimating them, by flipping the tombstone flag, is easier, because the object is preserved in its original form.

Unfortunately, Windows Server 2008 R2 *will not provide an actual Recycle Bin* in the form of an icon or container that you can use to easily access deleted objects. Deleted objects will still be essentially inaccessible from most native AD management tools, and you’ll need to use low-level directory editors, scripting, or other—frankly complex—means to reanimate objects from their “recycled” state. The term “Recycle Bin” is kind of misleading, because although the feature does provide a sort of “undo” capability, it doesn’t do so in the same easy-to-access way that the Windows Explorer Recycle Bin does.

Also, this new “recycled” state depends on changes made to AD in Windows Server 2008 R2—meaning you can’t leverage this new feature until *every domain controller* has been upgraded to this new version of Windows. You also have to upgrade *every domain* in your environment to the Windows Server 2008 R2 functional level, and upgrade your forest to the Windows Server 2008 R2 functional level. That’s a serious commitment for most organizations, requiring planning, new software licenses, and a significant amount of effort in order to reduce the risk of outages in a production environment. Figure 18 shows how to make the upgrade using the new Windows PowerShell AD cmdlets included in R2.

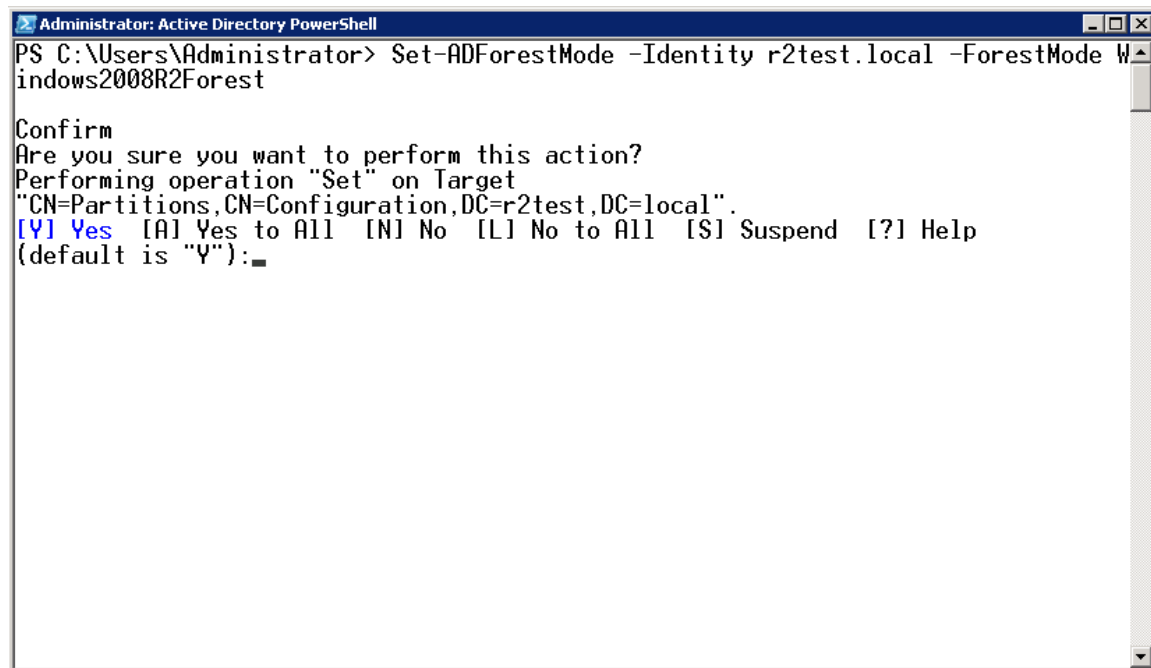
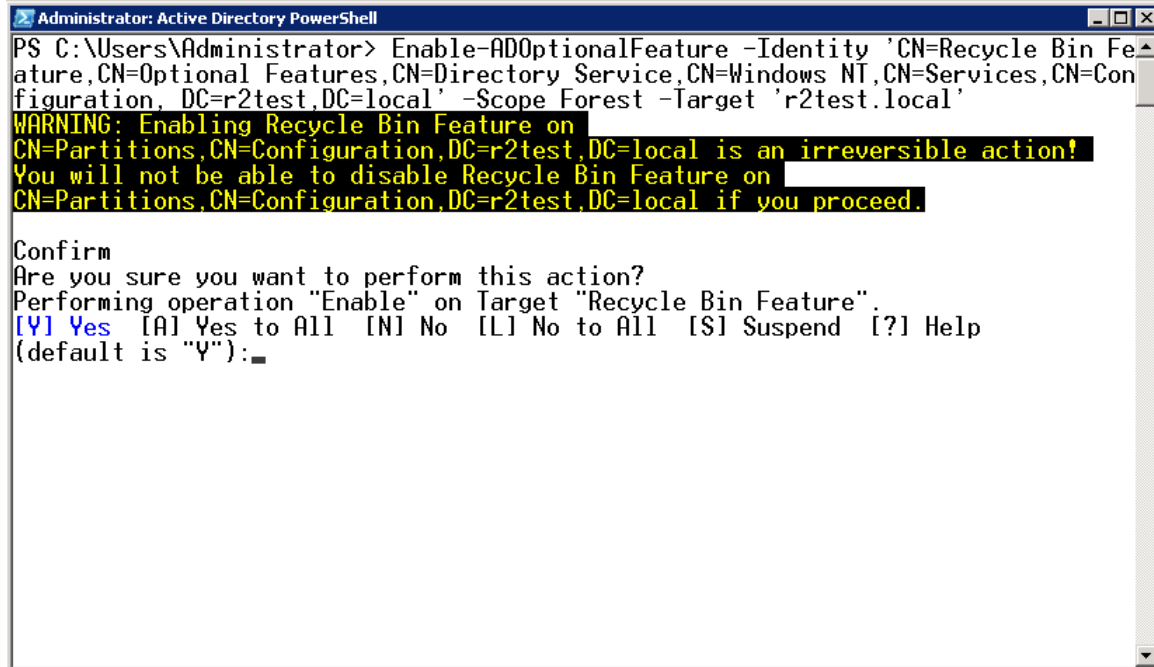


Figure 18: Upgrading the forest functional level.

But wait, there’s more to do: Once your domain controllers, domains, and forests are upgraded, you have to manually enable the “Recycle Bin” functionality in AD. Figure 19 shows this being done from Windows PowerShell.



```
Administrator: Active Directory PowerShell
PS C:\Users\Administrator> Enable-ADOptionalFeature -Identity 'CN=Recycle Bin Feature,CN=Optional Features,CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=r2test,DC=local' -Scope Forest -Target 'r2test.local'
WARNING: Enabling Recycle Bin Feature on CN=Partitions,CN=Configuration,DC=r2test,DC=local is an irreversible action!
You will not be able to disable Recycle Bin Feature on CN=Partitions,CN=Configuration,DC=r2test,DC=local if you proceed.

Confirm
Are you sure you want to perform this action?
Performing operation "Enable" on Target "Recycle Bin Feature".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help
(default is "Y"):
```

Figure 19: Enabling the Recycle Bin.

Once you've done that, you can start writing scripts that actually let you recover deleted objects with their attributes intact. Oh, and once the "Recycle Bin" functionality is turned on, you can't turn it off. So before enabling it, make *absolutely certain* that this new feature won't be in violation of any internal security rules, legislative security requirements, or industry security requirements. For example, in many European countries, it's illegal to retain personally-identifiable information (PII) in certain circumstances; enabling the "Recycle Bin" may unacceptably retain PII without you realizing it, as object attributes aren't deleted.

Accessing deleted objects isn't as simple as opening a "Recycle Bin" icon in the AD management console; far from it. You'll need a lower-level tool, like Ldp.exe, to access the newly-created Deleted Objects container, as shown in Figure 20.

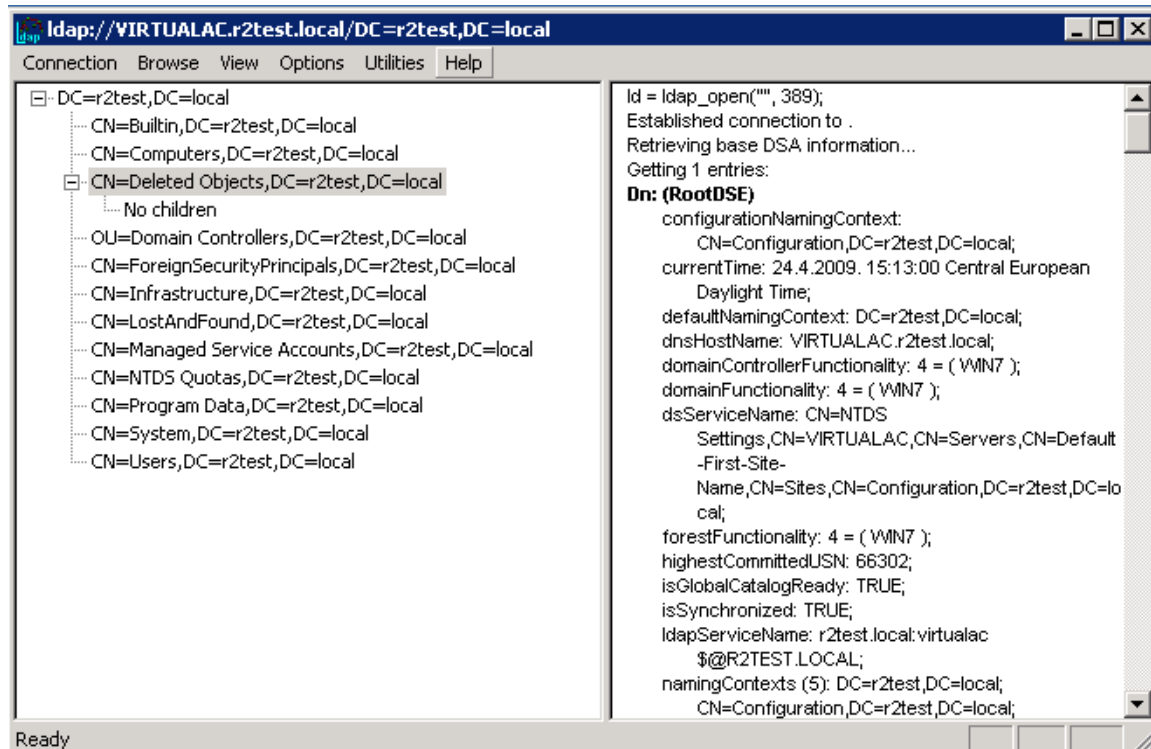


Figure 20: Accessing deleted objects in Ldp.exe.

The Recycle Bin is also only useful for deleted objects: *Changes* to objects aren't captured and preserved. Restoring multiple objects, especially those in a deep hierarchy, is still complicated. Non-directory objects, including Group Policy Objects (GPOs, which live on the file system, not in the directory) aren't protected by the Recycle Bin. The Recycle Bin also relies on AD itself being functional; if something goes wrong at the domain or forest level, you'll still need to have a backup made by other means.

So the new Recycle Bin feature can certainly be useful—but you need to understand its limitations before you rely on it, and you may still want to have third-party recovery tools in place for other scenarios and for ease of use. You'll certainly still want regular domain controller backups.

Tip, Trick, Technique 11: Classifying Files in R2

An entirely-new feature in Windows Server 2008 R2 is the Windows File Classification Infrastructure (FCI). This feature is designed to help administrators better manage file storage resources, enforce company policies regarding stored data, and so on. FCI is essentially designed to help classify the data on your file servers and to automate otherwise-manual processes using predefined policies that are based on the business value of your data. FCI is an *infrastructure* feature, meaning it provides a lot of ways for third-party vendors to “hook in” and provide features above and beyond what Windows includes natively.

Here's the basic problem FCI seeks to solve: Organizations would love to be able to clean up their file servers. But some data needs to be preserved for long periods of time, and today it's very difficult and time consuming to sort the "keeper" data from the "don't need it" data. FCI is designed to support predefined rules that help Windows automatically classify data, and then allow management processes—such as file cleanup and archiving, or security audits—to operate from the classifications.

Natively, R2's FCI helps classify files based on content and location. One classified, sensitive data might be moved or secured differently, backup solutions might prioritize highly-valuable files over less-valuable ones within a backup window, or stale data might be automatically archived or deleted.

The native FCI capabilities are accessed through the File Server Resource Manager (FSRM) console, shown in Figure 21.

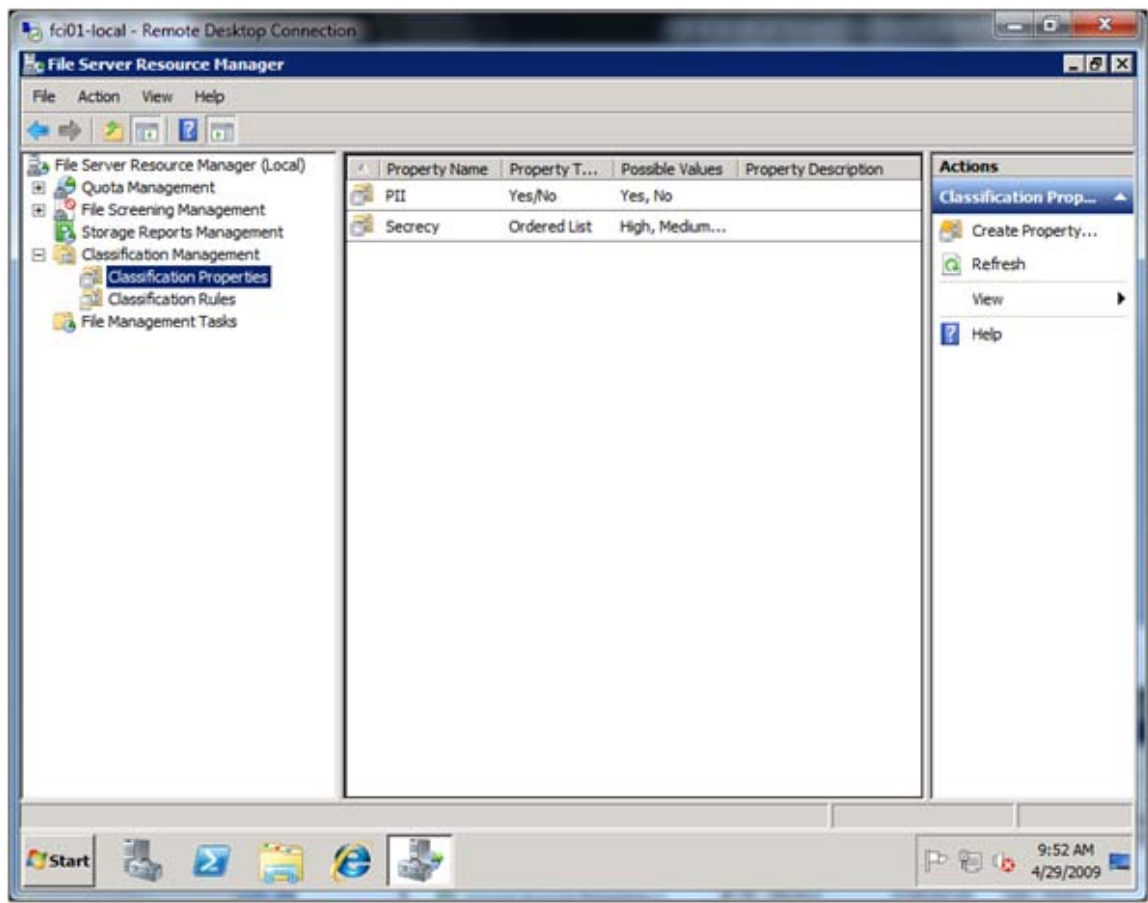


Figure 21: Accessing FCI through FSRM.

As you can see, classification starts with a list of classification properties. In this example, files can be classified as having personally-identifiable information (PII) or not, and can have a “secrecy” level applied. These properties essentially define the key aspects of information that might drive a business to make different decisions about the file: Files containing PII might be secured differently, or files with a high “secrecy” level might be backed up more frequently.

Next, rules are created to help automatically populate these properties for each file. Figure 22 shows the creation of a rule, where files in a particular location have a specific secrecy level applied automatically.

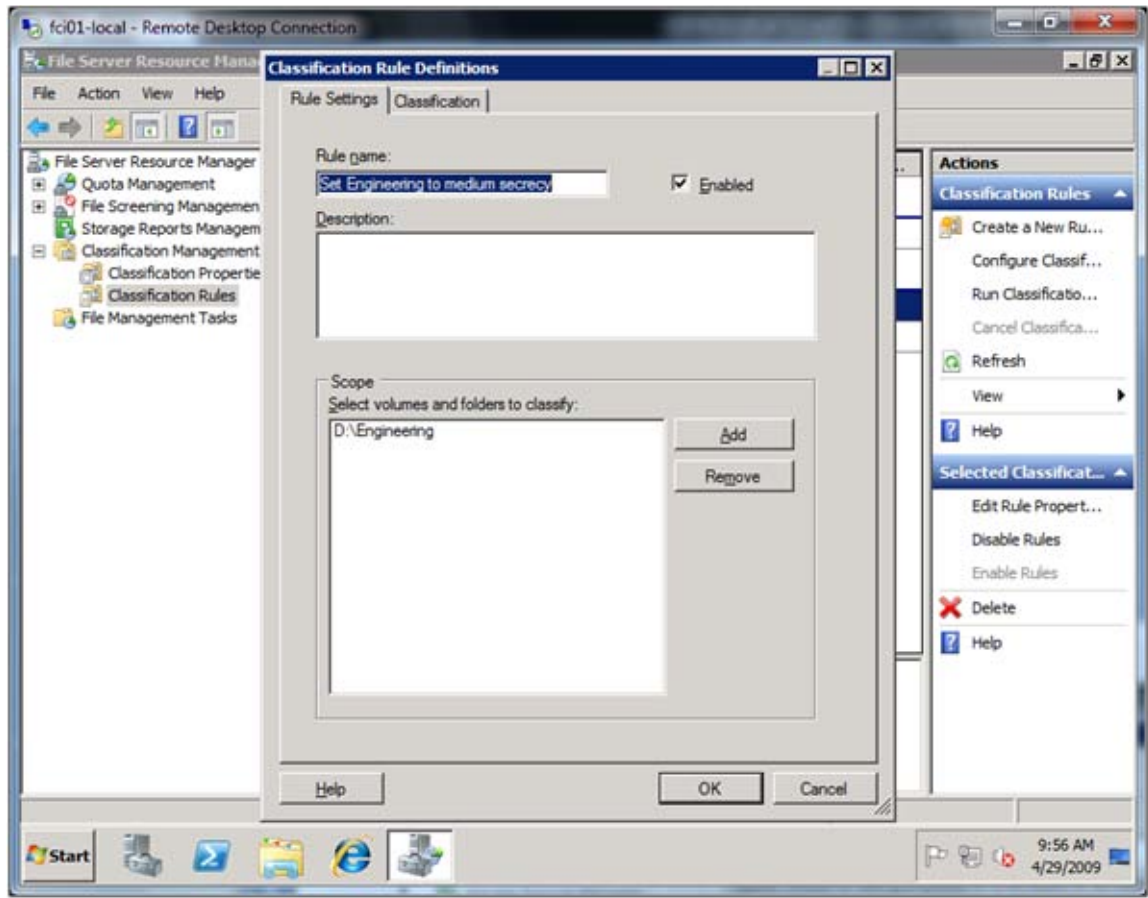


Figure 22: Automatic classification rules.

The content of files, rather than just their location, can also drive the classification. Figure 23 shows the Content Classifier being used to set the “PII” classification property.

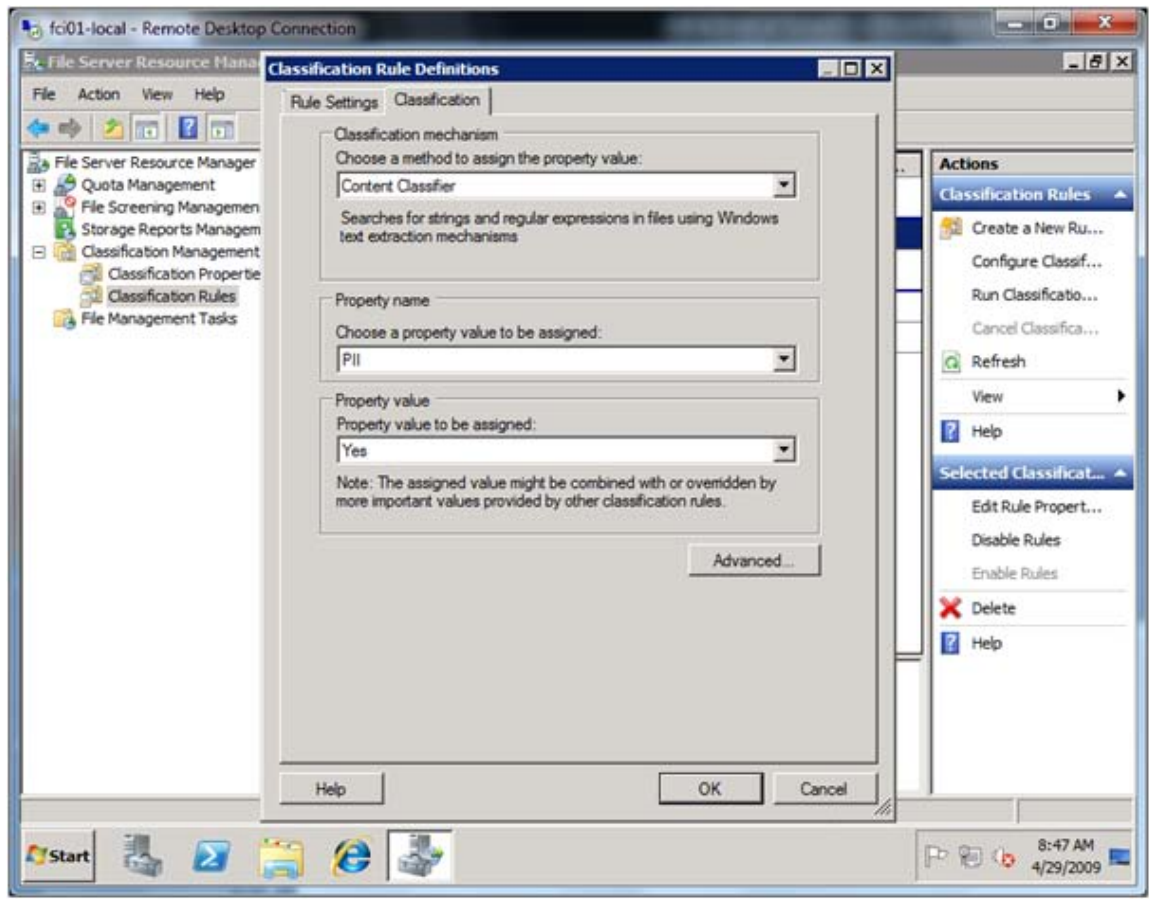


Figure 23: Defining a Content Classifier rule.

Figure 24 shows the content that's being searched for—in this example, a regular expression that matches on US Social Security Number patterns.

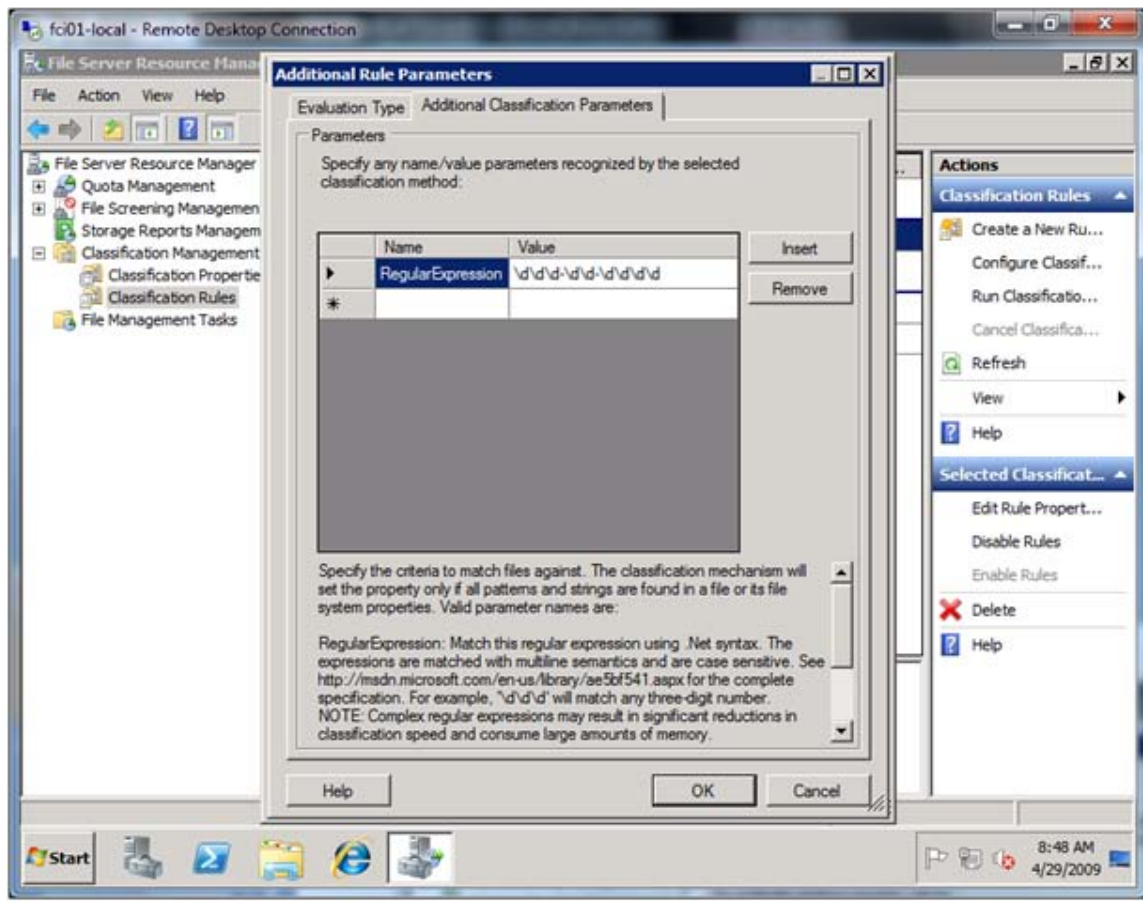


Figure 24: Defining the content to search for.

Third parties can provide additional classifiers, and third parties can also use the FCI application programming interface (API) to apply classification properties or to read those properties—for example, an auditing solution might use these properties to prioritize the files that are included in a security audit.

Note

The screenshots for FCI were taken from pre-release versions of R2 and may change in the final shipping product. These screenshots were drawn in part from <http://blogs.technet.com/filecab/archive/2009/05/11/classifying-files-based-on-location-and-content-using-the-file-classification-infrastructure-fci-in-windows-server-2008-r2.aspx>, which includes a full discussion of the feature.

Tip, Trick, Technique 12: Remote Command-Line Administration in R2

Windows PowerShell v2 introduces a new form of remote management based upon the industry-standard Web Services for Management (WSMAN) and Microsoft's Windows implementation, Windows Remote Management (WinRM).

WinRM is a Web Services-based protocol, meaning it operates over HTTP. By default, this means it uses ports 80 and 443, although those port numbers are configurable. The WinRM service listens for incoming requests, then passes those requests to registered applications—including PowerShell. For security purposes, administrators can govern the applications that are allowed to register with WinRM. Essentially, WinRM replaces the older and more cumbersome Remote Procedure Call (RPC) protocol; WinRM offers easier compatibility with firewalls.

PowerShell v2 includes a set of cmdlets designed to configure and enable remoting through WinRM, and a set of cmdlets designed to establish sessions with remote computers. Once you have created an authenticated session from your local PowerShell instance to a remote instance, you can engage in two distinct management scenarios: 1:1 and 1:n.

A 1:1 scenario basically provides you with a remote interactive command-line window, not at all unlike SSH found on most Unix/Linux operating systems (OSs). A 1:n scenario allows you to invoke PowerShell commands and have them run *on* multiple remote computers *in parallel*, with the results being brought back to your computer. This makes multiple-computer management virtually the same as single-computer management and makes it easier to manage even a highly-distributed IT infrastructure.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimedpublishers.com>.