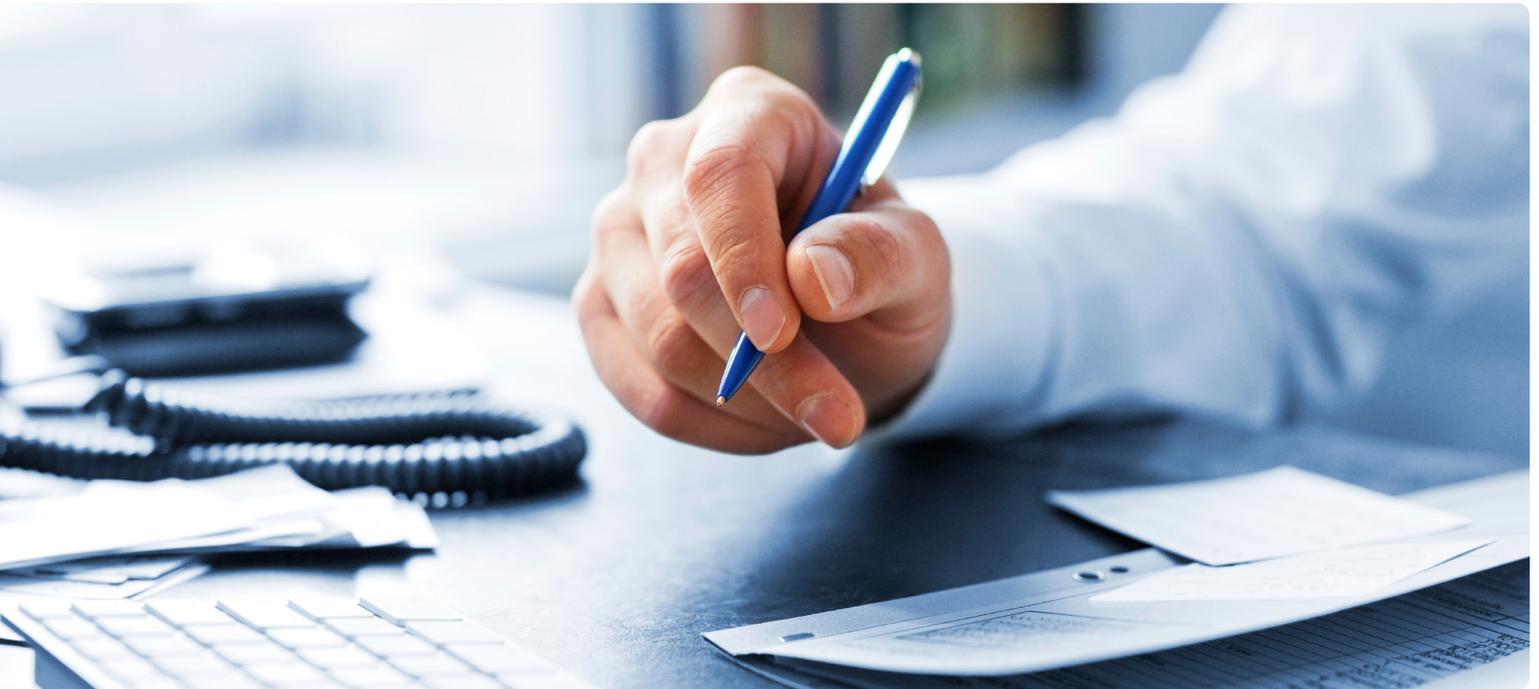


In Desktop Management, One Size Does Not Fit All



Abstract

One of the biggest complexities in desktop management is the fact that no organization has a single desktop configuration need. Sure, some organizations buy the same hardware and lock down the desktop configuration, but the fact is that some users continue to have unique needs. Some users dial (or VPN) in more than they connect to the corporate LAN. Some users need different applications than most everyone else, and some need the same applications configured somewhat differently. And, inevitably, differences in hardware and software start to creep in: new desktops and laptops, newer computer models when the manufacturer discontinues the old one, and of course newer versions of Windows, Office and other key applications.

Initially these differences can seem minor, and many organizations attempt to tackle them using native tools like Group Policy or home-built solutions like logon scripts. But those approaches tend to get increasingly complex over time, with more and more "conditions" to determine which

computers and which users get which specific configurations, software and other managed elements.

There should be a better way. This white paper explores the limitations of traditional approaches to desktop management and identifies the features you need in an effective solution.

The problem: there's no single scenario

Start at a high enough level of desktop management, and every user and computer looks the same. They all need Office, they all need certain corporate applications, and they all need the same anti-malware software. Easy enough.

But when you start digging just a little deeper, little differences crop up. Only **these** users need their M: drive mapped **here**, while others need it mapped **there**. The Finance department needs an F: drive mapped to support their old software. Human Resources employees need to use **these** printers, which are in a controlled space so that confidential documents aren't lying

Administrators spend an amazing amount of time coping with these “edge cases” that live just outside the “standard” template.

out for everyone to see; only the folks in the Marketing department are allowed to use **that** expensive color printer.

Keep digging, and the differences become more extreme. Account Managers are in the office only once a week, so we can't deploy anti-malware updates to them on the same schedule as everyone else; the updates will bog down their VPN connections. Only the engineers can have Visio, because we've bought only so many licenses for the software. Network admins need to get a set of mapped drives and printers, but **not** when they're logging onto a server. Users logged into a Remote Desktop Services (RDS; formerly Terminal Services) computer need an entirely different set of configuration elements.

Organizations try to address these needs by building ever more complex Group Policy filters or by writing logon scripts that have an expanding number of branching logic conditions.

Dig to the deepest level, and things start to get dynamic. We need to update computers that have **this** registry key set to **this** value—and part of the update will actually set that registry key to that value, so that we don't update the computer a second time. Or we need to deploy a patch to computers with **this** version of a particular DLL. Writing that criteria into a Group Policy filter can be impossible; creating a logon script to handle it can be just as difficult and impractical. Even massive solutions like System Center Configuration Manager can't always handle the complex, super-specific targeting criteria that we need in order to get the job done.

The point is that no matter how homogenous you think you've made your environment, these little differences always come up eventually. Administrators spend an amazing amount of time coping with these “edge cases” that live **just** outside the “standard template.” There is a better way.

What's needed: granular targeting criteria

Limitations of traditional approaches

Let's say you need to apply some configuration setting to just a subset of your computers, such as “all of the computers running Windows XP, that have Service Pack 3, that do not have some particular DLL file, that **do** have this registry key but **don't** have it set to some specific value.” That's a perfectly common set of **targeting criteria**, involving several elements that must be present and several that must **not** be present. The traditional problem with Group Policy is that getting that granular with targeting criteria can be difficult, if not impossible, depending on what your set of criteria includes. The criteria above, for instance, would be tough to use in a Group Policy WMI filter or to code in VBScript or PowerShell. Writing logon scripts to implement the criteria can also be impractical and time-consuming, and often requires specialized skills that just one or two administrators on your team may have.

Part of the problem with getting the granularity you need is that Windows hasn't traditionally supported any kind of client-side criteria-processing mechanism. Group Policy, remember, is primarily intended to be processed on the domain controller, which decides what Group Policy objects (GPOs) to send to a client computer. Newer versions of Windows do support more granular client-side Group Policy criteria, but that requires a bit of back-and-forth with the domain controller that doesn't always offer the best performance. Plus, specifying criteria can be tough—tell a GPO that it should apply only to computers with such-and-such a DLL installed locally, and you'll realize the complexity and limitations right away.



What a client-side agent can deliver

The solution lies in providing a client-side agent (noting that “client” can also refer to computers running a server operating system, since they are a “client” of whatever desktop management policies you are attempting to deploy). That agent can provide the rich, local, criteria-processing capabilities that Windows doesn’t really have built in. Once a local processing agent is in place, it can let you target your desktop management policies with a much broader range of criteria, including the following:

- The authenticating domain
- The computer’s home domain and group memberships
- The user’s home domain and group memberships
- OU membership for the user and computer
- The computer’s site
- The computer’s name and operating system version
- The computer’s machine type (desktop, laptop, tablet, embedded, and so on)
- The computer’s IP or MAC address
- The existence or version of a file
- The existence or value of a registry key
- Whether the computer is running inside a virtual machine
- For RDS sessions, the application name, IP address, initial program, session name, and so on
- The current time
- Whether the user is logging on or off, or whether the criteria is being evaluated as part of a regular policy refresh
- The type of network connection (LAN or dial-up)

A client-side criteria processor could also provide much more dynamic, at-the-moment evaluation of criteria. Things like environment variables, the location of system DLLs, and free space on specific disks could all be quickly evaluated. All of this would combine to give you the super-granular targeting criteria that you need—in fact, it would probably give you **greater** granularity than the scripts and GP WMI filters you’ve been relying on until now.

This ability to evaluate such dynamic criteria would be a real asset. Group Policy, at its core, is essentially static: it looks at things like site and OU membership, which change infrequently. Group Policy WMI filters can evaluate dynamic criteria, but are fairly complicated to write—and WMI itself isn’t well-documented anywhere, so simply **finding** the right bit to build into a filter can be a monumental task. With a solution that already **knows** where to find the data and that can evaluate it on **demand**, you can apply exactly the right configuration policies to exactly the right computers and users.

How criteria should be specified

What you **don’t** need is a specialized scripting language to specify all of your targeting criteria. Logon scripts have been a necessary evil for decades, but that doesn’t make them any less of an evil: they’re complex and time-consuming to develop, and require specialized skills that are all too rare in the world of Windows administrators. Besides, no smart organization will commit to any kind of proprietary scripting language that might or might not be around in a few years—especially when Microsoft itself doesn’t yet have a good track record for maintaining suitable scripting languages over the long haul.

Instead, administrators need to be able to specify targeting criteria via a simple, intuitive graphical user interface (GUI). Click on “Registry Value,” enter the key to check, and enter the value you’re looking for—bang, criterion specified. No need to mess around with WMI registry providers, PowerShell HKCU: drives, obscure command-line utilities, or anything else. Point, click, done—that’s what Windows has always been about, right?

Logon scripts have been a necessary evil for decades, but that doesn’t make them any less of an evil.

Point, click, done—
that's what Windows
has always been
about, right?

Administrators will need to create named sets of criteria for different circumstances: "All of the Finance users still running v9.2 of their software," or "All of the IT admins who aren't logging onto a server," or "Sales guys who are connected via dial-up, running Windows XP, and in possession of an old antivirus signature database." Very specific, but very easy to construct and save into a set. Specific configuration actions can then be applied to one of those criteria sets, getting exactly the right settings, software, and other managed elements to exactly the right users and computers, every time.

When criteria should be applied

We tend to think about applying desktop management policies at a single time: user logon. That makes sense, because there are a lot of configuration elements that need to be set up just before the user begins working: mapped drives, printer assignment, software deployment, and so on.

However, other elements are best applied at logoff, such as software patches, service packs, and other things that need to be done, but that will hold up the user's productivity if they're done at logon.

Obviously, scripts can be written for both logon and logoff—but scripts **can't** be used to continually apply configuration settings. With a local, client-side processor of some kind, you **can** have continual refresh, not unlike Group Policy's own every-hour-or-so refreshing. Periodic refreshes are good for configuration elements like security settings, or for enforcing a consistent user operating environment.

Summary: a wish list for granular desktop management

So, what's needed is more granular desktop management, without the complexity and overhead of scripting, and with more ease-of-use and flexibility than Group Policy filtering. What specific capabilities should you be looking for?

- Client-side processing of targeting criteria, which enables both dynamic evaluation and higher performance of the overall solution
- A vast range of items that can be included in targeting criteria, including registry, files, computer information, user criteria, platform information, and more
- Specific support for virtual and RDS environments
- The ability to create complex, multi-part criteria that include Boolean logic (AND, NOT, etc.)
- GUI-based criteria builder
- The ability to create custom scriptlets to implement targeting criteria not supported directly by the solution, particularly with regard to environment-specific scenarios such as custom applications or operating conditions
- The means to target a set of configuration actions to a defined set of targeting criteria, applying that set of actions to the targeted computers and/or users
- Support for dynamic criteria evaluation, enabling computers to be targeted based on their exact local conditions at the time

With these capabilities, you can start supporting the realities of your environment—including all of its awkward, ugly edge cases—without having to engage in complicated scripting, and without making your Group Policy infrastructure more complex or difficult to manage.

Desktop Authority Management Suite

Desktop Authority Management Suite enables administrators to proactively provision and manage a productive, secure and flexible Windows user environment that automates users' access to resources and applications. The suite comprises three solutions: Desktop Authority, Privilege Manager and MSI Studio.

Desktop Authority provides exactly the granular targeting criteria organizations need, and enables administrators to build complex criteria sets using an intuitive, simple GUI. Dozens of configuration elements can be checked, including more than a hundred dynamically evaluated criteria. Hundreds of configuration settings can then be applied based on those criteria sets. In short, Desktop Authority is creating a centrally-managed, high-performance, highly-scalable desktop management solution that provides all the benefits of Group Policy and logon scripts, but without any of the complexity. Desktop Authority can even enhance and supplement an existing Group Policy environment by providing more granular, client-evaluated criteria for key configuration elements. Visit www.quest.com/desktop-authority-management-suite for more information.

Desktop Authority Management Suite provides all the benefits of Group Policy and logon scripts, but without any of the complexity.

For More Information

© 2013 Dell, Inc. ALL RIGHTS RESERVED. This document contains proprietary information protected by copyright. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose without the written permission of Dell, Inc. ("Dell").

Dell, Dell Software, the Dell Software logo and products—as identified in this document—are registered trademarks of Dell, Inc. in the U.S.A. and/or other countries. All other trademarks and registered trademarks are property of their respective owners.

The information in this document is provided in connection with Dell products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Dell products. EXCEPT AS SET FORTH IN DELL'S TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT,

DELL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL DELL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF DELL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Dell makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Dell does not make any commitment to update the information contained in this document.

About Dell

Dell Inc. (NASDAQ: DELL) listens to customers and delivers worldwide innovative technology, business solutions and services they trust and value. For more information, visit www.dell.com.

If you have any questions regarding your potential use of this material, contact:

Dell Software

5 Polaris Way
Aliso Viejo, CA 92656
www.dell.com

Refer to our Web site for regional and international office information.

