Realtime publishers

The Shortcut Guide™ To

User Workspace Management

sponsored by

RES software

Greg Shields

Introduction to Realtime Publishers

by Don Jones, Series Editor

For several years now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We've made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book's production expenses for the benefit of our readers.

Although we've always offered our publications to you for free, don't think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you $40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the "realtime" aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We're an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I'm proud that we've produced so many quality books over the past years.

I want to extend an invitation to visit us at http://nexus.realtimepublishers.com, especially if you've received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you're sure to find something that's of interest to you—and it won't cost you a thing. We hope you'll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Introduction to Realtime Publishers i
Chapter 1: What Is User Workspace Management? 1
Decoupling the Layers of Windows 2
Encapsulation and Decoupling 3
The Benefits of Decoupling 3
User Workspace Management Enables Decoupling 6
The Ridiculous Pain of User Profiles 7
The Case for User Workspace Management 9
Content, Context, and Security 11
Content 12
Context 12
Security 13
Users Get Personalization, IT Gets Control 13
User Workspace Management Changes Everything 14

Copyright Statement

Chapter 1: What Is User Workspace Management?

I spent *four hours* yesterday upgrading *exactly one desktop* from Windows Vista to Windows 7. *Four hours!*

Yet that's not the most disheartening part of this story. You already know that installing a fresh copy of Windows has today become fantastically easy. You can create a new instance of Windows 7 in a few clicks of the mouse: Agree to an EULA, select a disk drive, enter a license key, and in the time it takes to brew a fresh pot of coffee, you've got a fully-functional OS instance.

The problem is *that's the easy part.* The real pain in terms of time, effort, and labor comes not in provisioning that Windows OS, but in dealing with all the little bits of user personality information that one must transfer from one OS instance to another.

You know the story: Upgrading an OS, or moving a user to a new desktop, or creating a remote application or hosted desktop infrastructure. All of these activities are core to the job of an IT professional. And all are actually fairly easy to accomplish—except for that all-important process of managing, maintaining, transferring, and/or replicating your users' workspace data.

In my experience yesterday, installing Windows 7 consumed about 30 minutes of the project's four hours. The other three-and-a-half hours saw myself and my user seeking out every scattered bit of their stored information, transferring what they needed to a temporary location, remembering and then subsequently re-transferring a few forgotten items, and eventually merging that data back into the new OS.

Three-and-a-half hours for all this work, *for a single workstation! Ridiculous!* Whether you're upgrading OSs, migrating them, or simply need a better way to administer user data across multiple (and sometimes simultaneous) points of access, *there has got to be a smarter approach to managing your users' workspaces.*

What that "smarter approach" might look like is the central theme behind this Shortcut Guide. In its four chapters, you'll learn how User Workspace Management can fully decouple your users' workspaces from their operating systems (OSs). As a result, your all-critical user data becomes capable of roaming across every class of IT infrastructure, while you gain more-powerful tools in managing it.

But before we talk about solutions, let's first analyze the problem by taking a look at the layers within the Windows OS. By decoupling those layers, User Workspace Management enables some very powerful trickery that make my problem—as well as many of yours—go away completely.

Decoupling the Layers of Windows

When you sit down in front of a Windows computer, it's easy at first blush to see that computer as a single entity. You interact with an OS. You install and work with applications. You customize settings for your printers, your desktop environment, your USB drives, as well as the settings specific to the applications you use. It is the combination of these elements that makes up the computer on which you're reading this guide.

Yet that single entity that is your computer is really made up of *a set of layers*. Each layer in that stack contributes in some way to the creation of a fully-useable computer.

Think for a minute about those layers that could make up a typical Windows computer. One representation is shown in Figure 1.1. At the figure's base is your hardware itself. That hardware might be a laptop or a desktop. Or it could be one of any number of virtual or otherwise non-standard technology infrastructures.

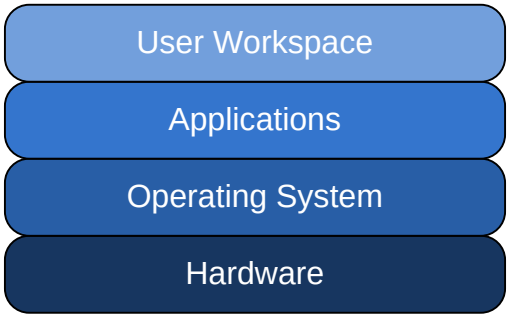

Figure 1.1: The many layers of the Windows OS.

For example, instead of being a traditional desktop, that computer's hardware might exist atop a hosted virtual desktop, located somewhere within a data center. It could also be sourced from a remote applications infrastructure such as Microsoft's Remote Desktop Services or Citrix XenApp. Essentially, any infrastructure that allows for the installation of an OS can be the base layer of this OS stack.

Atop that hardware is the installation of the OS itself. In my story, that OS started out as Windows Vista but was later upgraded to Windows 7. Microsoft Windows is available in a number of versions—both for servers and workstations—of which you probably support more than one version today.

Layered atop that OS is a set of applications. These applications might be installed locally. Some might be deployed through an automated software deployment solution, while others might be streamed down to that computer using application virtualization. Even others might be applications atop a remote applications infrastructure, with their access enabled through links within the local computer.

On top of all of these layers are your user customizations themselves. As stated before, these are the settings that make up your desktop environment. Some you may have customized yourself, while others may have been enforced through a stated company or IT policy. Important to recognize here is that the sum total of these configurations is what comprises your *user workspace*.

Encapsulation and Decoupling

With this understanding in mind, let's assume that through the use of one or more software products, it becomes possible to logically encapsulate these layers. This process of encapsulation creates hard lines between the user workspace and the applications and OS it customizes. It does the same thing between applications and the OS, as well as between the OS and the hardware on which it rests.

Without this logical encapsulation in place, managing that computer is a very difficult thing to do. Without some automated software deployment or application virtualization solution, adding a new piece of software means I have to manually touch each and every computer. Updating a piece of software requires a lengthy uninstallation-followed-by-reinstallation process. Lacking solutions like these, working with OSs and applications in any form is both difficult and time consuming.

The same holds true at the user workspace layer as well. User data is natively stored within a Windows user profile, with that profile containing all the customizations that elevate a freshly-installed Windows OS into something that's personalized for each user. Lacking logical encapsulation at this layer, moving that user data between different computers becomes challenging to the point of absurdity. If you've ever spent four hours trying to upgrade a single user's computer from one OS to another, you know this pain.

The Benefits of Decoupling

Now, envision an environment where each of these layers *has* been fully encapsulated. Such an environment would see a full decoupling of each layer from the others. Here, for example, a hard line would exist between the user's workspace and their applications. The user's workspace would be independent from the OS as well as its hardware.

In such an environment, that user's workspace would immediately become fully *mobile*. This becomes possible because that workspace is no longer directly reliant on the layers below it. Because the decoupled workspace now exists in its own independent and isolated layer, that layer can trivially move between different computer instances. Such an environment would gain some very interesting benefits to administration.

For example, consider the change that is represented in Figure 1.2. Here, the user workspace has been decoupled from its OS and installed applications. With the aforementioned encapsulation in place, it becomes possible to swap out the underlying OS along with its applications.

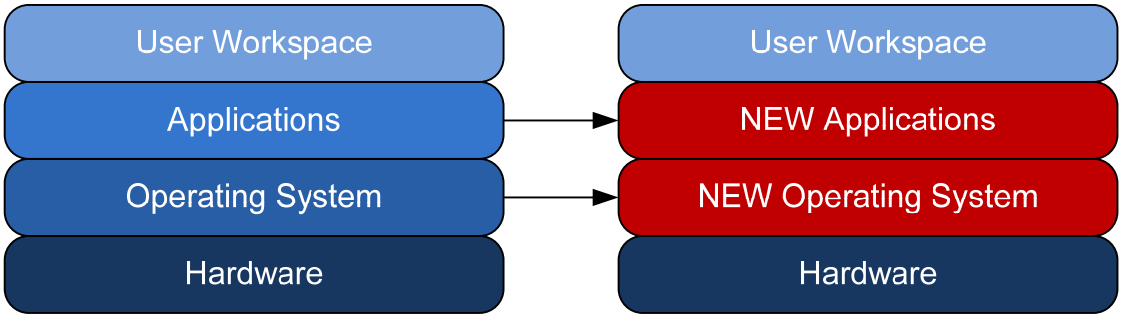

Figure 1.2: Decoupling the user's workspace enables it to work atop new computer instances.

This situation exactly mirrors the OS upgrade I explained in the beginning of this chapter. If the user workspace on the computer in my story were encapsulated in this way, upgrading that OS would require only three very simple steps:

- *Synchronize.* The first step would be to ensure the user workspace on the computer I plan to upgrade was synchronized with an offline copy. Depending on the solution, this process could have occurred automatically in the background or through some administrator action.
- *Update.* Once an offline copy of my user workspace was complete and verified, I would need only to insert the Windows DVD and complete the upgrade.
- *Reapply.* My final step once the upgrade and application installations were complete would be to reapply the user workspace back to the new OS. That workspace would seamlessly apply on top of the new OS, returning the user's personality information back to the upgraded OS.

This process sounds vaguely similar to the manual steps I went through to locate the user's information, copy to an offline location, and merge it back with the upgraded OS. However, different with a User Workspace Management solution is the logic that *fully automates these steps*.

User Workspace Management Compensates for OS Differences

Critical also to recognize is that different versions of the Windows OS have very different structures for user data. For example, versions prior to Windows Vista store user data in the C:\Documents and Settings folder. Windows Vista and later versions store that data in C:\Users. Windows Vista and Windows 7 also have different folder structures for storing user data within profiles. Registry structures can also be different between different OS versions.

Yet although these changes are substantial between versions, they can be compensated for. A decoupled environment would need to include the necessary logic to translate between different OS versions; however, one that included such logic would fully automate the transfer of user workspaces between OSs.

Along the same lines, a decoupled environment could also enable *hardware independence* for user workspaces. Because of its encapsulation, that same user workspace could be seamlessly applied to any class of hardware infrastructure as well (see Figure 1.3).

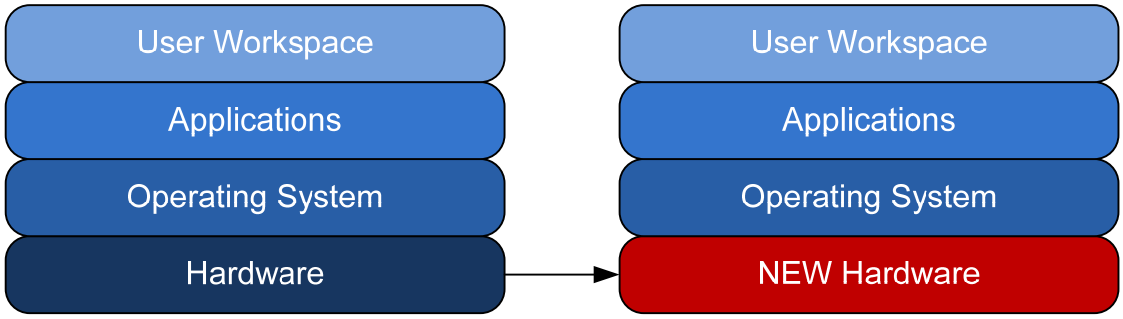

Figure 1.3: Decoupling creates hardware infrastructure independence.

That hardware could be realized as a replacement desktop, a changeover between a desktop and laptop, or among any of the multiple virtual or application infrastructure technologies available today. With this flexibility, IT immediately loses a set of traditionally challenging hurdles:

- *Seamless desktop to laptop.* Swapping a desktop for a laptop becomes immediately seamless. The same user workspace that was created for the desktop would be immediately available once logged into the laptop.
- *Seamless desktop to shared desktop.* Similar to the change for laptops, users who move from their primary computer to a shared desktop—such as one in a conference room or kiosk—would immediately and seamlessly have access to the same user workspace and same resources.
- *Seamless desktop to virtual desktop.* Environments making the move to hosted virtual desktops needn't worry about user settings between physical and virtual environments. Users who log onto their hosted virtual desktop automatically and seamlessly have the same experience as with their primary desktop.
- *Seamless desktop to remote application.* Lastly, remote applications infrastructures needn't necessarily create new workspaces for users. When users click links to launch remote applications or desktops, those environments include the exact same workspace configuration as on their primary desktop.

Decoupling additionally eliminates the reliance between the user workspace and *its installed applications* (see Figure 1.4). With a decoupled user workspace, it becomes trivial to change the set of available applications within any hardware infrastructure. In such an environment, user personalization data that functions with one application will automatically work with that application if it is available. Replacement or updated applications can leverage existing personalization data or create their own.

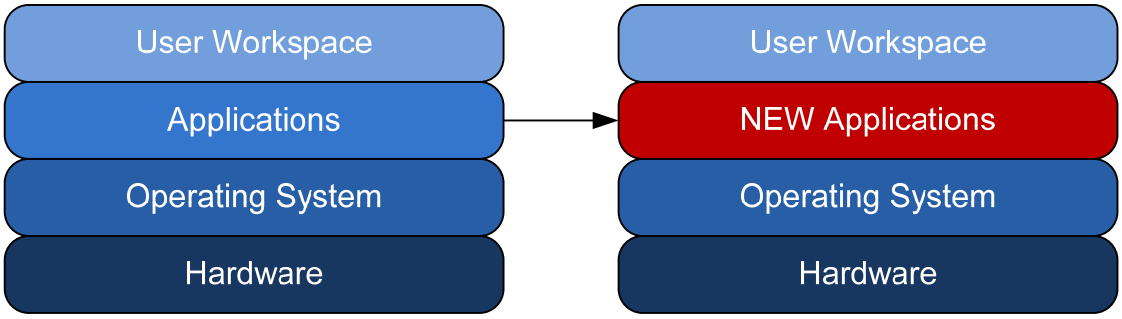

Figure 1.4: Decoupling enables application flexibility.

Methods for application delivery grow more flexible as well. When user workspaces are decoupled from applications, the applications can be made available to users in many different ways with the assurance that personalization is retained. For example, an application that was at one point directly installed to a desktop can also be made available through a remote applications infrastructure. Another within a remote applications infrastructure can be later streamed to desktops on-demand. In every case, users are always guaranteed that their personal settings will apply no matter how the application is delivered.

> **User Workspace Management and User Virtualization**
>
> To the keen eye, this concept of layer encapsulation and decoupling should resemble another important topic in IT—*virtualization*. The concepts of User Workspace Management and User Virtualization are similar in terms of scope and capabilities.

User Workspace Management Enables Decoupling

The workspace decoupling in each of these scenarios is made possible through the software solutions that enable User Workspace Management. Such solutions integrate with the various hardware infrastructures already in your environment to enable the seamless roaming of user workspaces.

Yet, how does this logical encapsulation and decoupling actually work? And how close are these solutions to long-held technologies such as traditional Windows profiles? Although they appear similar in construct, you'll find substantial differences in delivery as well as workspace management between a User Workspace Management solution and what you're used to seeing with Windows profiles.

The Ridiculous Pain of User Profiles

When you envision the concept of a user workspace, it is perhaps easiest to start by considering traditional Windows user profiles. User profiles are designed to contain settings and configurations that are unique to a particular user. For example, when a user changes their desktop background, the information about that configuration change is stored within their user profile. The same holds true for essentially every look-and-feel change to the user's workspace as well as custom configurations to that computer's applications.

Windows user profiles are obviously user-centric, but they are by nature device-centric as well. A traditional Windows user profile exists on exactly one computer (see Figure 1.5), with each profile corresponding to a unique user. Stored within that computer's Users or Documents and Settings folder, each user profile contains the custom settings that have been defined *for that user* and *for that device*.

Figure 1.5: With traditional user profiles, the user's workspace is defined for each user and for each individual device.

This solution works well when users log in exclusively to the same computer. This is a situation that was common in IT's "old days": Users begin their day by logging into their assigned computer, they interact with it throughout the day to do their jobs, and log out as they leave in the evening. *One person equals one computer.*

Yet this one-computer-per-person concept is a quickly-dying scenario. Today's business needs often require users to move between computers. Also needed are entirely-separate remote application infrastructures such as Microsoft's Remote Desktop Services or Citrix's XenApp. Some businesses leverage the use of virtual hosted desktops as a replacement to or in addition to traditional desktops.

In each of these examples, the job of a single user requires the support of more than one computer. In some cases, a single user can require the support of multiple computers *at the same time*. With a user's workspace constrained by device, each new login requires an entirely separate process of workspace configuration to make it ready for use. Essentially, every time a user logs in somewhere new, they must restart the process of making their workspace comfortable and useful for them. As the number of endpoints increases in such an environment, so does the number of separate workspaces. The result is a substantial amount of wasted time, and ultimately an unhappy user base.

Microsoft's native solution for this problem was and continues to be the creation of *roaming profiles*. These enable the "roaming" of a traditional user profile to multiple endpoint computers (see Figure 1.6). Using a roaming profile, the user's workspace is copied from a central storage location to the endpoint at the moment the user logs in. While logged in, the user enjoys the workspace customizations they've stored in their profile. When they later log out of that workstation, the roaming profile is returned back to the storage location to await the next login.

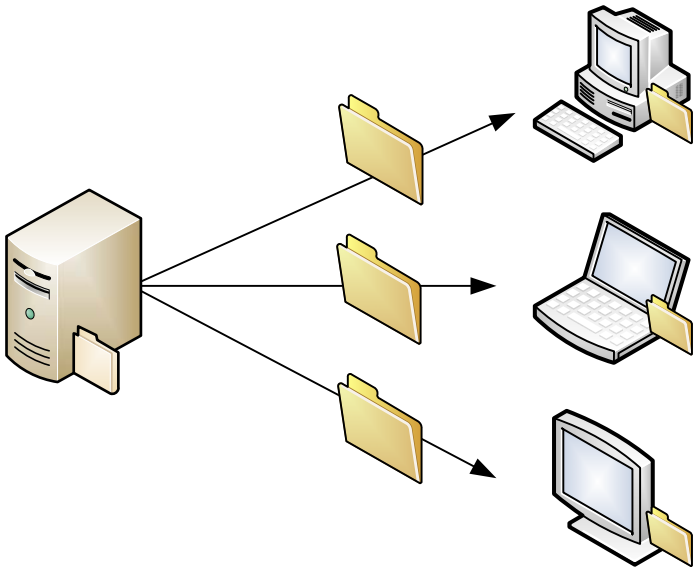

Figure 1.6: Roaming profiles centralize the storage of the user's workspace, delivering it via file copy to devices during login.

Although a great idea in concept, this all-or-nothing approach actually complicates rather than resolves the problem in a number of ways. Think about some of the problems environments experience today with roaming profiles:

- *Synchronization.* The process of copying profiles to and from a storage location at login and logout creates synchronization problems which are exacerbated when multiple endpoints are used simultaneously.
- *Capture.* Roaming profiles only capture and deliver configurations that are contained within the user's profile and user registry hive. Some applications do not store user-specific settings in these locations, and as such, they are not properly captured by the user's profile.
- *Accumulation.* Roaming profiles accumulate changes over time. Copied down to any number of very different endpoints—desktops, laptops, Terminal Servers, hosted desktops, and so on—a roaming profile gathers and stores artifacts from each. This occurs even when artifacts don't merge properly with some endpoints, causing broken links and orphaned configurations.

- *Delay.* The copy process for a roaming profile is folder-centric, which means that an entire folder of files must be transferred between storage and endpoint at each login and logout. Even a single large file on a user's desktop can significantly increase the time required to login and logout.
- *Management.* Lastly, and most importantly, the centralized management of user profiles can only be accomplished through Group Policy or via scripts. Both of these solutions occur after the profile is delivered rather than in conjunction with it, making profile management a reactive rather than a proactive solution.

It should be obvious within this short narrative to see the limitations in both solutions. Traditional user profiles offer stability but without the needed multiple-endpoint extensibility. Roaming profiles, in contrast, gain the extensibility but by sacrificing stability. In the end, neither is an effective solutions for today's dynamic computing requirements, *and neither fully decouples the user's workspace from the applications and OS that lie beneath.*

The Case for User Workspace Management

The central problem in both of these solutions is that each is focused on managing devices and users. Using native Windows profiles, User A gets Profile A, which is delivered no matter whether they're logging into Desktop A, Server B, or Remote Application C. The result is a monolithic answer to a fine-grained problem, when what is needed is a solution that doesn't really care about devices, users, or delivery endpoints. Rather than focusing on devices and users, such a solution instead *should desire only to manage the content* that makes up your user's workspace.

Think for a minute how a content-centric solution would be quite different than native Windows profiles. In a content-centric solution, neither the device nor the user really matter anymore. Such a solution dynamically constructs the user's workspace at each and every login. The composition of that workspace is based on a combination of enforced configurations as determined by company and IT policy along with others that are unique and personalized to each user. Each setting is granularly configured as an individual object that is associated with the user's workspace. Although the user's native "profile" still exists in a content-centric solution, there is no longer a sense of delivering that profile as a whole entity.

Figure 1.7 shows a graphical representation of how this might look. Here, the user's profile still exists as a structure within each possible endpoint. Different, however, is the presence of a third-party software agent on each endpoint that interacts with an external database. Contained within that database is the comprehensive set of workspace configurations that are required by every endpoint. When the user logs into any endpoint in the environment, their workspace is granularly composed rather than brute-force applied through an all-or-nothing profile transfer.

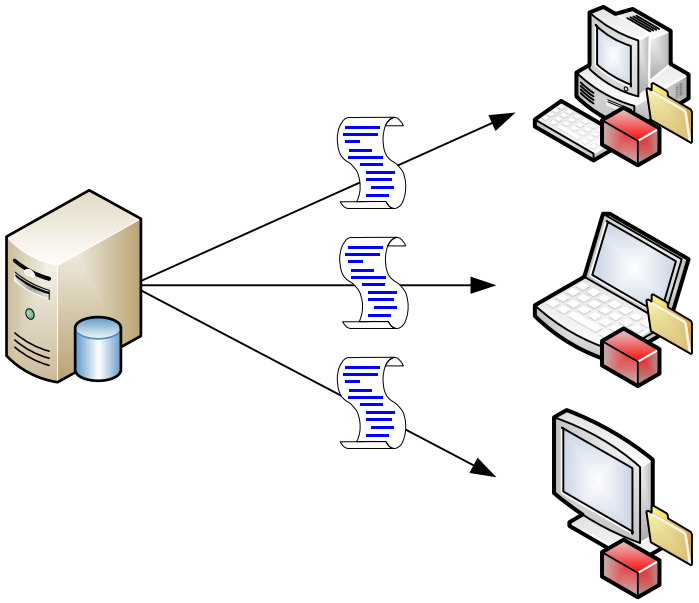

Figure 1.7: A content-centric solution granularly composes the user's workspace rather than resorting to an all-or-nothing folder transfer.

This granular composition of the workspace accomplishes a number of tasks all at once:

- *Decoupling.* Each possible workspace configuration is stored within the external database and composed within the user's session at the moment they log in. This decoupling of the user's workspace content from their actual profile eliminates the accumulation and capture problems noted earlier with traditional profiles.
- *Workspace pervasiveness.* By decoupling the configuration elements of the workspace from the workspace itself, that workspace composition becomes applicable to any endpoint. Thus, the same workspace configuration that applies to Desktop A could also seamlessly apply to Server B or Remote Application C. Further, if a user found themselves logging into Desktop A, Server B, and Remote Application C at the same time, they would assuredly find the same workspace in each.
- *Synchronization.* The communication to the third-party agent needn't flow in only a single direction. Such an agent can regularly watch for other configurations that have been personalized by the user, such as an updated desktop background or different screen saver, and synchronize those changes back with the database with relatively little delay.

- *Item-level Management.* Traditional profiles can generally only be centrally managed through Group Policy or the use of custom-coded scripts. Although the combination of both gives a measure of control over workspace elements, that control is always reactive—occurring after the profile is laid into place. A content-centric solution enables the management of individual workspace configuration items at an item-by-item level.
- *Standardization plus personalization.* Finally, such a solution has the capability of defining some elements that are required, while leaving other elements to be personalized by users. Users like the ability to customize certain parts of their environment, giving them a comfortable look and feel. Other parts must be controlled by corporate or IT policy. A content-centric solution's item-level management gives IT the power to determine which configurations are controlled and which can remain personalized.

The end result with a User Workspace Management solution is that resources and applications become exposed to the user as a function of the endpoint agent. That same agent enables administrators to wield granular control over which items are exposed, which are controlled or otherwise locked down, and which are available for user customization.

Content, Context, and Security

And yet the management of content and its exposure to users is only the first component of a fully-realized User Workspace Management solution. In environments with such solutions, the client-installed agent is charged with composing the user's workspace based on the union of three distinct but important factors: *content, context, and security*, shown graphically in Figure 1.8.

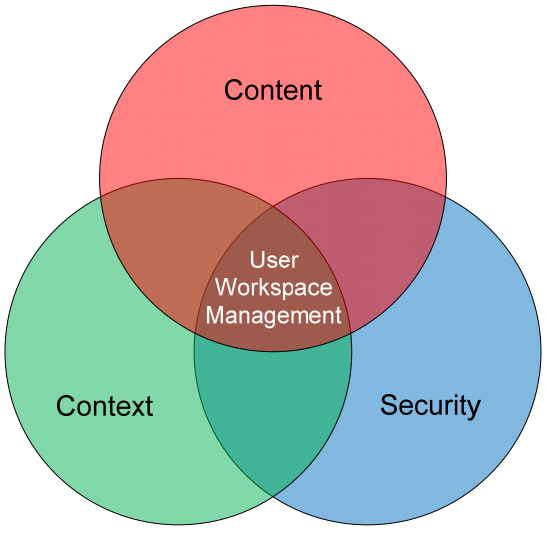

Figure 1.8: User Workspace Management is the union of content, context, and security.

Let's take a look at each these three elements in turn. You'll see how the combination of each is required to truly serve all the needs of the dynamic business computing environment.

Content

Content relates to the resources that users require to complete their assigned duties. This chapter has already discussed some of those components, with a more-detailed conversation to occur in Chapter 2. However, the recognition that users require access to various applications, data, printers, and other user settings is important to understanding the types of content that a User Workspace Management solution can deliver.

Context

Delivering content is one thing, but delivering it appropriately is yet another. Recognition of the user's context is a critical component of today's dynamic computing environments but remains a glaring omission in traditional profile-based solutions.

Today's users more and more require the use of multiple mechanisms for accessing their applications and data. Some of those mechanisms require the direct installation to local hardware, while others are hosted via network-based remote infrastructures. Others roam among areas of the network or even between security zones. The problem intrinsic to this horizontal spread of application availability is in recognizing *where the user is* and *via what mechanism are they accessing their resources*.

Within the parameters of a User Management Workspace solution, context can be based on a spread of different factors. Obviously, the user's identity itself is a central aspect of their context: For example, the "jdoe" account is logically linked to the person named "Jane Doe." However, context here can also be based on other factors, such as the location from which that user is connecting, the time of day or day of week, and the device on which the user is working. Each of these elements of *context* is used to determine what kinds of *content* are delivered to the user.

It's easiest to visualize this by means of a simplistic example. Consider a business that spans multiple floors in an office building. In such a business, there are likely a number of printers on each floor. In the traditional network, finding the right printer that is close to you and contains the right features can be a guess-and-check process. Without automation, users must search for printers that appear close in proximity based on their name or network label.

Context-sensitivity within a User Workspace Management solution enables the workspace itself to identify the right printer based on proximity and feature set. In this example, when a user logs out of one computer on the first floor and into another on the second, their workspace will automatically update the printer configuration to connect to the closest printer.

Security

Combining context with content is important not only for resource delivery but also in applying the right security. Completing the three elements in this union are the security configurations that apply to delivered content. Those configurations can and will be different based on the context of the user.

For example, consider a user that is accessing an application and associated set of data using their desktop within the brick-and-mortar office. Located within the protective confines of the office, that user likely requires relatively open access to these resources. They should be able to print the data, copy-and-paste it between applications, and accomplish all the tasks that are commonly associated with the resource.

Consider how things change when that user goes home for the evening. First, should that user have access to run the application? Should the user be able to use the application with the same set of data? If access is granted for users in this context, should they have the same privileges with the data that they do while in the office?

Lacking a User Workspace Management solution, it may not be possible to answer these questions. Further, the user's workspace may not be well-controlled enough to actually enact changes if their answers were known. Yet another feature of a User Workspace Management solution's client agent is the ability to granularly secure resource access based on that context.

This concept of context-sensitive security to resources is a fundamental part of a User Workspace Management solution's value proposition. As such, the whole of Chapter 3 is dedicated to exploring this topic of security in depth.

Users Get Personalization, IT Gets Control

The administrative power of such a solution shouldn't go unnoticed. Remember that a User Workspace Management solution is above all a mechanism for IT to define—and ultimately control—the workspace for its users. By entirely decoupling that workspace from its IT infrastructure components, users automatically gain the assurance that their comfortable workspace will always be available irrespective of how they're accessing resources. Yet also gained through the decoupling is the complete control of the workspace itself.

Built into your User Workspace Management solution should be an administrative toolkit that enables IT to identify and set configurations on common workspace elements. Using that toolkit, IT gains the power to identify which areas of the environment they want to control and which they want to leave for user personalization.

For example, using such a toolkit in a highly-locked down environment, IT can configure desktop settings to a corporate standard. Screen savers, desktop backgrounds, timeouts, printers, and attached devices can be assigned a specific configuration that cannot be overridden by experimenting employees. At the same time, applications can be identified, delivered, and configured based on the users' context. Users that require Microsoft Office can be granted access to it, enabling the automatic installation, streaming, or access via remote connection to the application. Users who do not have a need for Microsoft Office can be prevented access to use it through any or all delivery mechanisms.

Other environments don't place a priority on strict controls to user workspaces. They might encourage users to customize their environment with desktop wallpapers or custom printer connections but lock down problematic screen savers. These environments also can identify application and data access based on username, group, time of day, or other factors.

Important to recognize here is *the creation of a spectrum between IT and business control of the environment along with user personalization*. Depending on the needs of your business, your security policies, and your level of needed control, that spectrum can shift in either direction through just a few clicks in the administrative interface.

User Workspace Management Changes Everything

This first chapter has attempted to outline the framework that is User Workspace Management. Here, you began your look into how its powerful framework can dramatically change the ways you administer resource access to users. Continuing on this discussion are Chapters 2 and 3. Chapter 2 goes into detail about the combination of context and content, explaining in greater detail the kinds of content that can be deployed as well why and how user context impacts their availability. Chapter 3 continues the conversation by discussing the linkage between security and people as opposed to security and devices. Decoupling security from the perspective of the device enables a more cohesive application as people move around in their daily lives.

The last step is actually getting there. Chapter 4 concludes the discussion by focusing on the transformation process a business organization will use in moving from traditional profile management to real User Workspace Management.