Using Windows Terminal Services to Run a Single Application

Date Launched: Sep 07, 2004
Last Updated: Sep 07, 2004
Section: Articles :: Windows OS Security
Author: Amit Zinman
Printable Version
Rating: 3.6 / 5 - 23 Votes
1 2 3 4 5

Using Group Policy and some scripting to publish a single application to Remote Desktop users.

Introduction

Some users require only a single application. Installing a costly computer for this seems a bit unnecessary knowing that Windows Terminal Services have been available for quite some time.

However, since Windows is designed to deliver a lot of UI components as well as serve multiple applications it requires some work to get this done. With the benefit of Group Policy and some scripting magic, publishing a single application to users is easy.

All of you, planning on implementing complex Group Policy scenarios should download GPMC for Microsoft. It really helps you out when planning and troubleshooting group policy.

Group Policy

My Terminal Server has users which get only one application but also regular users who get a desktop with a few application. To facilitate this I set up one policy, **All Users**, for all of the users connecting to the Terminal Server, and **App1**, for users getting the first application.

These are the settings for all the users, as copied from GPMC. What it does is leave a clean UI for users, removing most of the Task Bar and Start Menu. It also hides the terminal server's disk drives, leaving the users access only to their own home directory.

All Users

Computer Configuration (Enabled)
No settings defined.

User Configuration (Enabled)

Windows Settings
Folder Redirection
My Documents
Setting: Basic (Redirect everyone's folder to the same location)
Path: D:\Users\%USERNAME%\My Documents

Options: show

Grant user exclusive rights to My Documents	Disabled
Move the contents of My Documents to the new location	Disabled
Policy Removal Behavior	Leave contents

Administrative Templates

Control Panel

Policy	**Setting**
Prohibit access to the Control Panel	Enabled

Desktop

Policy	Setting
Do not add shares of recently opened documents to My Network Places	Enabled
Don't save settings at exit	Enabled
Hide My Network Places icon on desktop	Enabled
Prohibit user from changing My Documents path	Enabled
Remove My Computer icon on the desktop	Enabled

Start Menu and Taskbar

Policy	**Setting**
Add Logoff to the Start Menu	Enabled
Do not display any custom toolbars in the taskbar	Disabled
Force classic Start Menu	Disabled

Hide the notification area	Enabled
Prevent changes to Taskbar and Start Menu Settings	Enabled
Remove access to the context menus for the taskbar	Enabled
Remove All Programs list from the Start menu	Enabled
Remove and prevent access to the Shut Down command	Enabled
Remove common program groups from Start Menu	Enabled
Remove Documents menu from Start Menu	Enabled
Remove Drag-and-drop context menus on the Start Menu	Enabled
Remove Favorites menu from Start Menu	Enabled
Remove Help menu from Start Menu	Enabled
Remove links and access to Windows Update	Enabled
Remove My Network Places icon from Start Menu	Enabled
Remove Network Connections from Start Menu	Enabled
Remove programs on Settings menu	Enabled
Remove Run menu from Start Menu	Enabled
Remove Search menu from Start Menu	Enabled
Remove Set Program Access and Defaults from Start menu	Enabled
Remove user's folders from the Start Menu	Enabled
Turn off personalized menus	Enabled

System/Ctrl+Alt+Del Options

Policy	Setting
Remove Task Manager	Enabled

Windows Components/Windows Explorer

Policy	Setting
Hide these specified drives in My Computer	Enabled
Pick one of the following combinations	Restrict A, B and C drives only

Policy	Setting
Prevent access to drives from My Computer	Enabled
Pick one of the following combinations	Restrict A, B and C drives only

Policy	Setting
Remove Windows Explorer's default context menu	Enabled
Removes the Folder Options menu item from the Tools menu	Disabled

Windows Components/Windows Explorer/Common Open File Dialog

Policy	Setting
Items displayed in Places Bar	Enabled
Places to display:	
Item 1	MyDocuments
Item 2	
Item 3	
Item 4	
Item 5	

The second group policy handles the publishing of the application.

App1

Computer Configuration (Enabled)

No settings defined.

User Configuration (Enabled)

Windows Settings
Scripts
Logon

Name	Parameters
D:\Netlogon\app1.vbs	

Administrative Templates

Desktop

Policy	**Setting**
Hide and disable all items on the desktop	Enabled

Start Menu and Taskbar

Policy	**Setting**
Turn off personalized menus	Enabled

As you can see the group policy does not do much except clean the desktop and run a login script.

Scripting Magic

The following script runs an application and logs you off when the application closes. It also deletes a few pesky icons if they appear on the start menu.

The script enumerates all the running instances of **app.exe** if none of those instances belongs to the logged on user it logs off the session. If the WMI syntax looks complex, don't worry. All you have to do to use this script is change the application path and the application name.

```
On Error Resume Next
Set fs = CreateObject ("Scripting.FileSystemObject")
Set WshShell = WScript.CreateObject ("WScript.Shell")

'Get the username and profile directory
MUser = WshShell.ExpandEnvironmentStrings ("%USERNAME%")
MUserProfile = wshShell.ExpandEnvironmentStrings("%USERPROFILE%")

'Delete icons
fs.DeleteFolder MUserProfile & "\Start Menu\Programs\Accessories",True
fs.DeleteFile  MUserProfile & "\Start Menu\Programs\*.lnk"

'Run the app

wshShell.Run "c:\myapp\app.exe"

' Connect to wmi
set objWMIService = GetObject("winmgmts:root\cimv2")
Do
  found = false
' List the processes
strQuery = "Select * from win32_process where name='app.exe'"
set colProcesses = objWMIService.ExecQuery(strQuery)

for each proc in colProcesses

   ' Get the reference class linking processes to sessions to get the session object path
   strQuery = "References of {win32_process.handle='" & proc.handle & "'} where
ResultClass=Win32_SessionProcess"
   set colSessionReferences = objWMIService.ExecQuery(strQuery)

   for each oSessionReference in colSessionReferences
      'Get associators of the session object that are user accounts (linked by win32_loggedonuser)
      strQuery = "Associators of {" & oSessionReference.antecedent & "} where
AssocClass=win32_LoggedOnUser"
      set colUsers = objWMIService.ExecQuery(strQuery,,48)
        for each user in colUsers
         if user.name = MUser then found = true
      next
   next
next
Loop While found = true

'Run the Windows 2003 logoff utility
wshShell.Run "c:\windows\system32\logoff.exe"
```

Conclusion

With some scripting and Group Policy tweaking, serving a single application to users can be simple and cheap. Both mechanisms are flexible and can be used for much more, catering to various needs.