The GPOGUY Group Policy FAQ

Updated December 2019

1. Is it possible to audit changes to GPOs using Windows auditing?

The following article details what is available natively in terms of Group Policy Change Auditing. The bottom line is that you can tell who changed a GPO but you may not be able to tell what changed in that GPO:

http://sdmsoftware.com/group-policy-blog/group-policy-change-auditing-group-policy-blog/understanding-group-policy-change-auditing/

2. What security context do startup scripts run in?

Startup scripts are machine specific and run before a user logs on. As a result they run in the context of the localSystem account on a computer. As localSystem, they have privilege to do just about anything on a Windows system. If a startup script needs to access network resources (e.g. a server share) however, localSystem will not work. Given that, Group Policy will change the security context of the script to that of the machine's computer account, which is a valid AD user (e.g. a machine named WorkstationA will have an account in AD called WorkstationA$–this account is hidden, but is represented by the computer object when you are viewing an AD domain). The machine account, just like any other user account in AD, is a member of the Authenticated Users and can thus access any network resources that a member of this group can. So, if you have a startup script that needs to access server resources, make sure that the computer's machine account has permissions to access those resources.

3. I know that scripts are only processed during foreground processing, but if I make a change to, say, a shutdown script, does the machine have to restart before the new script is picked up?

It's a good question. The answer is no, you don't have to go through a machine restart for the new shutdown script to be picked up. Even though scripts are only run during foreground processing, script policy is in fact processed in the background. In fact, Windows caches script information in the registry during policy processing, so if you add a new shutdown script policy while a machine is running, it will pick up that new script during background processing and will run it during the next shutdown. You can confirm that your workstations have picked up the new shutdown (or startup) script policy by looking under the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\System\Scripts
```

Or by looking in the following location for User Logon or Logoff scripts:

```
HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\System\Scripts
```

4. Can I copy startup/shutdown or logon/logoff scripts to the local GPO across machines in my environment?

Yes, essentially you need to do three things to activate a script within a local GPO:
Copy script file into appropriate directory on local system.
Create or modify scripts.ini file to refer to script
Modify gpt.ini file to ensure local GPO is processed
For example, if you need to deploy a startup script to a number of systems, you

will need to first copy your script file (e.g. .bat, .vbs or other) to %windir%\system32\GroupPolicy\Machine\Scripts\Startup. Next, you'll need to create or edit the scripts.ini file within the %windir%\system32\GroupPolicy\Machine\Scripts folder. The scripts.ini file holds the name of the scripts that get called, and any command-line parameters that get passed. For example, the following scripts.ini file calls a script called startup.vbs with no parameters.

```
[Startup]
0CmdLine=startup.vbs
0Parameters=
```

If you have more than one script to call within the local GPO, then it will follow the first one as follows:

```
[Startup]
0CmdLine=startup.vbs
0Parameters=
1CmdLine=startup2.vbs
1Parameters= /%computername%
```

Once you've copied the script and edited the scripts.ini file, you need to ensure that the version of the local GPO is non-zero, and is incremented to account for the new script. This is a bit trickier because the file that holds the version information–called gpt.ini– will exist already within the local GPO under %windir%\system32\GroupPolicy. You'll need to edit this file, parse the version= line and increment it. If you want to stick with the official incrementing scheme for Group Policy, then you need to increment the version number 1 for each machine-specific change you make (e.g. adding a startup script) and 65536 for each user-specific change (e.g. adding a logon script) you make. For example,

if I have a workstation with a GPT.INI file that has a version number of 0, as shown below:

```
[General]
gPCFunctionalityVersion=2
gPCUserExtensionNames=[{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{0F6B957E-
509E-11D1-A7CC-0000F87571E3}][{A2E30F80-D7DE-11D2-BBDE-
00C04F86AE3B}{FC715823-C5FB-11D1-9EEF-00A0C90347FF}]
Version=0
gPCMachineExtensionNames=[{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{0F6B957D-
509E-11D1-A7CC-0000F87571E3}{53D6AB1D-2488-11D1-A28C-
00C04FB94F17}][{B1BE8D72-6EAC-11D2-A4EA-00C04F79F83A}{53D6AB1D-2488-11D1-
A28C-00C04FB94F17}]
```

Then I'll need to change the value to 1 if I add a startup script, or 65536 if I add a logon script. Once these three changes are made, the script will be enabled on the local GPO for that machine.

5. I would like to deploy an application setup via Software Installation policy but the setup is not an .MSI file. Is there any workaround?

Yes, but there are some limitations involved. There is a way to deploy legacy .exe setups for use in GPO-based software deployment using something called a ZAP (Zero Application Packaging) file. A ZAP file is simply an INI style file with a .zap

extension. However, ZAP files have two significant limitations:
They can only be published per-user–they can not be assigned to computer or user.
They do not benefit from the privilege escalation feature that MSI-based packages do, when deployed via Group Policy. This means that the user who initiates the ZAP-based installation from Add/Remove Programs must have sufficient privilege to install the application
Beyond these two limitations. ZAP-based deployment can be a quick and dirty way to get an application distributed without having to take the time to re-package it into .MSI format. A little known fact is that when this feature was introduced in Windows 2000, Microsoft product team members referred to it as "Crappy ZAW", where ZAW stood for Zero Admin. Windows–an old term from NT 4 days! In any case, here is a sample .zap file created to deploy Winzip 9.0:

```
[Application]

FriendlyName = "Winzip 9.0"

SetupCommand = "\\server\packages\winzip90.exe"

DisplayVersion = 9.0

[Ext]

ZIP=
```

This ZAP file is pretty self explanatory. The FriendlyName key represents what you see in Add/Remove Programs when you go to select the package. The SetupCommand key lists the UNC path to the setup .exe file (Note that this must be a UNC–a drive letter path won't work). The DisplayVersion key is optional and shows the version of the application as it appears in the GPO. Finally, the extension section lists the extensions associated with this application. By using this EXT section, you guarantee that the application setup will automatically be started if the user clicks on a file with that extension (in this example, .zip), even if they don't explicitly run the setup from Add/Remove Programs. This is sort of a poor man's "install on first use" in the absence of a .MSI advertisement. Note that

the command you provide should also include any appropriate switches if you want the installation to run silently. Otherwise, it will just run interactively, requiring the user to answer prompts along the way.

6. Can I change the path of a package in a GPO once the package has been deployed to my clients?

The answer is, there is no supported way to change a package path without impacting clients who have already received the package. This is because the path to the package is stored in a number of places, including in AD, in the .aas file found in the GPT portion of the GPO, and most notably, on the client. If you need to move a package to a new server, the best solution is to try and keep the server name the same (or alias the old server name) or better yet, use DFS for all of your packages so that you can move the package around without changing the path.

7. What happens, behind the scenes, when I use the "Redeploy" button on an application that has been deployed already using GP Software Installation.

For any client machines that have installed the application that was deployed via GPO, when that application is redeployed, the client will essentially reinstall the application during its next foreground processing cycle. For example, if you redeploy an application that was machine assigned, at the next reboot of any client who had installed the application via that GPO, that client will perform a reinstallation of the application. Specifically, what happens is the client is

instructed to do a Windows Installer repair with the options **omusv**. Those options essentially reinstall most (but not all) major pieces of a package. So, keep in mind that when you do a redeploy, all clients that had previously installed that application via the GPO will attempt to reinstall that application. This can sometimes have a big impact on the network if many clients are hitting a package on a server share nearly simultaneously.

8. *I know that policy will normally not be processed on a given machine unless the GPO has changed, but it seems like security policy does not follow this model. Is that correct?*

Yes, in fact security policy is one of those anomalies with respect to the "Don't process if the GPO hasn't changed" rules. By default, security policy (which is defined as policy found in Computer Configuration|Windows Settings|Security Settings) will process every 16 hours in the background, even if the GPO hasn't changed. This ensures that, for a critical area like security configuration, if the user has made a change on the local system that contravenes policy, that change will be undone on a periodic basis. You can actually modify this background refresh interval by editing the following registry value:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon\GPExtensions\{827D319E-6EAC-11D2-A4EA-
00C04F79F83A}\MaxNoGPOListChangesInterval
```

This value is stored as a hexadecimal number that represents the number of minutes between background refreshes.

9. I have some machines that are not processing security policy while others in the same OU are working fine. What is the problem?

When security policy is processed, Windows uses the secedit security configuration engine to process Group Policy-based security policy. Part of this processing relies on using a local security database, found on each Windows system, called secedit.sdb. This file is found, by default, in c:\windows\security\database. Occasionally this database can get corrupted and prevent security policy from applying on that machine. You can check for this by running the following command:

```
esentutl /g c:\windows\security\database\secedit.sdb
```

If the command finds errors you can use the esentutl utility's /p option to attempt to repair the file.

10. How can launch the GP Editor against a domain-based GPO from the command line? For example, I can edit the local GPO by typing gpedit.msc. Is there a way to do that against an AD-based GPO?

In fact, there is. You can launch the GP Editor directly against a domain-based GPO from the command line using the following format:

```
gpedit.msc /gpobject:"LDAP://CN={GUID of the
GPO},CN=Policies,CN=System,DC=<domain>
```

For example, if my domain name is test.com and the GPO I wish to edit is the Default Domain Policy, I can use the following command:

```
gpedit.msc /gpobject:"LDAP://CN={31B2F340-016D-11D2-945F-
00C04FB984F9},CN=Policies,CN=System,DC=test,DC=com
```

11. How do I form a proper WMI Filter?

A WMI filter takes the form of a WMI Query, using the WMI Query Language (WQL). However, you must preface your query with the name of the WMI namespace you will be accessing. For example, if I wanted to query for all machines running XP, SP2, my WMI Filter would look like this:

```
root\cimv2;Select * FROM Win32_OperatingSystem WHERE Build= 2600 AND
CSDVersion = "Service Pack 2"
```

In this filter, I'm starting off by telling Windows that I want to perform a query against a WMI class that resides in the root\cimv2 namespace. After following that with a semicolon, I enter my WQL query. In the example above, I'm selecting all instances of the Win32_OperatingSystem class (there is only one on a given Windows system) whose properties Build and CSDVersion equal 2600 (the build # for XP) and "Service Pack 2". The best tool I've found for checking out the name of various WMI classes and their properties is WMIX by GoverLan,

Note that my WMI Filter Validation Tool on the SDM Software Freeware page lets you view, print and validate WMI filters against systems in your environment.

12. Why are GP settings not removed from my machine when I remove it from the domain?

When you remove the machine from the domain, the GP settings remain and, since it's no longer in the domain, only the local GPO will process. That means that unless the local GPO overrides those domain-based settings, they will remain "tattooed" on your system indefinitely. The best way to handle that is to either move the machine outside of the scope of GP settings before removing it from the domain and let it process the removal of policies, or use the local GPO editor to override the settings on the machine as it exists right now.

13. What is Loopback Policy used for?

Loopback policy is a special mode of policy processing that allows you to control what user policy a user receives based on the machine they are logging onto rather than their user account. It is used most often in Remote Desktop Services or "Kiosk" environments, since in both of these situations you want to tightly control the user lockdown of any users logging onto these special-use machines. Loopback is configured on the designated computers that your users will be logging onto. You can enable Loopback by setting the computer-based policy found at Computer Configuration\Administrative Templates\System\Group Policy\User Group Policy loopback processing mode. Loopback comes in two flavors–merge mode and replace mode. These different modes are described in the next question.

14. I am using Loopback processing to control user policy settings on a group of machines. When I run gpresult, or the Group Policy results wizard on those machines, it shows me that some GPOs are processing twice, and I even have some logon scripts running twice. What's going on?

The duplicate GPO processing is a function of using Loopback policy in merge mode only (replace mode doesn't cause this). What is going on is, with replace mode, Windows basically says, don't do any user-specific policy processing for the user logging into a loopback machine. So basically any GPOs that would normally be processed by the user, including local, site, domain and OU- linked ones, are just not processed in replace mode. Instead, all user settings come from any GPOs that apply to the loopback computer, including those linked at the local, site, domain and OU level. By contrast, merge mode says, first process all user GPOs that the user account would normally get. Then, process all user GPOs that the loopback computer would normally get. So, what that means is that policies that are higher in the hierarchy, like site and domain-linked GPOs that are processed both by the computer and the user, get processed twice. Since the computer-based loopback user settings process last, the result would normally be that any conflicting user-specific settings (like Admin. Template registry settings) would be overridden by the loopback computer settings. And that happens, however, certain policy extensions, like scripts or software installation, don't exhibit override behavior. If two scripts are in the path to be processed, they will process cumulatively rather than one overriding the other. Hence the reason you see logon scripts running twice.

15. What is the difference between synchronous and asynchronous policy processing?

These terms refer to whether or not Group Policy processing occurs while other things are happening in Windows. For example, if computer-specific foreground processing is set to run asynchronously (as is the case by default in Windows 10) then as Windows initially boots up, it will not wait for GP processing to finish before presenting a user logon dialog. Similarly, if user-specific foreground processing is set to run asynchronously, Windows will not wait for GP processing to finish before presenting the user's desktop. Asynchronous can speed up the startup and logon process but can also result in certain policy extensions (e.g. Folder Redirection, Software Installation) require two or more foreground processing events to take effect. A foreground processing event is a computer startup or user logon. When foreground processing is set to synchronous, then Windows waits for computer or user GP processing to complete before presenting the user a logon dialog or a desktop, respectively.

16. I have clients processing GP separated by a firewall from my domain controllers. That firewall blocks most ports between client and server and so Group Policy is failing. What protocols does GP require and how do I restrict the number of ports I have to open up?

Group Policy processing requires the following protocols and ports to be open between client and Domain Controller:

Application	Protocol	Ports Needed
DCOM	TCP/UDP	random port numbers between 1024 – 65534
ICMP (ping)	ICMP	
LDAP	TCP	389
SMB	TCP	445
RPC	TCP	135 and random port number btw. 1024 – 65534

You can configure RPC/DCOM to use a restricted set of high-level (>1024) by making changes to a client's registry.

17. Where can I get a list of ADMX files that Microsoft provides for its products?

The following site maintains a pretty good list:

http://social.technet.microsoft.com/wiki/contents/articles/4976.aspx.