

Documentation for the EZ GPO Power Management tool for Network Administrators

- [Background](#)
- [Installation](#)
 - [Server Installation - ADM Template](#)
 - [Client Installation - MSI Installer](#)
- [Upgrade](#)
- [Configuration](#)
 - [Base Options](#)
 - [Options](#)
 - [Environmental Variables for use in shell expansion of LogFile entry](#)
 - [Simple Scheme](#)
- [Non AD Installations](#)
- [Testing and verification](#)
- [Technical Reference](#)
 - [Registry Entries](#)

Background

EZ_GPO is a tool for network administrators who manage Windows client workstations using Group Policy Objects under Active Directory, Novell NDS/Zenworks or any other client registry management system. It gives centralized control to network administrators over the user's power management settings. Due to the designs of Microsoft's implementation of power management, this is impossible using normal techniques.

Although implementing MPM through GPOs should be straight forward since MPM uses the registry to store settings, the reality is far different. Using Microsoft's Native GPO feature in Active Directory to manage power management directly is not possible, because there is a limitation in the Administrative Templates Meta language (ADM) used to create the interface for custom Group Policies. Only two types of keys can be changed through the ADM interface to the GPO tree: single value strings and dword (integer) values. Binary keys are the storage of choice for power management settings and this limitation is one reason why an interface for handling power management settings is not provided. Additionally, changing this binary key using the same copy for each computer on the network will cause problems on heterogeneous networks with multiple Windows versions, as the binary strings used to store the power management settings are OS and machine dependent. This is despite being housed in the user portion of the registry and therefore being user based.

It is with this knowledge that EZ GPO was created to help administrators work around this in the most unobtrusive manner possible. EZ GPO is available for download at (<http://www.terranovum.com/projects/energystar/>) and is open source under a BSD style license.

The use of this license style allows all stakeholders to be equal beneficiaries of the knowledge and efforts of this open source project.

Basically, how this application works is there is a small client install of a binary application (user executed) and a service application (machine system account executed). The service is a standard Microsoft Windows service and controls the power management settings of the machine when users are not logged in. The user executed binary is executed via the local machine run entry by each user to login to the client machine at login time and is responsible for all changes to the user's portion of the registry. Both of these applications look for GPOs that are set by AD via the enclosed ADM template, alternatively (ex: if GPO's are not found), they look for registry entries set in the software branch (LM and CU respectively) that can be set using zenworks, or via any method that allows an admin to set client registry entries remotely.

These GPOs/Registry Entries are in integer and string value format and then, using Microsoft's core API, the appropriate changes to the PM settings, which will be available immediately thereafter. There is no threat of a race condition (ie; the binary executing before the policies are set) since GPOs are applied immediately after authentication whilst the run list is processed well after user services have initialized and are running.

Installation

ADM Template

Install the ADM GPO template located at "server GPO/EZ_GPO.adm" by opening the "Active Directory Users and Computers" MMC located in Administrative tools. Right click on the domain you want to manage and click properties. Click the "Group Policy" tab and highlight the "Default Domain Policy". Click edit and a new MMC will come up with two sub headings of "Computer" and "User". Expand the "User" hive and then under that expand the "Administrative Templates" hive. Right click on that hive and click "Add/Remove Templates". Click Add and navigate to where you unzipped/installed the EZ_GPO tool. The ADM file is under the "server GPO" directory. Highlight the file and click open. This will load the ADM template under the "Administrative Templates" hive under the name "EZ GPO by the Environmental Protection Agency". You will have three group policies to work with. **NB:** On Windows Server 2003, the default behavior has changed and the snapin will no longer copy the ADM file to the %SystemRoot%\System32\Inf directory when you import it so you will have to copy it there yourself before loading the template into the snapin although this is not required. You may have an alternate location for your imported ADM files.

Binary Executable

Installing the binary (client installer/EZ GPO Installer.msi) is straight forward and leaves a great deal of latitude for the administrator. The easiest option is to install it using a machine based (**NB:** not a user based) software installation policy. This also will allow you to cleanly manage upgrades through time easily. Please note that if using AD's Software Installation, that for

Windows XP clients, the client machines will require a few restarts before the software is fully installed. This is due to Fast Logon Optimization and Asynchronous policy refresh. For more info on that please see <http://support.microsoft.com/default.aspx?scid=kb;en-us;305293&Product=winxp> and http://msdn.microsoft.com/library/default.asp?url=/library/en-us/policy/policy/logon_optimization.asp

Upgrade

To upgrade first remove the ADM file from the list of available adm templates in the user portion of the GPO and then apply the changes. Then add in the new updated ADM file It should pick up your old settings but still pick up the new version numbers. Update the MSI installer according to you normal procedures.

Configuration

Base Options

The first group policy is the "Base Options" GPO. This is where mandatory configuration information is stored. The GPO Tool will not run without this information. Enable this and in most cases the default will work just fine.

- Control
- Settings Scheme - Dropdown to choose which scheme to edit. This relates to a GPO by the same name.
- Major Version
- Minor Version

The Control variable is simply a flag that allows the client based tools to determine if there is a valid GPO set or should they look to the alternate location. This should be left alone.

The settings Scheme is to tell the GPO Tool what type of configuration scheme you would like to configure. Currently there is just the "Simple" scheme which does six predetermined settings. This option will be the source of future expansion that will allow for more complex configuration options. This choice you make here relates to a GPO by the same name.

Major and Minor versions allow for ADM template changes to occur while protecting older clients from misinterpreting the options. In most cases when an ADM template is changed, that means that there is either a bug fix in the ADM template (a minor change) or a new version of the tool (a major change). In the latter case all clients should theoretically be upgraded but there is always a chance a few will get through the system. This protects those failed upgrades from getting mangled. This versioning feature will undoubtedly be put to other uses in the future.

Options

This GPO is for all optional configuration settings. Enable this GPO if any options are needed.

- SecurityBypass - A flag to get around the hard coded user rights issue in the PM library.(see below for more) **NB:** This setting is only found under the User Policy and not in the Computer Policy as the computer policy is set using the default system account.
- ForceStandby - A flag that allows an administrator to control whether only S3 capable machines get system standby or all machines. Good for non-Intel based chip sets.
- Log - A flag to turn on whether logging is enabled or not. **(Unimplemented)**
- LogLevel - Currently a range from 0-10 where 10 is the most verbose.**(Unimplemented)**
- LogFile - The file to log into. **(Unimplemented)**
- EventMessageFile **(Unimplemented)**
- LogServer **(Unimplemented)**

SecurityBypass is what the name implies. A way to bypass the security MS put on users of type "User" from being able to change their power management settings. There is some background needed to understand why this option is needed. In short, the power management subsystem is designed radically different from the rest of the operating system. It's settings are stored in binary strings which are machine and OS dependent despite half of those settings being stored in the user portion of the registry. Those binary strings are literally C data structures simply dumped from memory into the registry for ease of retrieval.

Most subsystems in the Windows OS tend to be designed in a way that separate the machine and user dependent settings into their respective portions of the registry. As well, most systems employ numerical and string based config info which is addressable via administrative tools like AD's group policy objects and allow administrators to determine to a certain resolution the ability for some users and not others to edit these settings. In the case of PM, the system is hard coded to deny users of type "User" and "Guest" the ability to edit the power management settings. This would not be a bad thing per se if the admin had a method to manage the settings remotely, or at least locally for users other than themselves. This ability is not provided however. The EZ_GPO tool has been designed to use the Windows API where ever possible but since there is a restriction hard coded in the API against the above named user groups, that is not possible.

However, since the registry keys for these settings are located in the user's hive (HKEY_CURRENT_USER) they are still writeable by any user who can access the key via regedit, etc. Since the EZ_GPO tool runs under the user's ID, this is used as a back door to make the changes by bypassing the API and reading the setting directly out of the registry directly into memory, making the changes and then writing the info back down into the registry. This is only done when the bypass is enabled and the user lacks the proper rights to make the changes through the API. Currently this is exactly how it is done through the API

except for the fact that EZ_GPO loads the C data structure directly from the registry instead of having it passed by pointer from an API call that directly reads it from the registry. There is only one known issue with this and that is the, under certain circumstances, settings do not show up as changed in the power management control panel, despite the fact that they are changed and active, until the system is rebooted. The known fixes for this are being evaluated and one will be implemented in a later version.

Environmental Variables for use in shell expansion of LogFile entry.

Unimplemented

- %AppData%
- %UserProfile%
- %AllUsersProfile%
- %ProgramFiles%
- %SystemRoot%
- %SystemDrive%
- %Temp%
- %Tmp%

Simple Scheme

- AC User Monitor Timeout - Time until monitor off occurs in minutes
- AC User System Standby Timeout - Time until System Standby occurs in minutes
- AC Machine Hibernation Timeout - Time until hibernation occurs in minutes (NB: should be 0 or at least one minute larger than system standby and requires that hibernation be enabled)
- DC User Monitor Timeout - Time until monitor off occurs in minutes
- DC User System Standby Timeout - Time until System Standby occurs in minutes
- DC Machine Hibernation Timeout - Time until hibernation occurs in minutes (NB: should be 0 or at least one minute larger than system standby and requires that hibernation be enabled)

The simple scheme is currently the only scheme implemented in EZ_GPO. It contains six settings, each of which affects either the AC (plugged in settings) or the DC (when on battery settings), and are expressed in minutes. To set any setting to never, input a value of 0 (zero). Please note that the tool is limited in what clients it will set system standby for by default. The limitation revolves around the presence of a new version of power management named ACPI (more specifically the support for the S3 sleep state). Most older hardware had Advanced Power Management v2 (APM2) and recent Pentium 3 and early Pentium 4 machines had a flavor of ACPI that was not fully implemented. Most Pentium 4s and higher machines these days have full hardware support for ACPI and the S3 sleep state. This behavior can be overridden by employing the ForceStandby option in the above options. This can be useful for machines that are AMD based as they support a form of S3 but do not always show it consistently.

To use the hibernation settings, hibernation must first be enabled to allow this setting to take effect. There is unfortunately no way that MS provides, in the form of a (published) API call, to make this happen. If they did, it would have been made an option. The only way for admins to enable this is to do so before deployment, via the source image.

Non AD Installations

The tool will alternately look in {HKCU | HKLM}\Software\TerraNovum\EZ_GPO ("Policies" missing) if it does not find something in the default location of {HKCU | HKLM}\Software\Policies\TerraNovum\EZ_GPO. This is to assist non AD networks to use the tool with something like Zenworks, et al and not run into permissions problems, depending on how they try to set the registry entries. For a complete list of registry entries, consult below and the ADM template.

Notes on Testing

A few notes on testing. First and foremost, DO NOT test this software on your admin workstation. For a variety of logical reasons, testing on a live box is bad. In addition, IT admins, being cut from pretty similar cloth, tend to have software installed on their workstations that is not typical of the organization as a whole. Much of this software is server based such as database apps, and other applications written and tested in 24x7 server environments and not part-time client workstations environments. Power management was designed to be a workstation/mobile solution.

There is no way to confirm, without waiting by the clock, the application of the PM settings on the machine in a logged off state. Microsoft does not provide any facilities for this. When confirming the application of the settings to users, you should restart the machine before opening the power management control panel app. When you do open the app, if you find drop downs that are blank, it likely worked. This is because if you picked a number other than what exists in the list of possible selection in the drop downs, it shows the idle time as blank. The only numbers that will show in the drop down are 1-5, 10, 15, 20, 30, 40...

There is a binary in the testing folder of the distribution that can be used to determine what options are needed for a particular user/machine.

Lastly, there is a bug in Microsoft's implementation of power management. There is an interaction with recently logged off users where their power management policy will be picked up by the machine when no others are logged off. To fix this, simply reboot. The machine's policy will be restored properly.

Technical Reference

Registry Entries

Software\Policies\TerraNovum\EZ_GPO

Software\Policies\TerraNovum\EZ_GPO\MajorVersion = dword (2)

Software\Policies\TerraNovum\EZ_GPO\MinorVersion = dword (0)

Software\Policies\TerraNovum\EZ_GPO\SettingsScheme = string (Simple,)

Software\Policies\TerraNovum\EZ_GPO\Control = string (Verify)

Software\Policies\TerraNovum\EZ_GPO\Options

Software\Policies\TerraNovum\EZ_GPO\Options\SecurityBypass = dword (1 or 0)

Software\Policies\TerraNovum\EZ_GPO\Options\ForceStandby = dword (1 or 0)

Software\Policies\TerraNovum\EZ_GPO\Options\Log = dword (1 or 0) (**Unimplemented**)

Software\Policies\TerraNovum\EZ_GPO\Options\LogLevel = dword (1 - 10) (**Unimplemented**)

Software\Policies\TerraNovum\EZ_GPO\Options\LogFile = ex_string

("%UserProfile%\Logfile.dat") (**Unimplemented**)

Software\Policies\TerraNovum\EZ_GPO\Options\EventMessageFile = ex_string

("%System%\config\PMEvents.dll") (**Unimplemented**)

Software\Policies\TerraNovum\EZ_GPO\Options\LogServer = string (localhost)

(**Unimplemented**)

Software\Policies\TerraNovum\EZ_GPO\{Scheme Name }

The Simple scheme

Software\Policies\TerraNovum\EZ_GPO\Simple

Software\Policies\TerraNovum\EZ_GPO\Simple\ACUserMonIdleTime = dword (0) in minutes

Software\Policies\TerraNovum\EZ_GPO\Simple\ACUserStandByIdleTime = dword (0) in minutes

Software\Policies\TerraNovum\EZ_GPO\Simple\ACMachStandByIdleTime = dword (0) in minutes (Hibernation)

Software\Policies\TerraNovum\EZ_GPO\Simple\DCUserMonIdleTime = dword (0) in minutes

Software\Policies\TerraNovum\EZ_GPO\Simple\DCUserStandByIdleTime = dword (0) in minutes

Software\Policies\TerraNovum\EZ_GPO\Simple\DCMachStandByIdleTime = dword (0) in minutes (Hibernation)

Software\TerraNovum\EZ_GPO\Backup

Varied and program generated.