

Faronics

DEEPFREEZE™

ABSOLUTE Workstation Integrity



Deep Freeze Enterprise - Patch Management

TECHNICAL WHITEPAPER

Last modified: March, 2007

Faronics

Toll Free Tel: 800-943-6422

Toll Free Fax: 800-943-6488

International Tel: +1 604-637-3333

International Fax: +1 604-637-8188

www.faronics.com

©1999-2007 Faronics Corporation. All rights reserved.

Deep Freeze, Anti-Executable, and WINSelect are trademarks and/or registered trademarks of Faronics Corporation.

All other company and product names are trademarks of their respective owners.

Contents

Introduction.....	3
Scheduled Patch Maintenance.....	3
Scheduling Windows Updates.....	3
Scheduling Windows Updates Through the Maintenance Option.....	4
Scheduling Windows Updates through Group Policy	5
Deep Freeze & Group Policy – Default Behavior.....	5
Deep Freeze & Group Policy – Recommended Configuration.....	6
Alternate Configuration Option - Group Policy Refresh	6
Scheduling Antivirus Updates.....	8
Scheduling Additional Program Updates	8
Logon Patch Maintenance	9
Logon Patch Maintenance Theory	9
Logon Patch Maintenance Example	10
Creating the Update Script.....	10
Creating the Group Policy	16
Modifying the Group Policy	17
Enforcing the Group Policy	17
Real Time Patch Maintenance	18
Disabling Deep Freeze Locally	18
Disabling Deep Freeze Through the Enterprise Console	18
Disabling Deep Freeze Through the Command Line Control.....	18
Configuring Software to Update in a Thawed Location	18
Appendix A - Deep Freeze and SUS/WSUS FAQ	19
Appendix B - Deep Freeze Update Script.....	20
Appendix C - Common Update Scenarios	24
Scenario 1 - Updating Clients in a Dynamic Update Environment.....	24
Scenario 2 - Updating in a 24-Hour Lab Environment	24
Scenario 3 - Updating in a Mobile Environment.....	24

Introduction

A major concern for all systems administrators is maintaining the security of their workstations. With new exploits and vulnerabilities being found all the time, a proper patch management strategy is critical to ensure the health and security of workstation deployment.

Deep Freeze allows systems administrators to ensure the integrity of their workstations against exploits — even ones that have yet to be discovered. However, it introduces some interesting challenges within the process of applying patches because Deep Freeze does not discriminate — it removes both the good and the bad changes and returns the workstation to its original, pristine state on every restart.

There are several methods for integrating Deep Freeze with Patch Management, and when properly done, users can enjoy the bulletproof reliability of a Deep Freeze protected system, and system administrators can have the peace of mind that comes from knowing their systems are fully up to date.

This white paper discusses the different methods available to update software in a Deep Freeze environment.

Scheduled Patch Maintenance

Scheduled patch maintenance allows the administrator to specify a period of time when the client machines restart with Deep Freeze in a Thawed state. During this maintenance period, software updates, Windows Updates, and antivirus definition updates can be scheduled. Scripts can be run and batch files can be executed. This is a very common method to keep client machines up to date with the most recent patches.

Scheduled patch maintenance is very popular in lab situations. During certain times on certain days of the week, labs are not in use. A maintenance period can be scheduled to run updates during these times.

The maintenance period is configured using the Deep Freeze Configuration Administrator. This program is used to configure workstation installation files, as well as configuration files. The configuration files are used to apply the changes to deployed workstations through the Deep Freeze Enterprise Console.

Depending on the policies in place, certain updates may need to be run. Windows and antivirus updates tend to be the most frequent. The following information explains some of the update scenarios encountered and the different methods available to handle these updates.

Scheduling Windows Updates

The following information describes the steps required in order to set up Deep Freeze to work with Windows Updates.

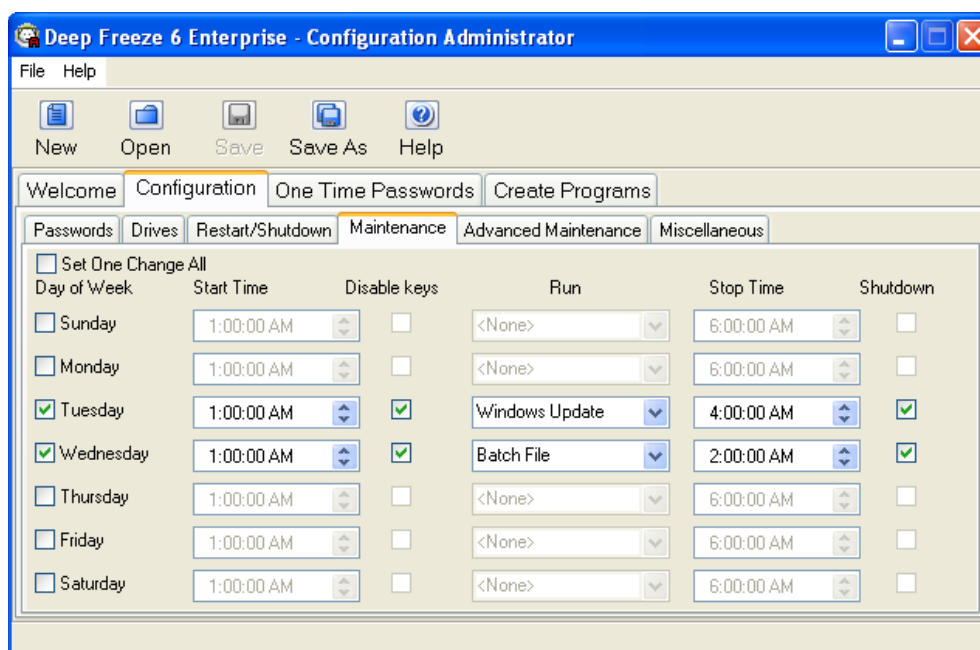
There are several different methods available to run a Windows Update in a Frozen environment. Deep Freeze can be set up to start a Windows Update at a particular point in time during the maintenance period. Deep Freeze can also be set up to execute a batch file at a particular point in time during the maintenance period, which would start the Windows Update. Finally, another program could start the updates during the Deep Freeze maintenance period.

Scheduling Windows Updates Through the Maintenance Option

The first method involves setting up a maintenance period using the Deep Freeze Configuration Administrator. An option is selected so Deep Freeze will run the Windows Updates after the machine goes into maintenance mode.

Complete the following steps to configure a maintenance period:

1. In the Deep Freeze Configuration Administrator, click the *Configuration* tab and click the *Maintenance* sub tab.
2. Specify days and times the maintenance mode will occur. The window should look similar to the following:

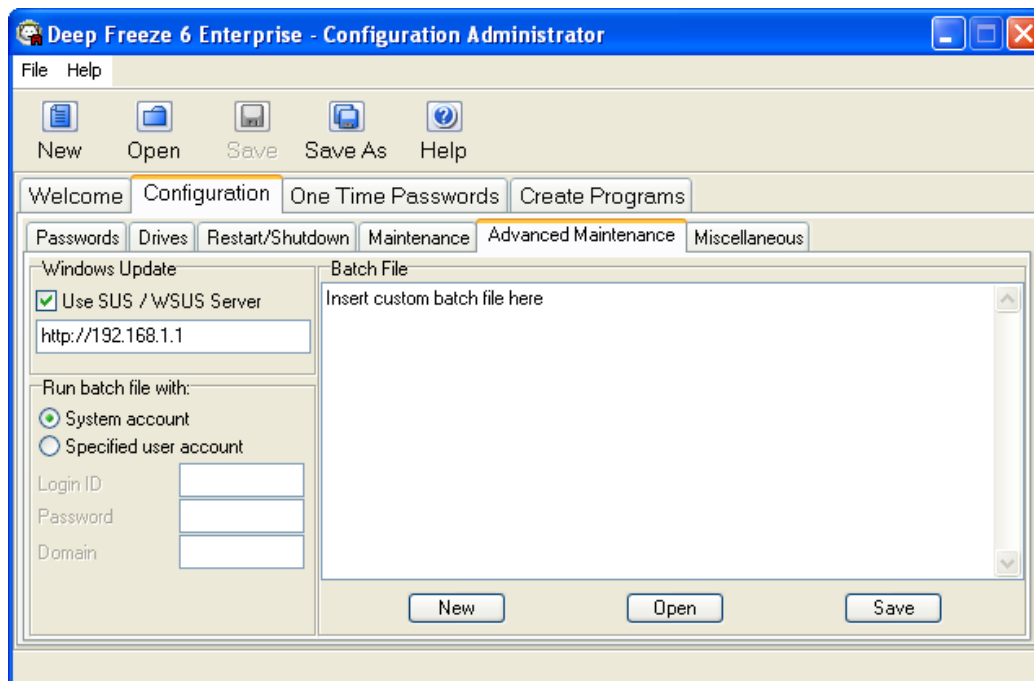


In the above screen, Tuesday has been selected for maintenance with the *Windows Update* option. At 1AM, the workstation enters a Thawed state. At 4AM, the workstation returns to a Frozen state. The *Disable keys* checkbox has been checked; this means that the keyboard and mouse are locked while the machine is in maintenance mode.

If a client machine is configured with this option, it would attempt to download and run updates from Microsoft's web site during the maintenance period.

If there is an SUS or WSUS server, this can be specified using the following steps:

1. Click the *Configuration* tab and the *Advanced Maintenance* sub tab.
2. Check *Use SUS/WSUS Server* and enter the IP address of fully qualified domain name of the server. The screen should look similar to the following:



The client machine with these settings would attempt to download and run updates from the specified SUS/WSUS server rather than from Microsoft's web site.

Scheduling Windows Updates through Group Policy

Although settings for Windows update can be administered through Deep freeze many administrators prefer to control settings for automatic updates through the use of Group Policy in a domain environment. Below we will explore how Deep Freeze will interact with Group Policy, specifically the update settings, and how to configure both to ensure that updates are applied to the workstations while the computer is in a thawed state.

Deep Freeze & Group Policy – Default Behavior

Deep Freeze manages Windows updates by manipulating a series of registry keys responsible for the configuration of the Automatic Update client on the local workstation. This is the same set of keys defined by the Windows Update group policy objects on the server. To ensure that settings that may have been configured when Deep Freeze is installed are not lost Deep Freeze backs up the settings that are pre-existing on the workstation and stores them in a hidden location on the local computer, these settings are restored by Deep Freeze each time that the workstation reboots, regardless of if the system is in the frozen or in the thawed state. This has the side effect of rendering any change to these settings, either manually or via policy, lost upon any reboot.

Due to this behavior administrators may see that policy settings that are being refreshed and applied to the workstation are being lost when the workstations reboot, and in environments where Windows update is being controlled through Group Policy this will result in unpredictable results with regards to the installation of policies on the local workstation.

To allow administrators to control Windows Update through the use of Group Policies an option has been provided to configure this default behavior so that Deep Freeze will not configure the automatic updates registry keys, and will not reset these values when the workstation is rebooted. The option is titled “Control Windows Update” and is found on the Configuration → Miscellaneous tab of the Deep Freeze Configuration Administrator in version 6.x of the product, shown below. Removing the checkbox from this option will prevent Deep Freeze from modifying the settings for Windows Update as set by policy. This option is available in Deep Freeze 5.7; however it is located on the Configuration → Maintenance tab of the 5.7 version of Deep Freeze Enterprise.

Deep Freeze & Group Policy – Recommended Configuration

If administrators wish to control the update policy strictly through the use of Group Policy objects it is recommended that the option to control windows updates in Deep Freeze be disabled, and that the following group policy settings be set with regards to Windows Update.

Computer Configuration → Administrative Templates → Windows Update:

- Configure Automatic Updates: Enabled
- Configure automatic Updating: Option #4
- Scheduled install day: 0 if maintenance will happen every day, the specific day if the maintenance will only occur once a week.
- Scheduled install time: Set to 30 min after the start of the maintenance window.
- Do not display ‘Install Updates and Shut Down’ option in Shutdown Windows dialog box: Enabled
- Do not adjust default option to ‘Install Updates and Shut Down’ in Shut Down Windows dialog box: Enabled
- Reschedule Automatic Updates scheduled installations: Disabled
- No auto-restart for scheduled Automatic Updates installations: Disabled

This policy will ensure that updates are installed during the maintenance window, and that any updates that are downloaded but not installed will not attempt to continually reinstall on the client workstation while frozen.

Administrators will need to ensure that the maintenance window configured in Deep Freeze is large enough to complete the download and installation of the updates from whatever source is configured, and that computers are either left on to enter the maintenance window, or are woken from a sleep state prior to the start of the maintenance window. It is further recommended that these settings be applied as a specific Group Policy Object specific to protected systems that will override any other automatic update policy that may be defined on the domain.

Alternate Configuration Option - Group Policy Refresh

An alternate configuration option available to Administrators is to force a refresh of the workstations policy settings each time the computer loads, this in most cases will override the settings put in place by Deep Freeze and will ensure that the administrator set policy is enforced.

This method of administering updates is not recommended simply because it can result in updates being installed on the systems while the workstation is in the frozen state if updates are downloaded and stored on the workstation without being installed. On subsequent reboots the computer may continually try to install the updates, losing the installed copy upon reboot.

If this method is pursued administrators should ensure that the workstations are configured with a long enough maintenance window to ensure the successful installation of all Windows Updates without interruption.

The following steps can be used to call a Group Policy update during the maintenance period:

1. In the Deep Freeze Configuration Administrator, click the *Configuration* tab and click the *Maintenance* sub tab.
2. Specify days and times the maintenance mode will occur.
3. Select *Batch File* from the *Run* drop-down.
4. Click the *Advanced Maintenance* sub tab.
5. Enter the following text into the *Batch File* text box: `gpupdate /force`
6. Check *Specified user account* under *Run batch file with*.
7. Enter a *Login ID*, *Password*, and *Domain*. You must use an account with the proper permissions to run a Windows Update on the local machine .



Deep Freeze does not prevent other domain policies from applying.

Scheduling Antivirus Updates

There are several different methods available to run antivirus updates depending on the antivirus solutions being used. The following are links to white papers for several of the most common solutions. These white papers explain several methods that can be used. Any of these white papers explain concepts that may be used with other solutions not listed here.

Computer Associates eTrust Anti-Virus

http://www.faronics.com/whitepapers/DFEnt_CAETrust.pdf

McAfee eTrust Orchestrator

http://www.faronics.com/whitepapers/DFEnt_McAfeeEPO.pdf

Sophos Anti-Virus

http://www.faronics.com/whitepapers/DFEnt_SophosAntivirus.pdf

Symantec Anti-Virus Corporate Edition

http://www.faronics.com/whitepapers/DFEnt_SymantecAntivirus.pdf

Trend Micro OfficeScan

http://www.faronics.com/whitepapers/DFEnt_TrendOfficeScan.pdf

Panda BusinessSecure Antivirus

http://www.faronics.com/whitepapers/DFEnt_PandaAntivirus.pdf

Scheduling Additional Program Updates

The concepts outlined for the antivirus definition updates can also be applied to updating other applications. However, not all methods described may work with a particular application. Refer to the above antivirus white papers for suggested methods, or the white paper entitled *Retaining User Data* at the following location: http://www.faronics.com/whitepapers/DF_RetainUserData.pdf

Logon Patch Maintenance

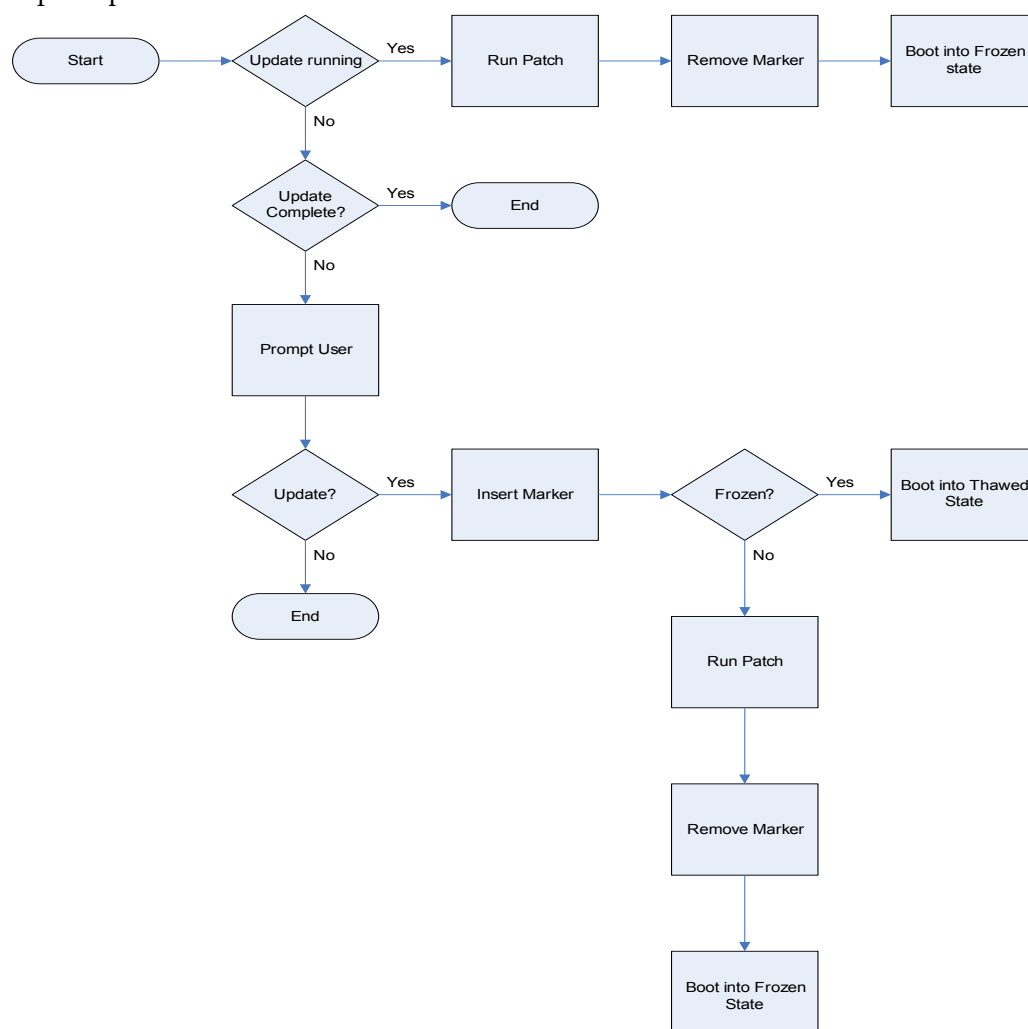
This option allows the administrator to install updates to the client machine when a certain user logs on. In an Active Directory environment, a logon script can be executed to update the client machine. Using the Deep Freeze Command Line Control, Deep Freeze can be disabled before the updates are run and re-enabled afterwards.

Logon patch maintenance is quite popular in a mobile environment where users are working with laptops and are often on the road. When they get back to the office and login, a script is executed to determine if the users require any updates. If they do, the users are prompted as to whether they would like to run the updates. If they agree, Deep Freeze is disabled, the updates are run, and Deep Freeze is re-enabled.

The example below assumes that the person implementing the script is familiar with Group Policy, Active Directory, and Visual Basic Scripting.

Logon Patch Maintenance Theory

This concept deals with updating a Frozen machine when the user logs on. With some slight modifications, the same theory can be applied to an environment where patches are scheduled or performed in real time. The following flowchart outlines the required steps, depending on the state of the update process:



Because the machine boots several times, the script needs to check a value to see what phase of the script is currently running. Because the machine will be Frozen at times, a value cannot be stored in the Frozen partition. This means the value must be stored either on the network or in a Thawed partition on the machine.

It is also important to understand that the above flowchart is a very simple model. In a real-world example, the flowchart would most likely have additional steps to disable the keyboard and mouse and check for the current version of the patch to run. Those steps are beyond the scope of this white paper.

Logon Patch Maintenance Example

The following example uses an Active Directory environment to call a script file when a user logs on. The following section describes how to create a script based on the earlier flowchart and implement Group Policy to call this script when a user logs on. A full version of the script can be downloaded from the following location:

http://www.faronics.com/exe/DFEnt_ADUpdateScript.zip

Creating the Update Script

This script checks to see if the machine requires updates. If the machine requires an update, it prompts the user. If the user selects Yes, the machine is put into a Thawed state. At this point, the patch is applied and the machine is returned to a Frozen state.

Use the following steps to create the script file one section at a time:

The script file can be created using many different editors. In this case, Notepad is used.

1. Open Notepad and enter the following text to create the global assemblies:

```
' ***** GLOBAL ASSEMBLIES *****  
Set objNet = CreateObject("WScript.Network")
```

This code segment creates an object called *objNet* used throughout the script.

2. Enter the following text to create the global variables:

```
' ***** GLOBAL VARIABLES *****  
strUNCPath = "\\FarDemo.local\NETLOGON\  
strMarkerFile = objNet.ComputerName & ".mar"  
strMarkerCompleteFile = "COMPLETED-" & objNet.ComputerName & ".fin"
```

strUNCPath is a variable that maps to a server. Modify the path to match that of the server being used. This is where the marker files are created. The Marker files are used to determine whether the machine requires an update and whether the update is completed.

strMarkerFile is a variable holding the name of the marker file used to indicate whether an update is running. Each marker file has the unique name equal to the machine the update is running on.

strMarkerCompleteFile is a variable holding the name of the file to indicate if the patch has been run. If this file exists, the update has been run and is not required to run again.

3. Enter the following text to create the main routine:

```
' ***** MAIN *****
' Calls all of the other routines...
If UpdateRunning = True Then
    RunPatch
    RemoveMarker
    BootFrozen
Else
    If UpdateComplete = False Then
        If UserPatchPrompt = True Then
            InsertMarker
            If Frozen = True Then
                BootThawed
            Else
                RunPatch
                RemoveMarker
                BootFrozen
            End If
        Else
            ' Exit Script
        End If
    Else
        ' Exit Script
    End If
End If
```

The main routine follows the structure of the flowchart. It calls the other routines as required.

4. Enter the following text to create the *UpdateRunning* function:

```
' ***** UPDATE RUNNING? *****
' Check for marker file. If exists, the update is running. Return True.
Function UpdateRunning
    Set objFS = CreateObject("Scripting.FileSystemObject")
    Set objFolder = objFS.GetFolder(strUNCPath)
    Set objRE = new RegExp
    objRE.Pattern = strMarkerFile
    objRE.IgnoreCase = True

    For Each objFile In objFolder.Files
        If objRE.Test(objFile.Name) Then
            UpdateRunning = True
            Exit Function
        End If
    Next
    UpdateRunning = False
End Function
```

The *UpdateRunning* function checks to see if the marker file exists on the server. If it does, the updates must be running and the function returns the value of *True*.

5. Enter the following text to create the *UpdateComplete* function:

```
' ***** UPDATE COMPLETE? *****
' Checks for completed marker file. If it exists, the update has already run.
Function UpdateComplete
    Set objFS = CreateObject("Scripting.FileSystemObject")
    Set objFolder = objFS.GetFolder(strUNCPath)
    Set objRE = new RegExp
    objRE.Pattern = strMarkerCompleteFile
    objRE.IgnoreCase = True

    For Each objFile In objFolder.Files
        If objRE.Test(objFile.Name) Then
            UpdateComplete = True
            Exit Function
        End If
    Next
    UpdateComplete = False
End Function
```

The *UpdateComplete* function checks to see if a marker file has been created which signifies the completion of the update. If this file exists, the function returns a value of `True`.

6. Enter the following text to create the *UserPatchPrompt* function:

```
' ***** USER PATCH PROMPT *****
' Prompt the user whether they would like to run the updates at this time.
Function UserPatchPrompt
    intAnswer=Msgbox("Anupdatehasbeendetected.Wouldyouliketoruntheupdatenow?"&vbLF&_
        "The update process will require several reboots!", vbYesNo, "Update Detected")
    If intAnswer = vbYes Then
        UserPatchPrompt = True
        InsertMarker
    Else
        UserPatchPrompt = False
    End If
End Function
```

The *UserPatchPrompt* function prompts the user with a *Yes/No* dialog. If the user selects *Yes*, the patch runs and the function returns a value of `True`. If the user selects *No*, the function return a value of `False` and the patch will not run.

7. Enter the following text to create the *RunPatch* routine:

```
' ***** RUN PATCH *****  
' The code to run the patches would occur here.  
Sub RunPatch  
    ' Enter code to execute the patch(es)  
    MsgBox "Patch has been applied"  
    InsertCompleteMarker  
End Sub
```

The *RunPatch* routine is used to run the patch. Any code to start a patch can be placed into this routine. After the patch is run, a message is sent to the user indicating the patch has been completed. Another routine, called *InsertCompleteMarker* is run to create a marker file to indicate the patch has been run.

8. Enter the following text to create the *Frozen* function:

```
' ***** DEEP FREEZE FROZEN? *****  
' Checks to see if Deep Freeze is Frozen and returns True or False.  
Function Frozen  
    Set objShell = CreateObject("Wscript.Shell")  
    intStatus = objShell.Run("DFC password /ISFROZEN", 1, True)  
    If intStatus = 0 Then 'DF is Thawed  
        Frozen = False  
    Else  
        If intStatus = 1 Then 'DF is Frozen  
            Frozen = True  
        Else  
            'A number of other reasons.  
        End If  
    End If  
End Function
```

The *Frozen* function checks to see if Deep Freeze is Frozen. If it is Frozen, the function returns a value of *True*. If Deep Freeze is Thawed, the function returns a value of *False*.

9. Enter the following text to create the *BootFrozen* routine:

```
' ***** BOOT FROZEN *****  
Sub BootFrozen  
    Set objShell = CreateObject("Wscript.Shell")  
    objShell.Run("DFC password /BOOTFROZEN")  
End Sub
```

The *BootFrozen* routine is used to put workstations into a Frozen State. The password in the DFC command line must be modified to match password created for the command line control.

10. Enter the following text to create the *BootThawed* routine:

```
' ***** BOOT THAWED *****  
Sub BootThawed  
    Set objShell = CreateObject("Wscript.Shell")  
    objShell.Run("DFC password /BOOTTHAWED")  
End Sub
```

The *BootThawed* routine is used to set workstations in a Thawed state. The password in the DFC command line must be modified to match the password created for the command line control.

11. Enter the following text to create the *InsertMarker* routine:

```
' ***** INSERT MARKER *****  
' Insert the marker file to indicate the patch is in progress.  
Sub InsertMarker  
    Set objFSO = CreateObject("Scripting.FileSystemObject")  
    Set objFile = objFSO.CreateTextFile(strUNCPath & strMARKERFILE)  
End Sub
```

The *InsertMarker* routine creates a marker file on the server to indicate the patch is currently being run. This marker file remains on the server until it is removed by the *DeleteMarker* routine.

12. Enter the following text to create the *RemoveMarker* routine:

```
' ***** REMOVE MARKER *****  
' Remove the marker file to indicate the patch is complete  
Sub RemoveMarker  
    Set objFSO = CreateObject("Scripting.FileSystemObject")  
    objFSO.DeleteFile(strUNCPath & strMarkerFile)  
End Sub
```

The *RemoveMarker* routine removes the marker file on the server to indicate the patch is no longer being run.

13. Enter the following text to create the *InsertMarkerComplete* routine:


```
' ***** INSERT UPDATE COMPLETE MARKER *****  
' This inserts an update completed file to prevent update looping  
Sub InsertCompleteMarker  
    Set objFSO = CreateObject("Scripting.FileSystemObject")  
    Set objFile = objFSO.CreateTextFile(strUNCPath & strMarkerCompleteFile)  
End Sub
```

The *InsertMarkerComplete* routine creates a file to indicate if the patch has been run on a machine. As long as this file exists on the server, the user is never prompted and the patch is never run.

14. Enter the following text to cleanup the script objects:

```
' ***** CLEANUP *****  
Set objNet = Nothing  
Set objFile = Nothing  
Set objRE = Nothing  
Set objFolder = Nothing  
Set objTS = Nothing  
Set objFS = Nothing  
Set objTextFile = Nothing  
Set objFSO = Nothing
```

This code cleans up all the objects that have been created throughout the script.

15. Save the file as *DF Update.vbs*. Make sure the file is saved as a *.vbs* and not a *.txt*. The icon should look like the following: 

The script is now ready to be implemented through a logon script in Group Policy.

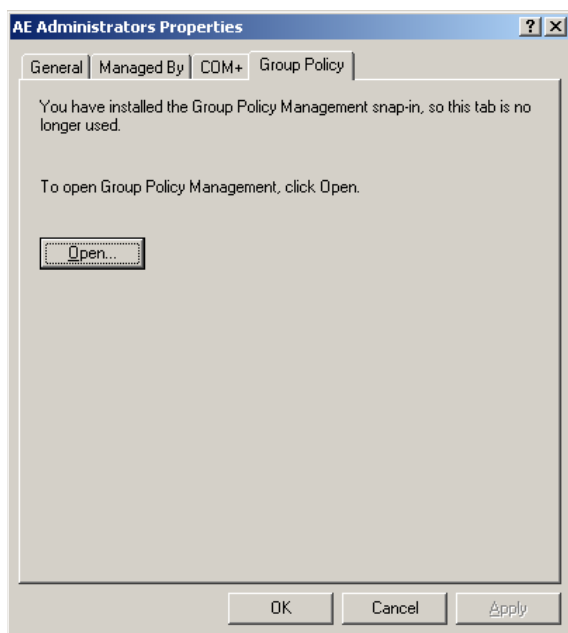
NOTE: The script does not contain any error handling in order to simplify it.

Creating the Group Policy

Before the policies are created, ensure the server has been updated to use the Group Policy Management Console. The following documentation assumes this patch has been downloaded and installed on the server. This utility can be found by searching Microsoft's Web site for *Group Policy Management Console*.

It is assumed there is an Organizational Unit (OU) for those users whom will be logging on to the network with a laptop machine requiring updates. Use the following steps to create the group policy:

1. Right-click on the desired User OU and select *Properties*. The properties dialog appears.
2. Select the *Group Policy* tab. If the Group Policy Management console is successfully installed, the following screen appears:



3. Click *Open*.
The *Group Policy Management* window opens, displaying all the OUs that have been created.
4. Right-click on the desired OU and select *Create and Link GPO Here*. The *New GPO* dialog appears.
5. Type *DfLogonPatchManagement* and click OK.
A GPO with the name of *DfLogonPatchManagement* appears under the desired OU.

Modifying the Group Policy

Now that the GPO has been created, it needs to be modified. In this case, the user Logon script is modified using the following steps:

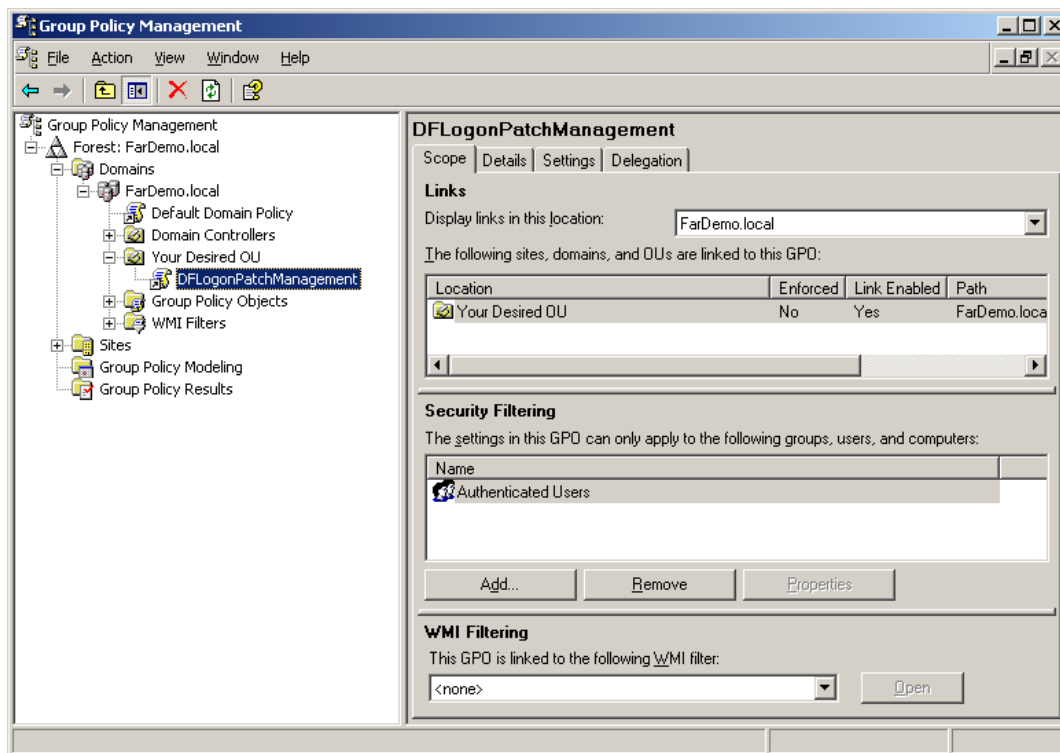
1. Right-click on *DfLogonPatchManagement* and select *Edit*. The *Group Policy Object Editor* opens.
2. Browse to the Logon/Logoff scripts for the user through *User Configuration>Windows Settings>Script (Logon/Logoff)*.
3. Double-click *Logon* to open the *Logon Properties* dialog.
4. Click *Show Files...* to open *Windows Explorer*. Place the script file created earlier in this folder.
5. Close *Windows Explorer*.
6. Click *Add* in the *Logon Properties* dialog. The *Open* dialog should appear and point to the folder where the script was just placed.
7. Select *DF Update.vbs* and click *OK*.
8. Click *OK* on the *Logon Properties* dialog to save the settings.

Enforcing the Group Policy

The logon script has been configured to execute when the user logs on. However, the GPO is not yet enforced. Enforcing a GPO indicates to the Active Directory server that it needs to run.

To enforce the newly created GPO, right-click on *DfLogonPatchManagement* and select *Enforced* to ensure the logon/logoff scripts are applied to the selected OU.

The policy now indicates that it is enforced. This can be verified by checking to see if the *Enforced* column in the *Group Policy Management* window displays a *Yes*.



Real Time Patch Maintenance

This method involves patching a machine in real time. This method is best used when the machines are not in use. Sometimes a patch needs to be manually applied to a group of machines. Scheduling the task may not be an option. This method involves disabling Deep Freeze locally at the client through the Enterprise Console or with the Command Line Control. The update can then be applied and Deep Freeze can be re-enabled.

Disabling Deep Freeze Locally

Use the following steps to put Deep Freeze into a Thawed state from the local machine:

1. To access the Deep Freeze control dialog, use one of the following methods to log on:
 - Press SHIFT and double-click the Deep Freeze icon in the System Tray
 - Use the keyboard shortcut CTRL+SHIFT+ALT+F6
2. The Deep Freeze password dialog appears. Enter your Deep Freeze password. This password would have been configured in the Configuration Administrator prior to creating the workstation installation file, or applied through a configuration update.
3. Under the *Boot Control* tab, select *Boot Thawed* and click OK. When the machine restarts, it is in a Thawed state. At this point any changes made to the machine are permanent.

Disabling Deep Freeze Through the Enterprise Console

Use the following steps to put a workstation into a Thawed state using the Deep Freeze Enterprise console:

1. Open the Deep Freeze Enterprise console. This program is created under the *Create Programs* tab in the Configuration Administrator.
2. Select the workstations that need to be put into a Thawed state.
3. Click *Thaw Workstation* in the toolbar. This puts the selected workstations into a Thawed state.

Disabling Deep Freeze Through the Command Line Control

The Deep Freeze Command Line Control can be used to disable Deep Freeze. This control can be used in scripts, batch files, and in conjunction with any 3rd party management utility capable of pushing scripts to systems. For more information about the different switches offered by the command line control, refer to the following document:

http://www.faronics.com/whitepapers/DF_RemoteAdministration.pdf

Configuring Software to Update in a Thawed Location

It is possible to update software that resides in a Thawed location. In these cases, the software would have to exist entirely on the Thawed partition. Remember the following rules when configuring software to run from a Thawed location:

1. If updates have to make changes to the registry, Deep Freeze needs to be in a Thawed state. The reason for this is that the registry is stored on the Frozen location.
2. Many programs store data to the user folders. The user folders can be mapped to a Thawed location. However, if the user folders are not being mapped, ensure that the updates are not making changes to settings stored there.

Appendix A - Deep Freeze and SUS/WSUS FAQ

Which Windows operating systems are able to use the Deep Freeze Run Windows Updates feature?

The following versions of Windows support the *Run Windows Updates* feature:

- Windows 2000 Service Pack 2 or 3 with SUS Client
- Windows 2000 Service Pack 4, which includes the SUS Client
- Windows XP with SUS Client
- Windows XP Service Pack 1 or 2, which includes the SUS Client
- Windows Vista

Does the Run Windows Updates feature require an administrator to be logged into the workstation?

The feature works while any type of user is logged in, or if the workstation isn't logged in at all. This feature uses the Windows Update service running under the local system account.

How can I be sure that the updates have been installed correctly?

Deep Freeze does not actually perform the updates or track which updates have been installed. Instead, the normal Microsoft method of installing updates is used. To check if the update took place, consult the workstation's update log at `C:\WINDOWS\Windows Update.log`.

What happens if an update is interrupted during download or installation because the Maintenance period ended or the workstation was restarted or powered off?

If an update is incomplete for any reason, the mechanism that Microsoft uses will correct and reinstall the update the next time the service is called.

Will the workstation restart during the update process if the update being installed requires it to do so?

Yes, the workstation will restart as many times as required, until the updates are completed.

What do I have to configure on each workstation to ensure that the updates are downloaded during the Maintenance period?

Deep Freeze automatically coordinates the update. Deep Freeze does not actually perform the update but calls the Microsoft update service during the Maintenance period. The Microsoft update service then performs the update either via the Internet or a designated SUS/WSUS server.

Can the IP address of the SUS/WSUS server be updated with a configuration update?

Yes, all of the Maintenance options can be changed with a configuration update.

I want to use the Run Windows Updates feature, but the Automatic Updates tab of my System Properties Control Panel does not allow me to enable Windows updates. How can I change this setting?

Deep Freeze disables the Automatic Updates settings in the System Properties Control Panel so they won't interfere with the normal operation of Deep Freeze. No settings need to be configured if you have selected *Control Windows Updates* in the Deep Freeze configuration that is installed. Deep Freeze automatically makes the call to update the software independent of the settings in this panel.

Why can't I select Run Windows Updates and Run Batch File on the same day? I am unable to select both options on the Maintenance tab of the Configuration Administrator.

Deep Freeze does not allow *Run Windows Updates* and *Run Batch File* to both be selected for the same day of the week. This is done to eliminate the possibility of a conflict occurring between the two options.

Appendix B - Deep Freeze Update Script

The entire script explained in the Logon Patch Maintenance section has been included here. This makes it easy to see the entire script rather than one segment at a time. The script can be downloaded from the following address: http://www.faronics.com/exe/DFEnt_ADUpdateScript.zip

```
' *****
' ***          DF SIMPLE UPDATE SCRIPT SAMPLE          ***
' ***
' *** Author:    Faronics Corporation          ***
' *** Date:      12/29/2005                    ***
' ***
' *** Associated Files:                        ***
' *** <ComputerName>.mar - Used to indicate patch is running ***
' *** COMPLETE-<ComputerName>.fin - Indicates patch complete ***
' *** DFC.exe - Deep Freeze Command Line Control          ***
' *****

' NOTES:
' The following script will turn off Deep Freeze, run updates and turn on Deep Freeze.

' ***** GLOBAL ASSEMBLIES *****
Set objNet = CreateObject("WScript.NetWork")

' ***** GLOBAL VARIABLES *****
' Modify the UNC path to match that of your server environment.
strUNCPath = "\\FarDemo.local\NETLOGON\"
strMarkerFile = objNet.ComputerName & ".mar"
strMarkerCompleteFile = "COMPLETED-" & objNet.ComputerName & ".fin"

' ***** MAIN *****
' Calls all of the other routines...
If UpdateRunning = True Then
    RunPatch
    RemoveMarker
    BootFrozen
Else
    If UpdateComplete = False Then
        If UserPatchPrompt = True Then
            InsertMarker
            If Frozen = True Then
                BootThawed
            Else
                RunPatch
                RemoveMarker
                BootFrozen
            End If
        End If
    End If
End If
```

```

        End If
    Else
        ' Exit Script
    End If
Else
    ' Exit Script
End If
End If

' ***** UPDATE RUNNING? *****
' Check for marker file. If exists, the update is running. Return True.
Function UpdateRunning
    Set objFS = CreateObject("Scripting.FileSystemObject")
    Set objFolder = objFS.GetFolder(strUNCPath)
    Set objRE = new RegExp
    objRE.Pattern = strMarkerFile
    objRE.IgnoreCase = True

    For Each objFile In objFolder.Files
        If objRE.Test(objFile.Name) Then
            UpdateRunning = True
            Exit Function
        End If
    Next
    UpdateRunning = False
End Function

' ***** UPDATE COMPLETE? *****
' Checks for completed marker file. If it exists, the update has already run.
Function UpdateComplete
    Set objFS = CreateObject("Scripting.FileSystemObject")
    Set objFolder = objFS.GetFolder(strUNCPath)
    Set objRE = new RegExp
    objRE.Pattern = strMarkerCompleteFile
    objRE.IgnoreCase = True

    For Each objFile In objFolder.Files
        If objRE.Test(objFile.Name) Then
            UpdateComplete = True
            Exit Function
        End If
    Next
    UpdateComplete = False
End Function

' ***** USER PATCH PROMPT *****
' Prompt the user whether they would like to run the updates at this time.
Function UserPatchPrompt
    intAnswer=Msgbox("Anupdatehasbeendetected.Wouldyouliketoruntheupdatenow?"&vbLF&_
        "The update process will require several reboots!", vbYesNo, "Update Detected")
    If intAnswer = vbYes Then

```

```

        UserPatchPrompt = True
        InsertMarker
    Else
        UserPatchPrompt = False
    End If
End Function

' ***** RUN PATCH *****
' The code to run the patches would occur here.
Sub RunPatch
    ' Enter code to execute the patch(es)
    ' The next two lines would run a program by the name of update.exe
    ' Set objShell = CreateObject("Wscript.Shell")
    ' objShell.Run("update.exe")
    MsgBox "Patch has been applied"
    InsertCompleteMarker
End Sub

' ***** DEEP FREEZE FROZEN? *****
' Checks to see if Deep Freeze is Frozen and returns True or False.
Function Frozen
    Set objShell = CreateObject("Wscript.Shell")
    intStatus = objShell.Run("DFC password /ISFROZEN", 1, True)
    If intStatus = 0 Then 'DF is Thawed
        Frozen = False
    Else
        If intStatus = 1 Then 'DF is Frozen
            Frozen = True
        Else
            'A number of other reasons.
        End If
    End If
End Function

' ***** BOOT FROZEN *****
Sub BootFrozen
    Set objShell = CreateObject("Wscript.Shell")
    objShell.Run("DFC password /BOOTFROZEN")
End Sub

' ***** BOOT THAWED *****
Sub BootThawed
    Set objShell = CreateObject("Wscript.Shell")
    objShell.Run("DFC password /BOOTTHAWED")
End Sub

' ***** INSERT MARKER *****
' Insert the marker file to indicate the patch is in progress.
Sub InsertMarker

```

```
        Set objFSO = CreateObject("Scripting.FileSystemObject")
        Set objFile = objFSO.CreateTextFile(strUNCPath & strMARKERFILE)
End Sub

' ***** REMOVE MARKER *****
' Remove the marker file to indicate the patch is complete
Sub RemoveMarker
    Set objFSO = CreateObject("Scripting.FileSystemObject")
    objFSO.DeleteFile(strUNCPath & strMarkerFile)
End Sub

' ***** INSERT UPDATE COMPLETE MARKER *****
' This inserts an update completed file to prevent update looping
Sub InsertCompleteMarker
    Set objFSO = CreateObject("Scripting.FileSystemObject")
    Set objFile = objFSO.CreateTextFile(strUNCPath & strMarkerCompleteFile)
End Sub

' ***** CLEANUP *****
Set objNet = Nothing
Set objFile = Nothing
Set objRE = Nothing
Set objFolder = Nothing
Set objTS = Nothing
Set objFS = Nothing
Set objTextFile = Nothing
Set objFSO = Nothing
```

Appendix C - Common Update Scenarios

The following section presents some update scenarios and possible solutions to these scenarios.

Scenario 1 - Updating Clients in a Dynamic Update Environment

REQUIREMENT

In this environment, the policy in the organization is to update the machines as soon as possible with the latest critical updates and antivirus definitions. Using management software, the updates are delivered to the client machines as soon as the updates are available. Because the clients have Deep Freeze installed, these updates are removed with a restart. How can patches be deployed in this type of environment?

SOLUTION

In this scenario, a scheduled maintenance period can be used to permanently update the client machines with the newest updates. This period could occur once a day or once a week. Any changes made during this maintenance period are permanent. Any new updates made to the machine while it is Frozen are only present until the machine is restarted. This has the same exact effect of a machine that is not Frozen with all the benefits of a Frozen machine. For more information, refer to the section entitled [Scheduled Patch Maintenance](#).

Scenario 2 - Updating in a 24-Hour Lab Environment

REQUIREMENT

Some cases exist where computers are in use for 24 hours. In these environments, it can be rather difficult to take machines offline to apply changes. Most patches do not require a restart. In order to disable Deep Freeze, a restart is required. How can patches be deployed in this type of environment.

SOLUTION

In these types of environments, the machines should be kept in a consistent state. Deep Freeze ensures these machines are reliable any time the machines are in use. The solution for patching is based on a rotation. As one system is rotated in, another is taken offline to be updated. The other solution would be to have the whole lab taken offline to perform a real-time update. For more information, refer to the section entitled [Real-Time Patch Maintenance](#).

Scenario 3 - Updating in a Mobile Environment

REQUIREMENT

In this environment, all users work with portable laptops. These laptops are almost never connected to the network. When these machines do connect to the network, updates are run. It is not possible to configure a maintenance period. What methods are available here?

SOLUTION

This environment will most likely use a logon script when the machine is attached to the network. This script initializes the update procedure. Deep Freeze can easily be disabled at this time using the command line control. This control can be called within the script. For more information, refer to the section entitled [Logon Patch Maintenance](#).

