



WHITEPAPER

# DEEP FREEZE ENTERPRISE PATCH MANAGEMENT



# Content Page

|  |    |
|--|----|
| Introduction .....   | 03 |
| Scheduled Patch Maintenance .....  | 03 |
| Scheduling Windows Updates .....   | 04 |
| Scheduling Windows Updates using a Windows Update Workstation Task ..... | 04 |
| Scheduling Windows Updates through Group Policy .....                    | 08 |
| Scheduling Antivirus Updates .....                                       | 10 |
| Scheduling Additional Program Updates .....                              | 10 |
| Logon Patch Maintenance .....  | 10 |
| Logon Patch Maintenance Example .....                                    | 13 |
| Creating the Group Policy .....  | 24 |
| Modifying the Group Policy .....   | 25 |
| Enforcing the Group Policy .....   | 26 |
| Real-time Patch Maintenance .....  | 27 |
| Disabling Deep Freeze Locally .....                                      | 27 |
| Disabling Deep Freeze Through the Enterprise Console .....               | 28 |
| Disabling Deep Freeze Through the Command Line Control .....             | 28 |
| Configuring Software to Update in a Thawed Location .....                | 29 |
| Appendix A: Deep Freeze and SUS/WSUS FAQ .....                           | 29 |
| Appendix B: Deep Freeze Update Script .....                              | 31 |
| Appendix C: Common Update Scenarios .....                                | 37 |

## 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 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. 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

Deep Freeze has two states: Frozen and Thawed. In a Thawed state, any changes made to a workstation are retained after a reboot. In a Frozen state, any changes can be made during a user session, but after a reboot, these changes will disappear. Scheduled patch maintenance allows the administrator to specify a period of time when the client workstations restart with Deep Freeze in a Thawed state. During this Maintenance Period, software updates, Windows updates, and antivirus definition updates can be applied. Scripts can be run and batch files can be executed.

Scheduled patch maintenance is an appropriate strategy for workstation labs. 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.

Workstation Tasks are configured using the Deep Freeze Configuration Administrator. The Configuration Administrator is used to configure workstation installation files as well as configuration files. Configuration files can be used to apply the changes to deployed workstations through the Deep Freeze Enterprise Console.

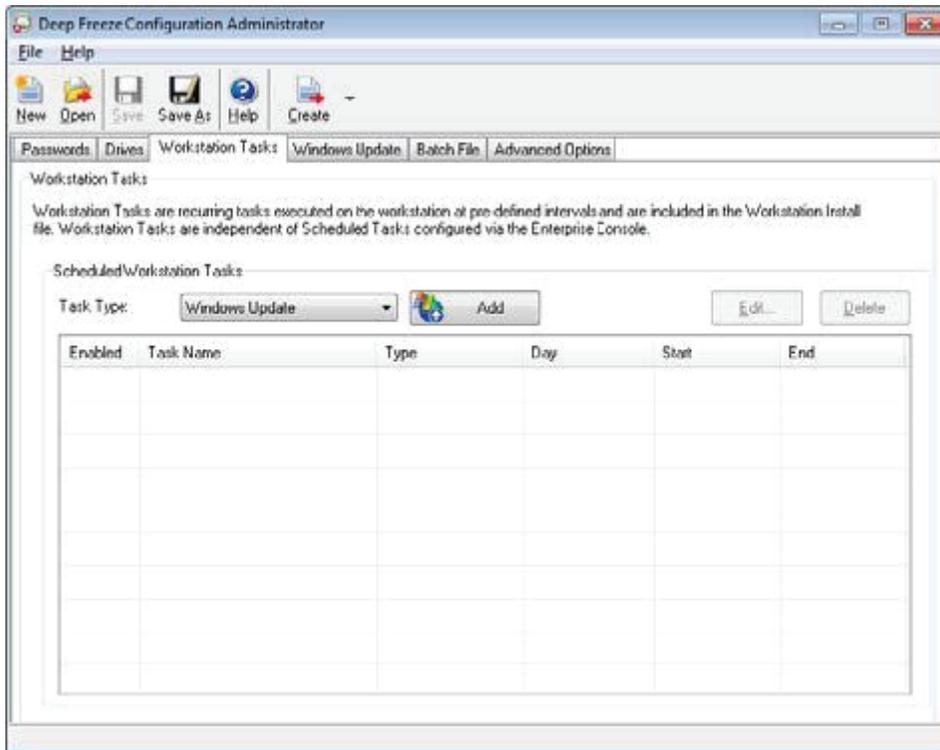
Depending on the policies in place, certain updates may need to be run. Windows and antivirus updates tend to be the most frequent. The following information explains some of the update scenarios encountered and the different methods available to handle these updates.

## Scheduling Windows Updates

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 during a Workstation Task. Deep Freeze can also be set up to execute a batch file during the Maintenance Period. A batch file could be used to start the Windows Update process.

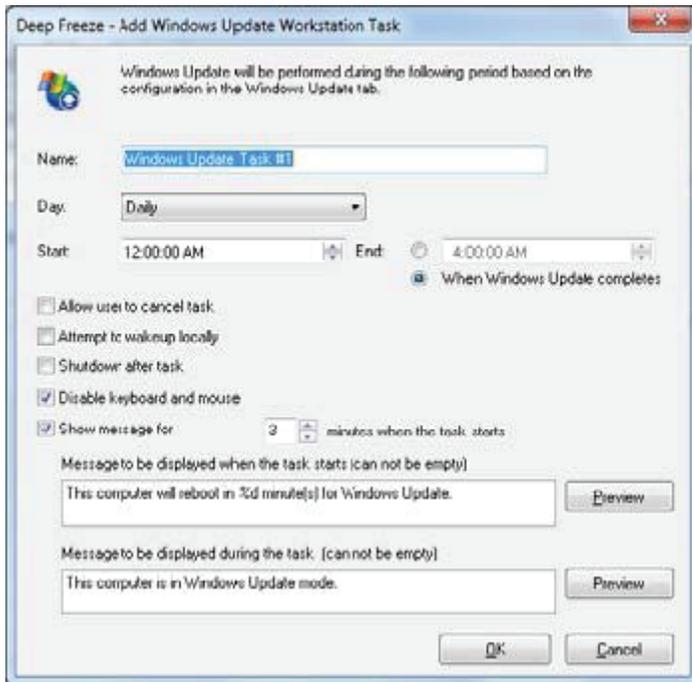
## Scheduling Windows Updates using a Windows Update Workstation Task

Windows Update tasks are scheduled for downloading and installing Windows updates on the workstation. Windows updates can be downloaded even when the workstation is in a Frozen state. A Windows Update task has a Start Time and an optional End Time. After downloading Windows updates, the workstation reboots in a Thawed state to apply.



The Windows Update task can be scheduled by completing the following steps:

1. Select Windows Update from the Task Type drop-down and click Add.
2. The following options are displayed:

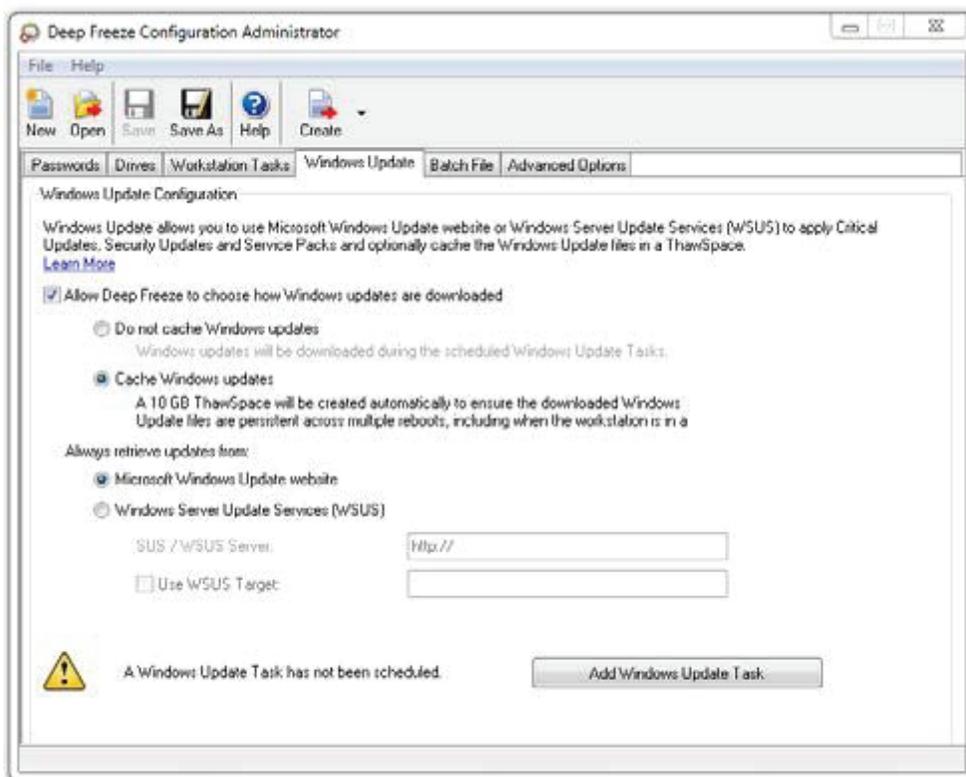


- **Name** - Specify a name for the task.
- **Day** - Select the day, or specify if the task will occur on Weekdays or Weekends.
- **Start** - Select the Start Time.
- **End** - Select the End Time. The minimum interval is 15 minutes. Alternatively, you can select "When Windows Update completes". If the Windows Update Task is not completed in 6 hours, Deep Freeze will end the task gracefully.
- **Allow user to cancel task**- Select the check box if the user is allowed to cancel the task before it starts.
- **Attempt to wake up locally** - Select this check box to wake up the workstation locally without requiring any communication from the Enterprise Console.
- **Shutdown after task** - Select the check box to shutdown the workstation after the task.
- **Disable Keyboard and Mouse** - Select the check box to disable keyboard and mouse during the task.
- **Show message** - Select the check box to display a message on the workstation Before and During the task. Specify the time interval in minutes and enter a brief message to be displayed before the task starts.

3. Click OK. You will be taken to the Windows Update Tab to configure additional settings if it has not been configured earlier.

In the above screen, a Windows Update task has been created to perform Windows updates at the workstation daily at 12:00 AM and end when Windows Update completes. The task is configured to display a message to the user before Windows Update. The keyboard and mouse are disabled.

The Windows Update tab allows you to customize settings for Windows Update. When you first create a Windows Update Task, you will be given an option to modify the default settings in the Windows Update tab. The settings in the Windows Update tab will apply to all Windows Update tasks.



Configure the settings in the Windows Update tab as follows:

Allow Deep Freeze to choose how Windows updates are downloaded: —select this checkbox to allow Deep Freeze to choose how Windows updates are downloaded. The following options are available:

#### Select the Windows updates download options:

Do not cache Windows updates — select this option to download Windows updates only during the Windows Update task.

#### Cache Windows updates

Select this option to download when the workstation is in a Frozen or Thawed state and install during the Windows Update Task. This option creates a 10 GB ThawSpace (for version 8.35 or higher) and the Windows updates are stored in the ThawSpace to ensure that Windows Update files are persistent across multiple reboots.

## 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.

It is recommended that the following configuration be used to configure the Automatic Update client on a workstation running Deep Freeze:



|   |                |
|---|----------------|
| Configure Automatic Updates   | Enabled        |
| Scheduled install day <i>If Maintenance will happen every day</i>   | 0              |
| <i>If Maintenance will only occur once per week</i>   | (Specific Day) |
| Scheduled install time <i>Set to 30 minutes after the start of the Deep Freeze Maintenance Period. For example, if the Deep Freeze maintenance period starts at 10 AM, set the scheduled install time to 1030 AM.</i> | 30 minutes     |
| 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 Windows updates are installed during the Thawed Period Workstation Task and that any updates that are downloaded (but not installed) will not attempt to reinstall on the client workstation while Frozen.

Administrators need to ensure that the Thawed Period Workstation Task configured in Deep Freeze is long enough to complete the download and installation of the updates from whatever source is configured and that workstations are either left onto enter the Maintenance Period or are woken up prior to the start of the Maintenance Period.

## 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.

For additional white papers describing antivirus products that may have been added to the Faronics Content Library after publication of this white paper, refer to:

<http://www.faronics.com/library>

## 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](http://www.faronics.com/whitepapers/DF_RetainUserData.pdf)

## Logon Patch Maintenance

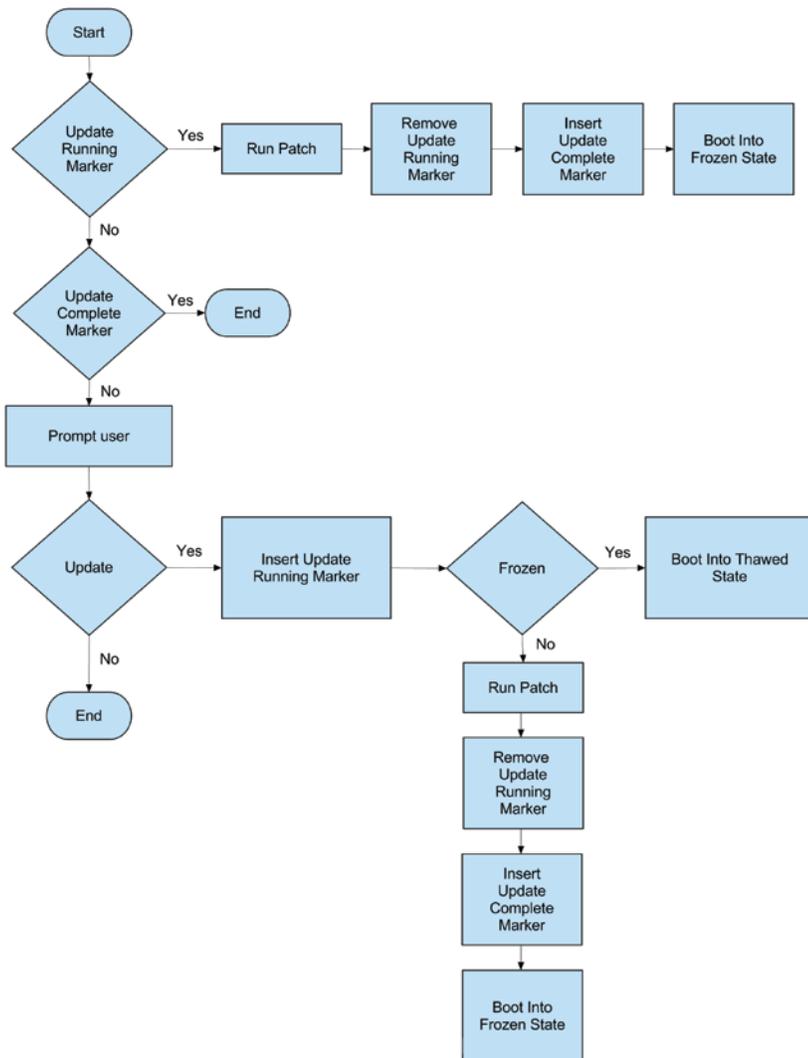
This option allows the administrator to install updates to the client workstation when a certain user logs on. In an Active Directory environment, a logon script can be executed to update the client workstation. Using Deep Freeze command line control (DFC), Deep Freeze can be disabled before the updates are run and re-enabled afterwards.

Logon patch maintenance is appropriate in mobile environments where users are working with laptops and are often off the local network. When they get back to the office or school 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 workstation 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 workstation boots several times, the script needs to check a value to see what phase of the script is currently running. Because the workstation 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 workstation.

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](http://www.faronics.com/exe/DFEnt_ADUpdateScript.zip)

### Creating the Update Script

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

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

NOTE: The script does not contain any error handling.

## 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. The 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 who will be logging on to the network with a laptop requiring updates. Use the following steps to create the Policy:

**01. Right-click on the desired User OU and select Properties.**

The properties dialog appears.

**02. Select the Group Policy tab.**

If the Group Policy Management console is successfully installed, the following screen appears:



**03. Click Open.**

The Group Policy Management window opens, displaying all the OUs that have been created.

**04. Right-click on the desired OU and select Create and Link GPO Here.**

The New GPO dialog appears.

**05. 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:

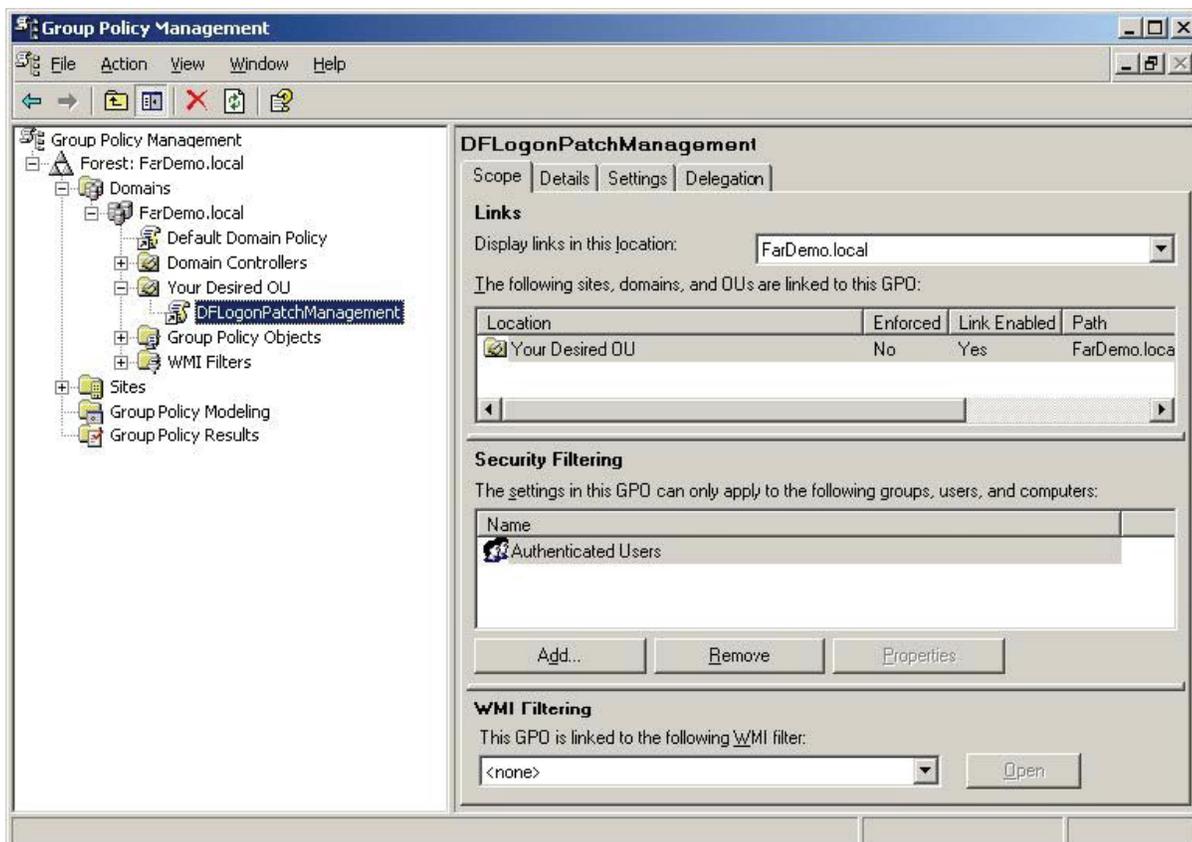
- 01.** Right-click on DfLogonPatchManagement and select Edit. The Group Policy Object Editor opens.
- 02.** Browse to the Logon/Logoff scripts for the user through User Configuration>Windows Settings>Script (Logon/Logoff).
- 03.** Double-click Logon to open the Logon Properties dialog.
- 04.** Click Show Files... to open Windows Explorer. Place the script file created earlier in this folder.
- 05.** Close Windows Explorer.
- 06.** Click Add in the Logon Properties dialog. The Open dialog should appear and point to the folder where the script was just placed.
- 07.** Select DF Update.vbs and click OK.
- 08.** 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 workstation in real-time. It is best used when the workstations are not in use. Sometimes a patch needs to be manually applied to a group of workstations and 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 (DFC). 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 workstation:

- 01.** To access the Deep Freeze login 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
- 02.** The Deep Freeze login 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.
- 03.** Under the Boot Control tab, select Boot Thawed and click OK. When the workstation restarts, it is in a Thawed state. At this point any changes made to the workstation 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:

- 01.** Launch the Deep Freeze Enterprise Console.
- 02.** Select the workstations that need to be put into a Thawed state.
- 03.** Click the Reboot Thawed icon in the toolbar or right-click and select Reboot Thawed from the menu. The selected workstations will reboot in the Thawed state.

## Disabling Deep Freeze Through the Command Line Control

The Deep Freeze command line control (DFC) can be used to disable Deep Freeze. It can be used in scripts, batch files, and in conjunction with any third 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](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:

- 01.** 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.
- 02.** Many programs store data to the user folders. The user folders can be redirected to a Thawed location. However, if the user folders are not redirected, ensure that the updates are not making changes to settings stored there.

## Appendix A: Deep Freeze and SUS/WSUS FAQ

### **01. 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. It uses the Windows Update service running under the local system account.

### **02. How can I be sure that the updates have been installed correctly?**

If you are using Deep Freeze to install the updates, you can view the status of the updates in the DFWuLogfile.log in the Faronics folder. You cannot change the name or location of the log file.

**03. What happens if an update is interrupted during download or installation because the Workstation Task 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.

**04. 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.

**05. What do I have to configure on each workstation to ensure that the updates are downloaded during the Windows Update Workstation Task?**

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.

**06. 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.

## Appendix B: Deep Freeze Update Script

The entire script explained in the Logon Patch Maintenance section has been included here. It can be downloaded from the following address:

[http://www.faronics.com/exe/DFEnt\\_ADUpdateScript.zip](http://www.faronics.com/exe/DFEnt_ADUpdateScript.zip)

```
' *****
' *** DF SIMPLE UPDATE SCRIPT SAMPLE ***
' *** ***
' *** Author: Faronics Corporation ***
' *** Date: 12/29/2005 ***
' *** ***
' *** Associated Files: ***
' *** <workstationName>.mar - Used to indicate patch is running ***
' *** COMPLETE-<workstationName>.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  
        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:**

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

**Solution:**

A scheduled Maintenance Period can be used to permanently update the client workstations with the newest updates. The Maintenance Period could be scheduled to occur as often as desired. Any changes made during the Maintenance Period are permanent. Any new updates made to the workstation while it is Frozen are only present until the workstation is restarted. This has the same exact effect of a workstation that is not Frozen with all the benefits of a Frozen workstation. 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 workstations are in use for 24 hours. In these environments, it can be difficult to take workstations 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 workstations should be kept in a consistent state. Deep Freeze ensures these workstations are reliable and available at all times. The solution for patching is based on a rotation. As one workstation is rotated in, another is taken offline to be updated.

An alternate solution is 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:

All users work with portable laptops. These laptops are almost never connected to the network. When they 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 laptop is attached to the network. This script initializes the update procedure. Deep Freeze can easily be disabled at this time using its command line control (DFC). This control can be called within the script. For more information, refer to the section entitled Logon Patch Maintenance.



[www.faronics.com](http://www.faronics.com)

Faronics' solutions help organizations increase the productivity of existing IT investments and lower IT operating costs. Incorporated in 1996, Faronics has offices in the USA, Canada, and the UK, as well as a global network of channel partners. Our solutions are deployed in over 150 countries worldwide, and are helping more than 30,000 customers.

**CANADA & INTERNATIONAL**  
1400 - 609 Granville Street  
P.O. Box 10362, Pacific Centre  
Vancouver, BC, V7Y 1G5  
Phone: +1-604-637-3333  
Fax: +1-604-637-8188  
Email: [sales@faronics.com](mailto:sales@faronics.com)

**UNITED STATES**  
5506 Sunol Blvd, Suite 202  
Pleasanton, CA, 94566 USA  
Call Toll Free: 1-800-943-6422  
Fax Toll Free: 1-800-943-6488  
Email: [sales@faronics.com](mailto:sales@faronics.com)

**EUROPE**  
8 The Courtyard, Eastern Road,  
Bracknell, Berkshire  
RG12 2XB, England  
Phone: +44 (0) 1344 206 414  
Email: [eurosales@faronics.com](mailto:eurosales@faronics.com)

**SINGAPORE**  
20 Cecil Street, #104-01,  
Equity Way, Singapore,  
049705  
Phone: +65 6520 3619  
Fax: +65 6722 8634  
Email: [sales@faronics.com.sg](mailto:sales@faronics.com.sg)