iTunes for Windows - Deploying via GPO

This article is the outgrowth of some work I did for a customer in recent weeks. Big companies are buying iPhones by the truckload to use in their architecture. These same companies have enough employees to make centralized iPhone activation a little unrealistic. Throw in a heterogeneous OS client environment and, well, system administrators want to know i) how to deploy iTunes for Windows so their users can activate their iPhones themselves and ii) how to enforce pre-configured parental controls on iTunes for Windows.

I'm going to assume that if you're reading this article and hoping to do something with it that you've got the following - a Windows 2008(3) server with Active Directory in place, some Windows XP / Vista (7?) clients bound to this AD, some Group Policy in place to manage the Windows client machines, and a little experience deploying software via Group Policy.

An additional tool you'll need for this project is:

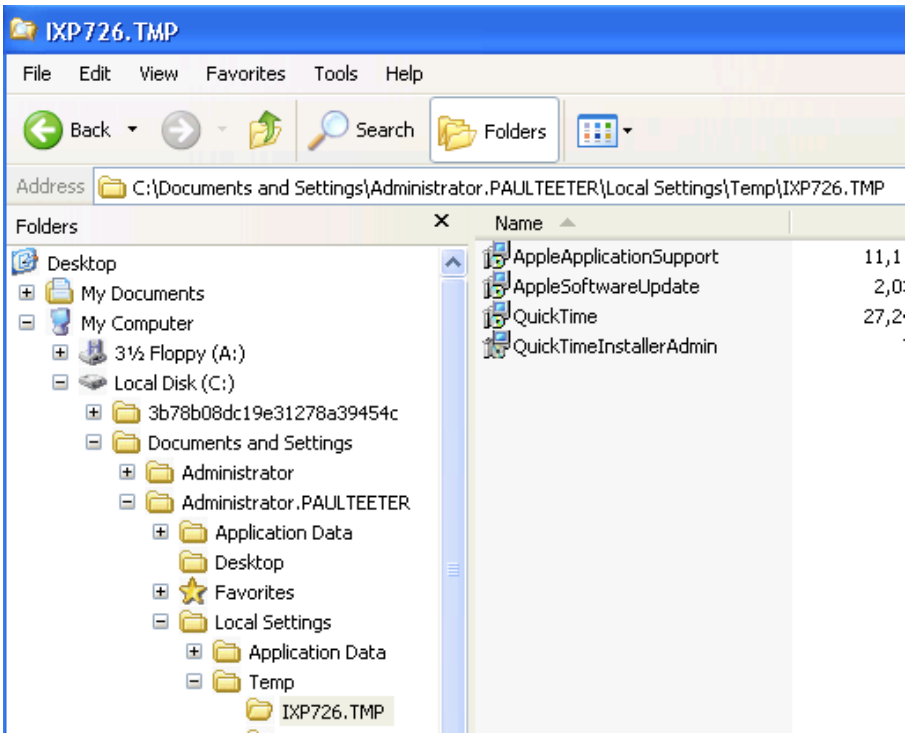
- Orca.exe included with Microsoft Windows SDK for Windows 7

See the link in the references section to download this tool. Also note that by default on a typical 64-bit Windows 2008 Server instance Orca.exe is installed in 'C:\Program Files (x86)\Orca'. Best to add that path to the PATH environment variable on whatever server/workstation you'll be using to accomplish this deployment.

Start by downloading the iTunes and QuickTime installers from Apple's website. Definitely download them separately. One of the required, supporting MSIs doesn't come with the iTunes installer and can only be acquired with the QuickTime installer.

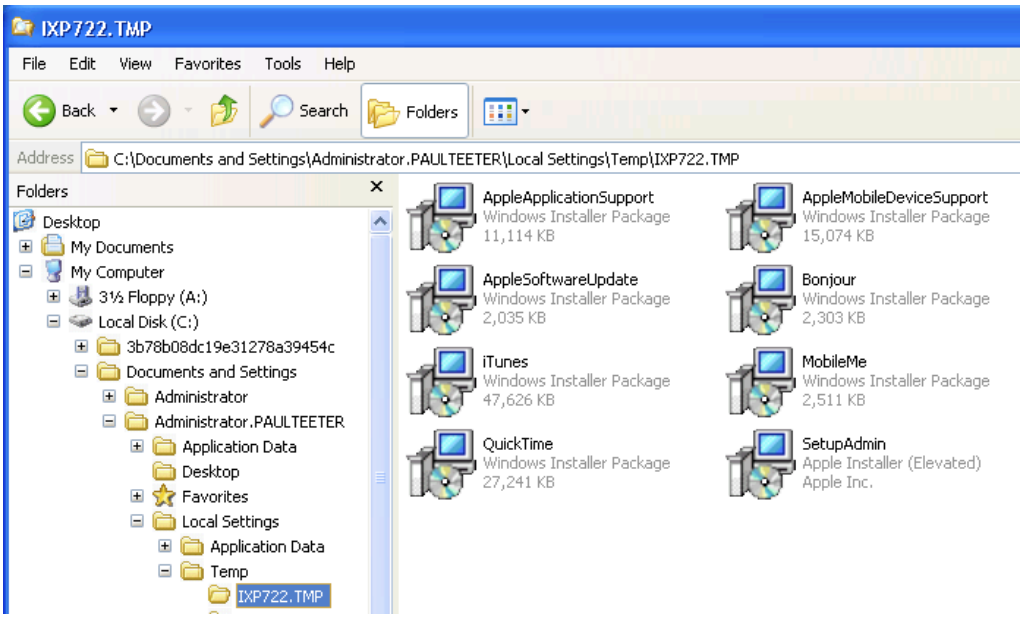
The files you download will be of the .exe variety. Don't panic, the MSIs get copied to a hidden location during the install process. To extract the necessary MSI files, enable 'show hidden files' in Windows Explorer, see the references section for a link to specific instructions.

Next launch the QuickTime installer and navigate to the 'home' folder of the current user. With hidden files viewable, navigate to Local Settings\temp (XP) or AppData\Local \Temp (Vista). You'll see a folder named #####.tmp, open it and see the enclosed .msi files.



Copy two MSIs to the Desktop - QuickTime.msi and AppleApplicationSupport.msi. Quit/ Cancel the QuickTime installer.

Now launch the iTunes installer. Navigate to the just about the same location within the current user's 'home' folder.



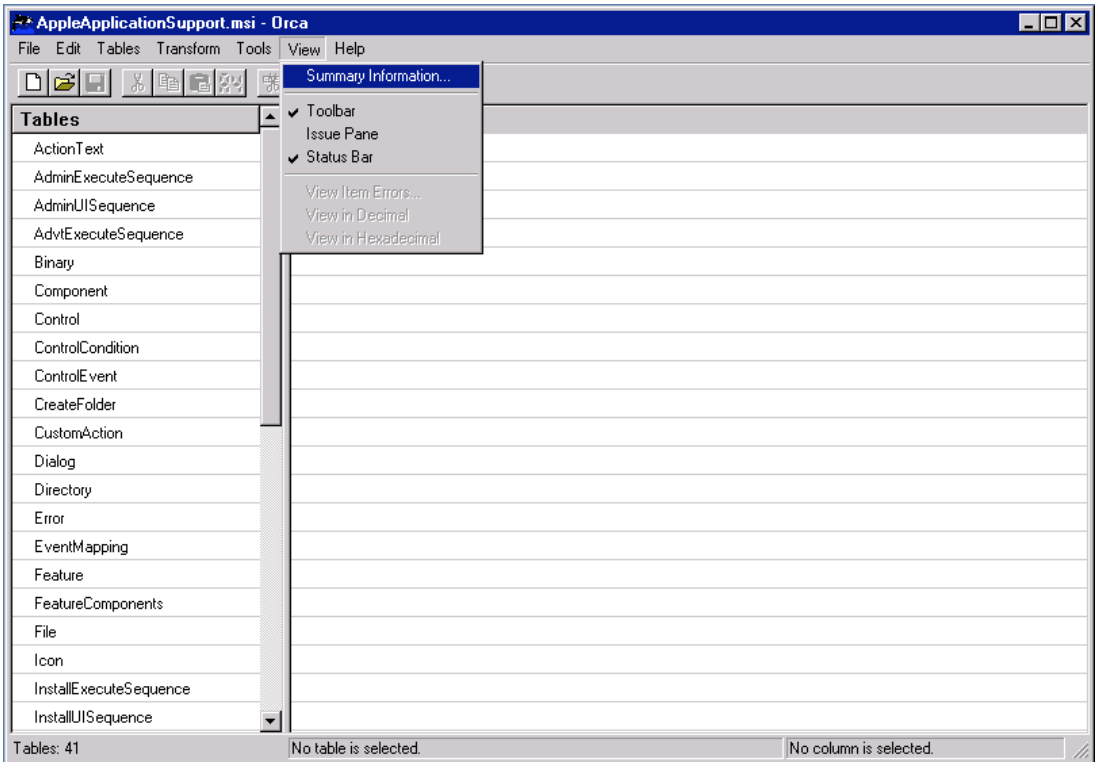
Copy two more MSIs to the Desktop - AppleMobileDeviceSupport.msi and iTunes.msi. Quit/Cancel the iTunes installer. Feel free to reverse the 'show hidden files' setting in Windows Explorer too.

That leaves us with 4 installer files. Two are for the major applications - iTunes & QuickTime. Two are supporting installers, required by the apps - AppleApplicationSupport and AppleMobileDeviceSupport. iTunes requires AppleMobileDeviceSupport to work with iPhones or iPod Touch's. The order in which these MSIs are installed is actually crucial and should follow this pattern:

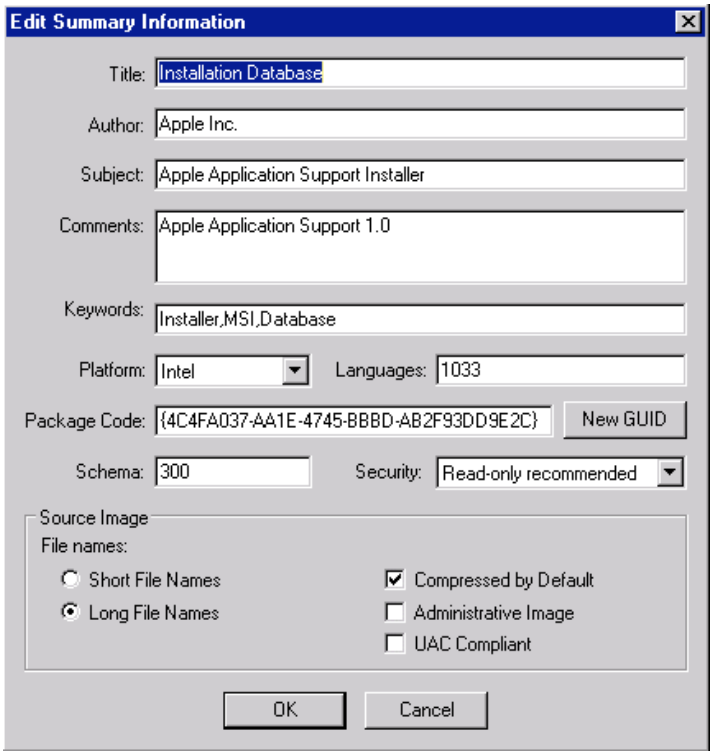
1. AppleApplicationSupport (AAS)
2. AppleMobileDeviceSupport (AMDS)
3. QuickTime
4. iTunes

We'll revisit this when we create and re-order the software install GPOs.

The 'supporting' MSIs will deploy without much modification. The only attribute that I'd choose to change is the language / locale. This could also be addressed when the software installation GPO is created, it's a personal preference. To modify the MSI, launch Orca and open the AAS or AMDS. From the View menu, select Summary.



In the ensuing popup windows, find the Languages text field. Delete all entries except the language/locale you need to support, e.g. 1033 for English - United States. See the references section for a link to a complete list of locale codes.



Click the OK button to finish with the popup window. From the File menu Save the modified MSI. Follow this process twice, once for AAS and then for AMDS.

The remaining two MSIs - iTunes & QuickTime - have a great wealth of discussion readily available online about modifications, transformations, etc. Go visit AppDeploy.com, see how many discussion threads exist about each. Links to the primary 'packaging' pages for each application at AppDeploy.com are provided in the references section at the end.

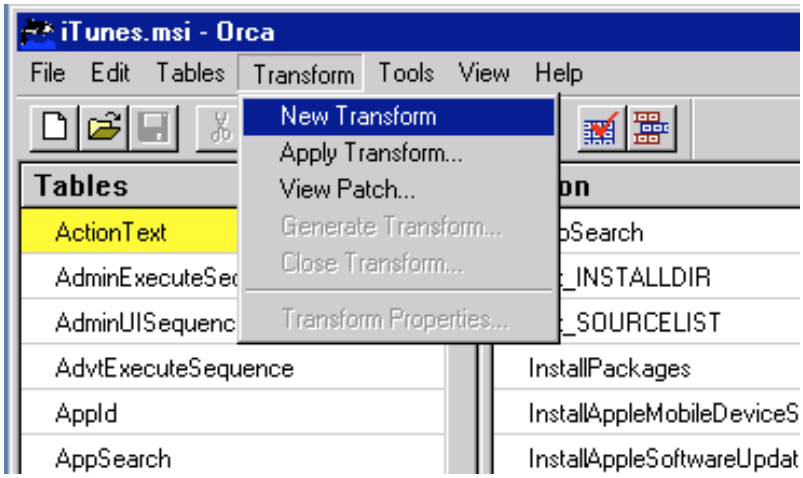
We'll keep it simple for this article. Perform the same language/locale modification to the QuickTime and iTunes MSIs as with the AAS and AMDS MSIs. It's especially crucial for the QuickTime MSI because QuickTime likes to default to Chinese as the installation language.

MSI transform files (.mst) can be used to change the specific way in which an MSI is installed on a client machine. Install options that can be changed with a transform in iTunes or QuickTime include:

- pre-acknowledge license agreement
- desktop shortcuts removal
- change the items added to the Start menu

- QuickTime ‘system tray’ icon removal
- disable automatic software updates

I’ll discuss how to create a transform with Orca, but leave the details of what you’d like your transform to do up to you. Launch Orca and open, for example, the iTunes MSI. From the Transform menu, choose ‘New Transform’.

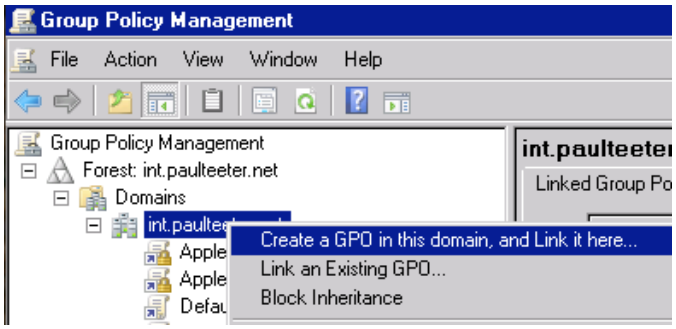


Select the table on which you’d like to execute a change. Make the change. Make whatever other value changes are necessary. From the Transform menu, select ‘Generate Transform’. Give the .mst file an appropriate name, save it to your desired location, and exit Orca.

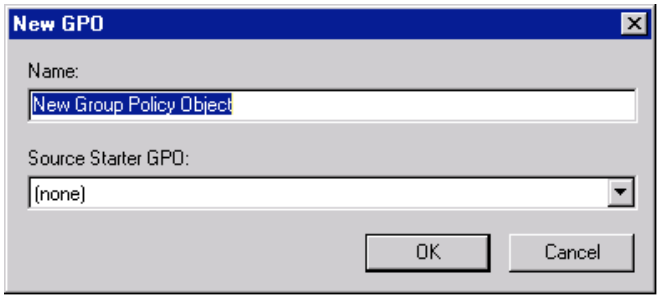
Now you have four MSI files and potentially two MST files. Collect them and place them in a shared folder on a Windows server in your environment. The shared folder and it’s contents should be accessible by bound client machines and authenticated domain users. My preference is to group the installers in unique subfolders within the network share - e.g. Network Share\iTunes, Network Share\QuickTimes, Network Share \AppleApplicationSupport, and Network Share\AppleMobileDeviceSupport.

Depending on your experience with creating Group Policy for an Active Directory domain, the next several paragraphs will be more or less important to you. I’m going into painful detail. Also, I know we in the Mac Enterprise community beat on Adobe mercilessly for making their software impossible to easily deploy on the Mac platform. As badly as their Mac installers suck, Adobes tools and documentation for deploying software on the Windows platform is quite helpful. Much of what I’ll detail here is a reinterpretation of an Adobe document listed among the references at the end.

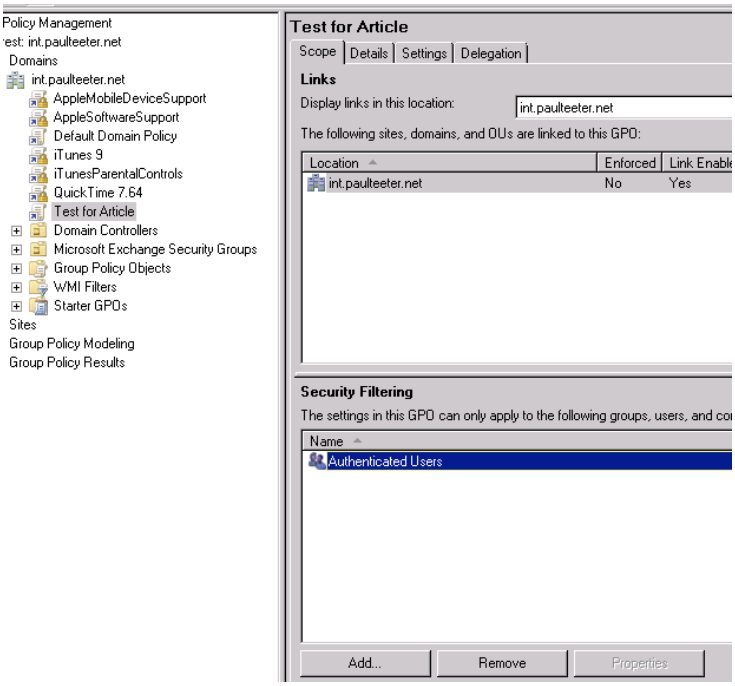
Find and launch the Group Policy Management tool on your Windows server, it’s buried among the Administrative Tools in the Start Menu. Drill down to the domain you plan to use for deploying your software and right click on it.



From the contextual menu that appears, choose ‘Create a GPO in this domain, and Link it here...’. Give the GPO an appropriate name like ‘AMDS MSI Deployment’ or ‘iTunes Deployment’, just some title that distinguishes it from other GPOs we’ll create.

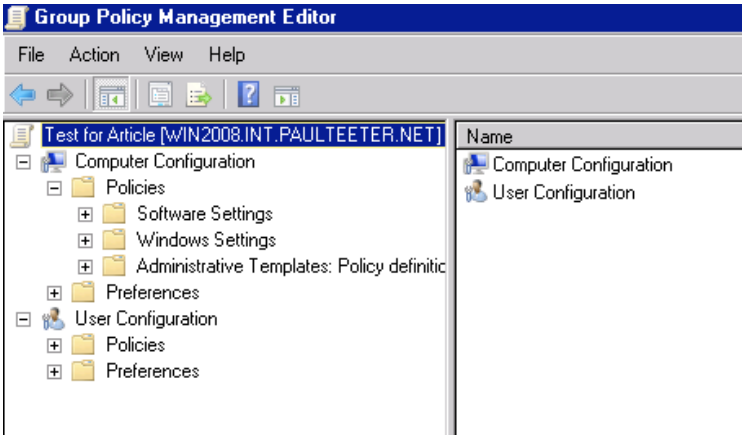


Leave the ‘Source Starter GPO’ field with the default value of ‘(none)’ and click the ‘OK’ button. This new GPO will appear in the list of GPO’s associated with the domain in the left-hand navigation pane. It’ll be muted in color because it’s not enforced yet.



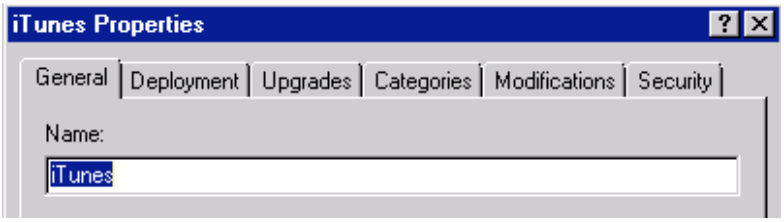
Click to select this new GPO. In the 'Security Filtering' pane, bottom right of the right window pane, select 'Authenticated Users' and click the 'Remove' button. Confirm it too when asked by clicking 'OK'. In the 'Links' pane, top right side, right click on your chosen domain. Select 'Enforce' and notice that the 'Enforced' status changes to 'Yes'.

Now we'll edit the GPO we created. In the left-hand navigation pane of the Group Policy Management window, select this new GPO. Right click on it and choose 'Edit'. The ensuing popup window opens the GPO in the Group Policy Management Editor.



Expand the 'Policies' folder that is subordinate to 'Computer Configuration'. Expand the 'Software Settings' folder and select the enclosed 'Software installation' item. Right click on 'Software installation' and choose 'New > Package'. Navigate to the network share that contains the MSI to deploy, i.e. \\SERVER\share\folder\iTunes.msi. Select the MSI and click the 'Open' button. Choose 'Advanced' for the deployment method and click the 'OK' button.

The Properties window for the MSI will open momentarily. You'll see 6 tabs in this

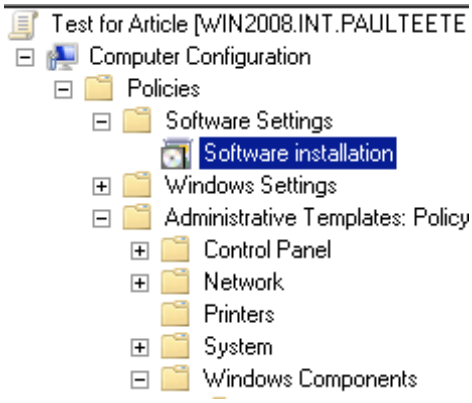


popup window. Follow these steps for each tab:

General - update the name if you prefer
Deployment - check the box for 'Uninstall this application when it falls out of the scope of management'
Upgrades - n/a
Categories - n/a

Modifications - Add the transform associated with the MSI if you created an MST for it (iTunes or QuickTime), reference the MST via \\SERVER\share\folder \transform.mst
Security - add or remove users, groups, or computer records as required in your environment (*we'll revisit this attribute when we choose the AD element on which the policy of enforced*)

Click the 'OK' button. Next expand the 'Administrative Templates' folder subordinate to 'Computer Configuration', then expand 'Windows Components'.



Scroll down and find the 'Windows Installer' folder. In the right hand window pane, we're going to change settings for i) Always install with elevated privileges and ii) Logging. Double click on 'Always install with elevated privileges'. In the ensuing popup window, check the radio button for 'Enabled' and click the 'OK' button. Double click on 'Logging'. In the ensuing popup window, check the 'Enabled' radio button and add 'rcv' to the Logging modes. The final mode value should be 'iweaprcv'. Click the 'OK' button.

That's it, your GPO is ready to be enforced and deployed. Repeat these steps for each of the MSIs - AAS, AMDS, QuickTimes, and iTunes.

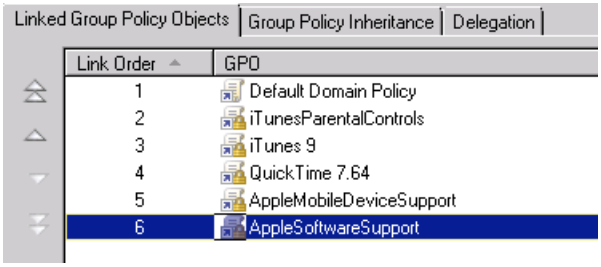
Since the order of installation for these Apple MSIs is crucial, let's order the GPOs. Again, the order of installation should be:

1. AppleApplicationSupport (AAS)
2. AppleMobileDeviceSupport (AMDS)
3. QuickTime
4. iTunes

In the Group Policy Management application window, select the domain on which you are applying these GPOs.

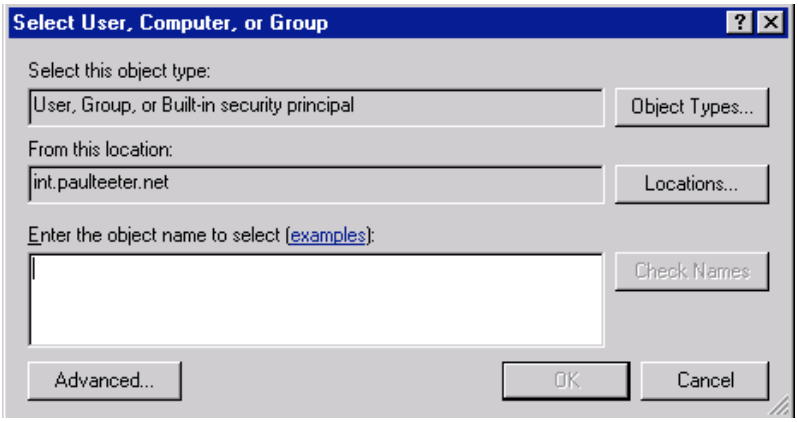


In the right hand window pane, select the 'Linked Group Policy Objects' tab. The GPO link order works like this - the higher the value, the earlier the object is applied in the GPO enforcement process.



In our context, AppleSoftwareSupport should have the highest link value to be installed first. And the (as yet undefined) iTunes Parental Controls GPO will have the lowest value to be installed last. Use the arrows at the very left of this window pane to push a policy object up or down in the list.

Last but not least, we need to add AD objects that will be governed by a created group policy object. The kind of AD object you choose to add depends on your company policy. Whatever AD object you choose, it must have at least READ access on the network share/MSI or MST in question. Refer to the 'Security' tab of the Installer package properties. If the computer, group, or user isn't listed in that tab in some form - make sure to add it. I'm going to work with computers but it's entirely acceptable to choose user groups or individual users. Choose one of the GPOs we just created in the left hand navigation window pane. In the lower right, focus on the 'Security Filtering' configuration area. Click the 'Add' button.



In the ensuing popup windows, click the 'Object Types' button. Check the box next to 'Computer' and click the 'OK' button. You may need to update the 'Locations' if you have multiple domains. Type a machine name in the 'Enter the object name to select' text area, click the 'Check Names' button when its available to verify the computer name. Then click the 'OK' button. Repeat this process for each of the four software installation GPOs created.

To trigger re-application of updated GPO, you can run 'gpupdate /force' on both the server and the chosen client machine. Execute it from the command prompt or the Start Menu's 'Run' item. The client machine will prompt for a reboot because of the policy update. Choose not to reboot initially. To verify that the client machine is poised to have this GPO applied on the next reboot, run 'gpresult /V' from the command prompt on the client.

```
Resultant Set Of Policies for Computer:
---------------------------------------

    Software Installations
    ----------------------
        GPO: AppleSoftwareSupport
            Name:              Apple Application Support
            Version:           1.0
            Deployment State:  Assigned
            Source:            \\WIN2008\gpoDeploy\AppleApplicationSupport\AppleApplicationSupport.msi
            AutoInstall:       True
            Origin:            Applied Application

        GPO: QuickTime 7.64
            Name:              QuickTime
            Version:           7.64
            Deployment State:  Assigned
            Source:            \\WIN2008\gpoDeploy\QuickTime\QuickTime.msi
            AutoInstall:       True
            Origin:            Applied Application

        GPO: iTunes 9
            Name:              iTunes
            Version:           9.0
            Deployment State:  Assigned
            Source:            \\WIN2008\gpoDeploy\iTunes9\iTunes.msi
            AutoInstall:       True
            Origin:            Applied Application

        GPO: AppleMobileDeviceSupport
            Name:              Apple Mobile Device Support
            Version:           2.6
            Deployment State:  Assigned
            Source:            \\WIN2008\gpoDeploy\AppleMobileDeviceSupport\AppleMobileDeviceSupport.msi
            AutoInstall:       True
            Origin:            Applied Application
```

Scroll up in the command prompt window to verify that the client thinks the GPO that has been applied is actually applied.

At this point, you could reboot your client machine to verify that the MSIs get installed properly. The reboot will take a bit longer, but you should see status messages that



indicates the MSI(s) are being installed.

The last concept in this article is that of pre-configured Parental Controls for iTunes. Apple's knowledge base has a detailed article about parental control settings, see the references section for a direct link. The crucial registry key to modify is **HKEY_LOCAL_MACHINE\Software\Apple Computer, Inc.\iTunes\Parental Controls**

\Default\AdminFlags. For the purposes of this article, we'll assume iTunes is being deployed on a new machine or a machine without pre-existing user accounts. (Applying iTunes Parental Controls registry settings to existing accounts is probably fodder for it's own article.) The possible values for AdminFlags are defined in hexadecimal (REG_DWORD in the Windows registry). To achieve a desired set of AdminFlags values, one must **add** the hexadecimal values assigned to each setting. What?! This table might help:

AdminFlags Flag Name	Hexadecimal Value	Decimal Value	*Base 16 Calculation*
kParentalFlags_Locked	0x00000001	1	*1 * 16^0*
kParentalFlags_DisablePodcasts	0x00000002	2	*2 * 16^0*
kParentalFlags_DisableMusicStore	0x00000004	4	*4 * 16^0*
kParentalFlags_DisableSharing	0x00000008	8	*8 * 16^0*
kParentalFlags_DisableExplicitContent	0x00000010	16	*1 * 16^1*
kParentalFlags_DisableRadio	0x00000020	32	*2 * 16^1*
kParentalFlags_RestrictMovieContent	0x00000040	64	*4 * 16^1*
kParentalFlags_RestrictTVShowContent	0x00000080	128	*8 * 16^1*
kParentalFlags_DisableCheckForUpdates	0x00000100	256	*1 * 16^2*
kParentalFlags_RestrictGames	0x00000200	512	*2 * 16^2*
kParentalFlags_DisableMiniStore	0x00000400	1024	*4 * 16^2*
kParentalFlags_DisableAutomaticDeviceSync	0x00000800	2048	*8 * 16^2*
kParentalFlags_DisableGetAlbumArtwork	0x00001000	4096	*1 * 16^3*
kParentalFlags_DisablePlugins	0x00002000	8192	*2 * 16^3*
kParentalFlags_DisableOpenStream	0x00004000	16384	*4 * 16^3*
kParentalFlags_DisableAppleTV	0x00008000	32768	*8 * 16^3*
kParentalFlags_DisableDeviceRegistration	0x00010000	65536	*1 * 16^4*
kParentalFlags_DisableDiagnostics	0x00020000	131072	*2 * 16^4*
kParentalFlags_AllowITunesUAccess	0x00040000	262144	*4 * 16^4*
kParentalFlags_RequireEncryptedBackups	0x00080000	524288	*8 * 16^4*

Let's say your desired AdminFlags settings are - lock iTunes, Disable Mini Store, and Disable Plugins. In hexadecimal you'd add 0x00000001 + 0x00000400 + 0x00002000 to get 0x00002401. In decimal this equates to 1 + 1024 + 8192 or 9217.

The kParentalFlags_Locked flag is important to note. Without setting it, unprivileged Windows users on a client machine have the ability to modify the parental control settings. If you hope to control the way iTunes is used and not used in your environment, always add the 0x00000001 (1).

There are a couple different ways to apply these registry settings via group policy. The easiest method is to use a logon/logoff batch script. This script could use the command line tool 'reg.exe' in a syntax like:

reg add "HKLM\Software\Apple Computer, Inc.\iTunes\Parental Controls\Default" /v AdminFlags /t REG_DWORD /d 0x00000001 /f

The 'regedit.exe' command could also be used with an exported registry key/setting file.

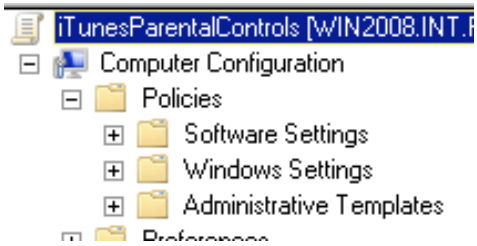
A more elegant way to apply the settings is to massage them into something called an ADM template that can be directly exported into the Group Policy Management application, look for a few links in the references that expand on ADM's and how to create them. Using a tool called 'RegToAdm' I've written such a file for others to utilize. For the interested party, 'RegToAdm' comes with 'NUTS' - see the reference section for a link. The ADM looks something like this:

```
CLASS MACHINE
CATEGORY "SOFTWARE\Apple Computer, Inc.\iTunes\Parental Controls\Default"
        POLICY "AdminFlags"
        KEYNAME "SOFTWARE\Apple Computer, Inc.\iTunes\Parental Controls\Default"
                PART "AdminFlags" NUMERIC
                        VALUENAME "AdminFlags"
                        MIN 1
                        MAX 1048575
                        DEFAULT 1
                END PART
        END POLICY
END CATEGORY
```

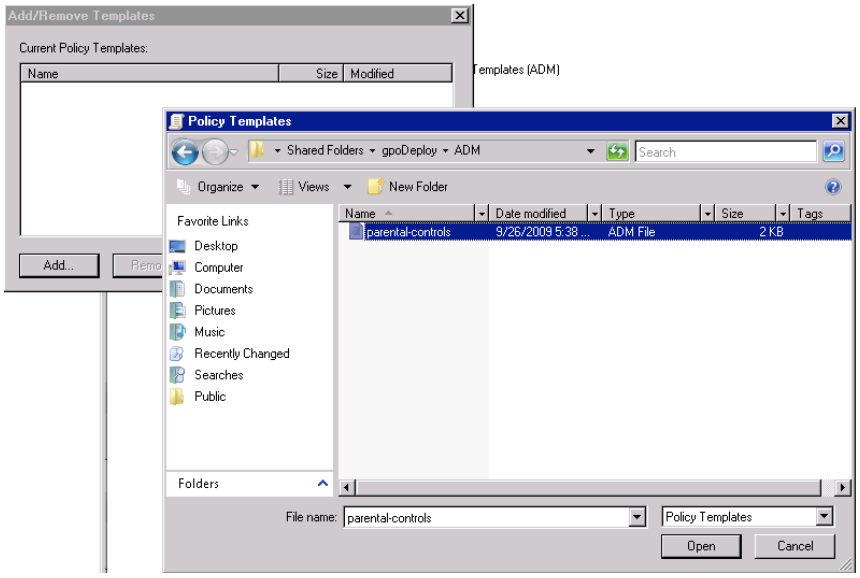
I removed a lengthy explanation comment for the purposes of brevity, but you get the idea. The most interesting thing about the ADM file is the **lack** of hexadecimal flag values. Turns out it's considerably easier to pass decimal values via an ADM template to a client machine's registry. The minimum decimal value for AdminFlags is 1 and the maximum is the sum of all the flags, 1048575. It is possible to select an unsupported value for AdminFlags because the ADM template forces incrementation by units of 1 for the registry key value. So be careful to define the settings you want based on the table above, add the decimal values, and choose the right value.

To make use of this ADM template, save it to a known & secure location on your server. The template does not have to live in a shared folder. Open the Group Policy Management application and follow the previously documented steps for creating & linking a new GPO. Name this GPO something like 'iTunes Parental Controls'.

Editing the 'iTunes Parental Controls' GPO will be unique from the previously created GPOs. Expand the 'Policies' folder beneath 'Computer Configuration' to reveal the 'Administrative Templates' folder.

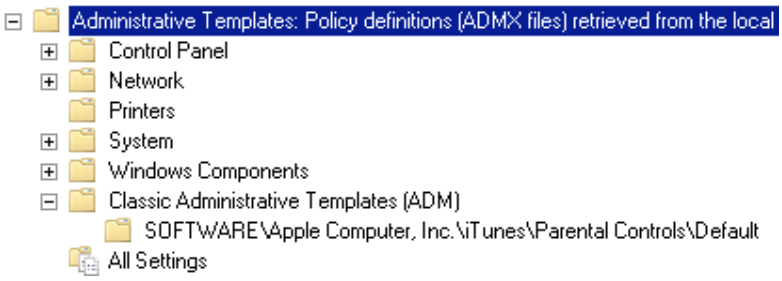


Right click on 'Administrative Templates' and select 'Add/Remove Template...'. In the ensuing popup window, click the 'Add' button.

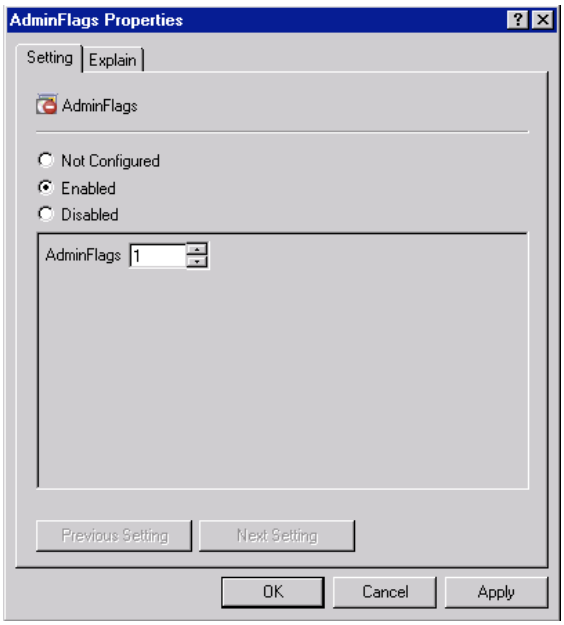


Navigate to the location in which you saved the ADM template, it doesn't need to be a network path. Select it and click the 'Open' button. Your ADM template will now appear in the 'Add/Remove Templates' list and you can click the 'Close' button.

In the left hand navigation pane, expand the 'Administrative Templates' folder to reveal the 'Classic Administrative Templates' folder. Expand it to reveal our custom template.



Highlight the 'SOFTWARE\Apple Computer, Inc.\iTunes\Parental Controls\Default' template and double click the 'AdminFlags' attribute in the right hand window pane.



In the ensuing popup window, check the radio button for ‘Enabled’ and choose your desired **decimal** AdminFlags value. Click the ‘OK’ button.

Follow the previously defined process for installing with elevated privileges and enabling increased logging. Set the GPO link order to give ‘iTunes Parental Controls’ the lowest value. You want to apply the settings after the application is installed. Choose a machine, user, or group from AD on which this GPO will be enforced. Do the gpupdate/ gpresult dance, then reboot your client.

Of course all of these GPOs can be created, ordered properly, and applied to your client machines in one fell swoop if you like. Just make sure the GPO link order installs the MSIs and Parental Controls in this order:

1. AppleApplicationSupport (AAS)
2. AppleMobileDeviceSupport (AMDS)
3. QuickTime
4. iTunes
5. iTunes Parental Controls

Future directions include updating the ‘classic’ ADM template to a modern ADMX template. Please feel free to offer comments and criticisms.

References

1.Adobe Acrobat 8 for Microsoft Windows Group Policy and the Active Directory Service - http://www.adobe.com/devnet/acrobat/pdfs/gpo_ad_8.pdf
2.AppDeploy.com iTunes - http://www.appdeploy.com/packages/detail.asp?id=1369
3.AppDeploy.com QuickTime - http://www.appdeploy.com/packages/detail.asp?id=520
4.How to Write a Simple .Adm File for Registry-based Group Policy - http://technet.microsoft.com/en-us/library/cc738443(WS.10).aspx
5.iPhone Enterprise Deployment Guide - http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdf
6.Locale ID Chart - http://msdn.microsoft.com/en-us/library/0h88fahh(VS.85).aspx
7.Microsoft Windows SDK for Windows 7 - http://www.microsoft.com/downloads/details.aspx?FamilyID=c17ba869-9671-4330-a63e-1fd44e0e2505&displaylang=en
8.MSI Editing with Orca - http://msdn.microsoft.com/en-us/library/aa370557(VS.85).aspx
9.Network Utilities Set (NUTS) - http://yizhar.mvps.org/files/NUTS.EXE
10.Show Hidden Program or System Files - http://www.microsoft.com/windowsxp/using/helpandsupport/learnmore/tips/hiddenfiles.mspx
11.Writing Custom ADM Files for System Policy Editor - http://support.microsoft.com/kb/225087
12.Windows OS Managed Client: How to manage iTunes control features - http://support.apple.com/kb/HT2102