Autodesk University 2008

Learn to Hack AutoCAD® Installations for Custom Deployments

Darren Young – Southland Industries / Los Angeles, San Francisco, Las Vegas, Reno, Washington DC

CM215-2 Learn what it takes to create, maintain, install, and even hack software deployments—from figuring out what to call and when, to automated installations of software and service packs, to hacking out your own custom deployment with VB Script or Batch files. This class will teach you what you need to know to automate most of your AutoCAD installations, service pack, and hot fixes.

About the Speaker:

A Midwestern transplant now based in Southern California, Darren has held a variety of positions over the last 15 years, such as CAD/CAM engineer, CAD administrator, and CAD/CAM systems developer. Currently Darren is the Systems Integration Manager for Southland Industries, one of the top 10 mechanical contractors in the United States. While Darren's true interest is the automation of manufacturing systems, his experience ranges from manufacturing to architecture, and this has led him to projects varying in scope from dress patterns to gas turbine piping. He founded a consulting and development business called Minnesota CADWorks, has been a technical editor for the 2002, 2004, and 2005 versions of the AutoCAD Bible (by Autodesk Registered Author Ellen Finkelstein) and has been a regular contributing author for Inside AutoCAD since 2001.

Work / dyoung@southlandind.com
Home / darren@mcwi.com

Installation 101 – Installation types

1. EXE based installs – Command line options (if available) dependent on author's implementation
    - Command line switches as defined by the author
    - INI file referenced from EXE with parameters used for EXE installation
2. MSI (MSP) based installs – Command line options based on Microsoft's MSIEXEC.EXE installation.
    - Type "**MSIEXEC.EXE /?**" from DOS command prompt or Windows "Start->Run" to view list of command line options.
    - Go to http://www.microsoft.com and search on "**MSIEXEC Command line**" for more detailed documentation.
3. Hybrid (MSI/MSP embedded within EXE) – Command line options (if available) depends on authors implementation
    - Special switch to pass MSIEXEC command line to embedded MSI file
    - INI file referenced from EXE with parameters used for MSI installation
    - Often, a self-extracting compressed file (but not always). Attempt to open with WinZIP/WinRAR type of compression utility to extract contents of installation.

MSIEXEC.EXE - Command line options

msiexec /Option <Required Parameter> [Optional Parameter]

Install Options
</package | /i> <Product.msi>
Installs or configures a product
/a <Product.msi>
Administrative install - Installs a product on the network
/j<u|m> <Product.msi> [/t <Transform List>] [/g <Language ID>]
Advertises a product - m to all users, u to current user
</uninstall | /x> <Product.msi | ProductCode>
Uninstalls the product

Display Options
/quiet
Quiet mode, no user interaction
/passive
Unattended mode - progress bar only
/q[n|b|r|f]
Sets user interface level
n - No UI
b - Basic UI
r - Reduced UI

f - Full UI (default)
/help - Help information

Restart Options
/norestart
Do not restart after the installation is complete
/promptrestart
Prompts the user for restart if necessary
/forcerestart
Always restart the computer after installation

Logging Options
/l[i|w|e|a|r|u|c|m|o|p|v|x|+|!|*] <LogFile>
i - Status messages
w - Nonfatal warnings
e - All error messages
a - Start up of actions
r - Action-specific records
u - User requests
c - Initial UI parameters
m - Out-of-memory or fatal exit information
o - Out-of-disk-space messages
p - Terminal properties
v - Verbose output
x - Extra debugging information
+ - Append to existing log file
! - Flush each line to the log
* - Log all information, except for v and x options
/log <LogFile> - Equivalent of /l* <LogFile>

Update Options
/update <Update1.msp>[;Update2.msp]
Applies update(s)
/uninstall <PatchCodeGuid>[;Update2.msp] /package <Product.msi | ProductCode>
Remove update(s) for a product

Repair Options
/f[p|e|c|m|s|o|d|a|u|v] <Product.msi | ProductCode>
Repairs a product
p - only if file is missing
o - if file is missing or an older version is installed (default)
e - if file is missing or an equal or older version is installed
d - if file is missing or a different version is installed
c - if file is missing or checksum does not match the calculated value
a - forces all files to be reinstalled
u - all required user-specific registry entries (default)
m - all required computer-specific registry entries (default)
s - all existing shortcuts (default)
v - runs from source and recaches local package

Setting Public Properties
[PROPERTY=PropertyValue]

CD/DVD Image installation – Major Autodesk Products

Most major Autodesk products can be installed without using the graphical user interface. Installations can be initiated from a standard DOS batch file or VBScript file.

For AutoCAD and AutoCAD based verticals, explanation of command line switches and syntax can be found in the in the “Network Administrators Guide” in the product CD/DVD. Browse for the “Docs” folder and look for the “acad_nag” file.

For non-AutoCAD based verticals, file name and location can and does vary (e.g. Inventor) Often, information is less detailed, more confusing and in some case, may be missing completely and/or not supported (e.g. Revit). Search the product CD/DVD and view any PDF, HTM/HTML, CHM, TXT. Additional searching on Autodesk discussion groups or web resources can be helpful. Use others experience and examples as a basis for your quest for information and examples. Most importantly, don't forget your reseller and/or subscription support.

Deployments & Administrative Images

Administrative Images are copies of the application installation files from your CD/DVD media. Deployments are named, customized versions of the application installation. An administrative Image will be created with the first deployment made with the deployment wizard.

Deployments are perhaps the easiest way to perform a silent install of Autodesk products. For some products, it's the only way. Deployments eliminate the need for complex VBScript or batch files to deploy the application with your required options.

- Use the deployment wizard included in your application CD/DVD to create deployments in an administrative image. If no administrative image exists, one will be created as well.
- Applications consisting of multiple CD/DVDs can be copied to a server/local computer to eliminate problems during deployment creation where the additional media is not being read.
- When copying multiple CD/DVDs to a server/local computer, they typically can be copied into the same folder. Some files are duplicated on the different disks of media. They typically can be overwritten safely.

Updates (formerly Service Packs)

Updates and service packs can always be installed separately from the application. This can be done manually, from a batch file, or VBScript immediately after the installation of the primary application.

Updates are typically distributed as an MSP file or an EXE. When distributed as an EXE, calling the EXE will usually install the update. The EXE often will support the command line switch "/q" to suppress the "Successfully Completed" dialog at the end of the update installation which is useful for silent or unattended installations from a batch file or VBScript.

```
Autocad2009lockedsp1.Exe /q
```

Updates can be applied to the primary install image rendering a second (or more) installation unneeded. You must use an MSP file from the Deployment Wizard to apply the update. When distributed as an EXE, the command line switch "/e" can be used to extract the MSP from the EXE.

```
Autocad2009lockedsp1.Exe /e Autocad2009Sp1.Msp
Autocad2009lockedsp1.Exe /e "c:\my service packs\Autocad2009Sp1.Msp"
```

Consult the readme document for the update to verify if the command line switches are valid, determine their exact use and see if there are any other switches available that may be of use to you.

2007 (and earlier) series Autodesk products

- Use Deployment Wizard to modify and existing administrative image. All deployments under this administrative image will have the update/service pack(s) applied.
- Use Deployment Wizard to create a new administrative image. All deployments under this administrative image will have the update/service pack(s) applied.

2008 (and later) series Autodesk products

- Use the "Modify <*deployment name*>" shortcut to modify either that specific deployment, or the entire administrative image.
- Use the Deployment Wizard to create a new administrative image which includes the update/service pack.
- When applying an update/service pack to a specific deployment, only that deployment within the administrative image will have the update/service pack applied. Each additional deployment needing the update/service pack will need to have it applied separately.
- When applying a update/service pack to the entire administrative image, all deployments within that administrative image will have the update/service pack applied.
- If a update/service pack is applied to a one or more deployments within an administrative image, it no longer can be applied to the entire administrative image as whole.
- If a update/service pack is applied to an entire administrative image, it can no longer be applied individually to specific deployments.

Hot fixes

Hot fixes are very specific to a particular software defect or update. Automation of hot fixes will vary depending on the hot fix. Refer to the documentation to determine the best practice for installation.

Installation can consist of an EXE which may or may not offer command line switches, a self extracting EXE which will allow you to extract an MSI/MSP or other files for processing. It may also consist of only copying new files into the installation folder structure on the client PC.

Hot fixes which consist of updated files found in the installation can often be copied into the administrative image but not always. Verify the hot fix is needed after all service packs have been applied. Documentation regarding which hot fixes are included or not included in an update/service pack is often sparse or non-existent.

It's recommended to NOT include a hot fix unless it specifically addresses an issue that affects you. If you do include hot fixes or updates and they are not able to be applied to the administrative image, standard batch files or vbscript can be used to efficiently install them.

Manually Tuning Installations

MSI files can be installed directly be calling them directly (double clicking). However, no prerequisites will be installed and you can't override the setup dialogs.

Acad.Msi

They can be called with installer override options specified by calling the installation package as a parameter to Microsoft's installer Msiexec.exe. Prerequisites will not be installed. Refer to Msiexec.Exe command line parameters for explanation of switches.

*Msiexec.exe /i Acad.Msi /l*v c:\logs\acad.log /qb*

The preferred method for installing, is be calling the Setup.Exe program. Calling Setup.Exe by itself will launch the standard installation wizard which we don't want. You can call Setup.Exe and specify the deployment INI file to launch a customized deployment. Include the "/MD" switch to "modify" the "deployment".

\\server\deploy\AdminImage\Setup.exe \\server\deploy\AdminImage\AcadAU08.ini
\\server\deploy\AdminImage\Setup.exe /md \\server\deploy\AdminImage\AcadAU08.ini

As an alternative to passing a command line argument to the Setup.Exe program, you can make a copy of Setup.Exe using the same name as the INI file which describes the deployment. By default, the Setup program will use the INI file with the same name if called with no arguments.

\\server\deploy\AdminImage\AcadAU08.exe

Using Deployments, you can have several, custom installations. You can also, further tune those configurations by editing the INI file related to deployment or even the base product installation if you aren't using deployments. Keep a backup of the INI file in the event you break the installation's INI. Examine the entire file and try to get a sense of what its doing. No two are the same and 2007 and earlier products differ greatly from 2008 and later products. Depending on what's going on and how it's packaged, there may be a variety of things you can do such as…

Change log file locations…

LOG=%tmp%\DWF Viewer Install.log
→ *LOG=\\server\logs\DWF Viewer Install.log*

Make a component less "silent"…

EXE_PARAM=/quiet /norestart

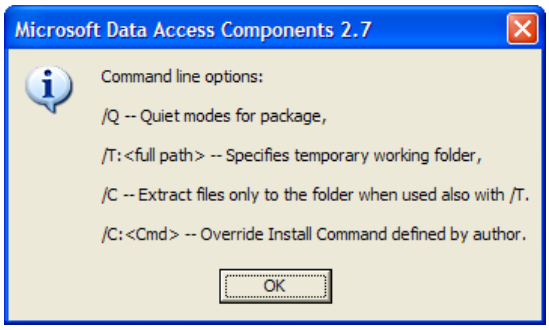
→ *EXE_PARAM =/norestart*

Make a log file less verbose…

*INSTALL_CMD_ARGS=/L*v %temp%\DWGVIEWRInstall.log*
→ *INSTALL_CMD_ARGS=/L*v %temp%\DWGVIEWRInstall.log*

Don't install some prerequisites you know are already there…

PREREQUISITE=OS;MSI;DOTNET20;DOTNETLANG;VBA;VBALANG;DIRECTX;MSXML;DWFV;FLASH;WMF;MDAC;IE
→ *PREREQUISITE=OS;MSI;VBA;VBALANG;MSXML;DWFV;FLASH;WMF;MDAC;IE*



Be very cautious about removing prerequisites. They may cause components further in the install to fail. If a prerequisite is not included, you can also remove those data files from the administrative image if you wish.

For prerequisites packaged as EXE files, try calling the various prerequisites' EXE file with the "/?" command line switch to see if additional command line options are available. In this example, you have a couple switches that will extract the EXE's contents. Try extracting it's contents and examine the files to see what other options may exist that you might want to leverage. Repeat as needed.

At the very least, examining the setup INI files gives you a good idea how to install, and in many cases uninstall the various components should you need to package and deploy them separately as you would if pushing out a deployment via group policy.

Installation – Minor Autodesk Products

As with major Autodesk product installations, the key to fine tuning your installations of minor Autodesk products like TrueView, or AOEMView, is to use customized INI files. Other products like Design Review, or the DGNImporter, JTIMporter or Batch Print updates for Design Review, use command line switches for product.

The TrueView & TrueConvert programs from Autodesk are (were) distributed as EXE files. Using a program like WinZip or WinRAR, opened these EXE files and you'll see that they are really self-extracting archives. Extract the contents of the archive to a folder and look for the Setup.Ini. Make a copy of the Setup.Ini file with a new name, such as Setup-Silent.Ini. It's in this INI file that we'll make our changes.

TrueConvert (2007 version)

Locate this line in the [SETUP] section on the INI file. It looks promising.

SETUP_UI_MODE=;q - no UI; qb - basic UI; qf - full UI

The “q” settings look similar to those used with MSIEXEC. Let’s try the standard “qb” setting like this…

SETUP_UI_MODE=qb

Try running the customized install by making a copy or Setup.Exe using the same name as your custom INI or by calling Setup.Exe and passing the Setup-Silent.Ini files as a command line argument.

You’ll notice this doesn’t appear to work. The TrueConvert installer must override this setting, or this isn’t implemented. We’ll have to find another way to make this an automatic install. Look for the following line in the [BDC] section of the Setup-Silent.Ini file…

*INSTALL_CMD_ARGS=/L*v "%temp%\DWGTrueConvert Install.log"*

You’ll also notice that this section is installing from an MSI not an EXE so we know the standard MSIEXEC command line options will apply. Try changing the line to the following, adding the “/qb” command line switch to use a “basic” interface…

*INSTALL_CMD_ARGS=/L*v "%temp%\DWGTrueConvert Install.log" /qb*

Now try running the customized install again as before. This time it works and you have a single click, automated install of TrueConvert.

TrueView (2007 version)

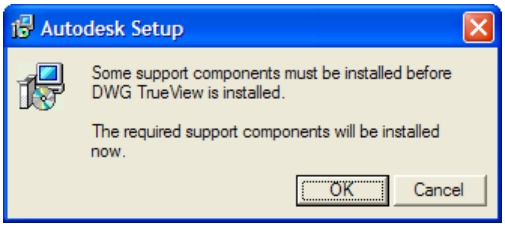
Locate the Setup.Ini and make a copy as Setup-Silent.Ini as we did for TrueConvert earlier. Recall the “SETUP_UI_MODE” in TrueConvert that didn’t appear to work? It’s not even in TrueView. Let’s look and see if we can make the same change in TrueView. Look for the following line…

*INSTALL_CMD_ARGS=/L*v %temp%\DWGVIEWRInstall.log*

Again, we’re dealing with an MSI based install so we can add the “/qb” switch as before like the following…

*INSTALL_CMD_ARGS=/L*v %temp%\DWGVIEWRInstall.log /qb*

Now try running the custom install as before, my calling Setup.Exe and passing the custom INI file name as a command line argument or by making a copy of the Setup.Exe file that uses the same name as the custom INI file.



This time, we’re prompted with a message. Something else must be going on because after this message, the install proceeds as we’d expect. Let’s look in the INI file again for the following line in the [SETUP] section…

SETUP_STARTUP_MESSAGE=YES

This looks like it might apply. Let's change it to look like this…

SETUP_STARTUP_MESSAGE=NO

When we try our custom install now, it now eliminates the startup message and again, gives us the automated installation we were looking for.

InventorView 11

InventorView 11 is another MSI based install. It's not distributed as an EXE but a ZIP so we still extract the contents using a compression utility line WinZip/WinRAR. Locating the INI file we see it looks structured very similar to that of TrueView 2007. We can add the "/qb" switch to the INSTALL_CMD_ARGS line in the [INVENTORVIEW] section of the INI file and if we look at the SHOW_STARTUP_MESSAGE line in the [SETUP] section, we see it's already set to "NO".

TrueView 2008 (and later) & InventorView 2008 (and later)

TrueView is distributed as an EXE and InventorView is distributed as a ZIP file. Both need to be extracted. For 2008 and later products, Autodesk completely revamped their installation and setup infrastructure which means yet another method to perform a silent install. This time, the process is a little easier.

As before, make a copy of the Setup.Ini file for the application and open it in Notepad. Let's find the section [AOEM] where the viewer and not the prerequisites get's installed. It's an MSI based install so MSIEXEC command line options should apply. There are two lines that look like they might be what we're looking for.

EXE_PARAM=
USE_EXTERNAL_UI=YES

However, adding the "\qb" to the EXE_PARAM line and/or changing the USE_EXTERNAL_UI to "NO" doesn't do what we want. This is because in this case, the Setup program implements it's own user interface that wraps around all the other component installations. These settings would only apply to the component but they don't affect the bug Setup user interface that's already been displayed by now.

Luckily, for 2008 or later based installs, it's a lot easier than that. Look in the [SETUP] section, scroll down to the following line…

UI_MODE=

Now, let's just add a setting like the following…

UI_MODE=SILENT

Now let's try running our custom installation as we have before. This time it works. Most 2008 products will work just like this. Even major applications like AutoCAD will work line this

however, you'll get the default install settings which is why creating a deployment for the major applications is the process you'll use for those products.

Do NOT simply edit the Setup.Ini file. Always make a copy with a different name. 2008 and later based products cache a copy of the Setup.Ini file locally (not the INI file you used to install from) which will make using "Add/Remove Programs" silent because it too will contain that setting. Because "Add/Remove Programs" needs to know if you are going to Reinstall, Repair, or Remove the application and as such, if the Silent flag would be there, you aren't prompted for the action and "Add/Remove Programs" will do nothing.

DesignReview 2008 & DesignReview Updates

DesignReview 2008 and the various updates to it are all distributed as EXE packages. When attempting to open those files with a utility like WinZip or WinRAR, you'll see that they are not standard self-extracting compressed files. The next step is to call the EXE passing the "/?" command line switch to it to see if there's any help associated with those programs. In this case, there is a dialog that's displayed showing you the various command line switches available.

You'll notice a switch "/s" which hides the initialization dialog. That's only the initialization dialog, not the entire install UI. There's also a "/v" switch that passes parameters directly to the MSIEXEC installer which is why the "/s" switch tells you to use "/s /v/qb" to a silent install line this…

<product>.Exe /s /v/qb

In this case, "/s" hides the setup initialization dialog, and "/v" passes the "/qb" switch directly to MSIEXEC which installs the MSI package (even though we can't extract it) with a basic user interface. Note that you are not limited to just the "/qb" switch to pass to to MSIEXEC. You can pass any valid MSIEXEC switch like the following which includes logging and a log file specification…

*<product>.Exe /s /v"/qb /l*v c:\mylogs\product.log"*

Notice that because there are multiple switches that need to be passed to MSIEXEC and the setup program doesn't always know what's being specified as its switches as opposed to those it should sent to MSIEXEC, include the MSIEXEC switches in quotes.

DesignReview 2009

DesignReview 2009 is distributed as an EXE but it can be extracted. Once extracted, you can look at the Setup.Ini file and see it uses the newer Autodesk installer method meaning you can make a simple modification of *"UI_MODE=SILENT"* like most of the major products. 2009 based plug-ins for DesignReview use the preview method above and the update/service pack for DesignReview is distributed as an MSP file.

Uninstall / Removal

Uninstalling your application can be automated as well, such as uninstalling a previous version of software prior to installing the next version. Manual uninstalls can be performed by using Add/Remove Programs, or rerunning Setup.Exe and choosing the Uninstall option.

Automated silent uninstalls can be performed in one of 3 ways…

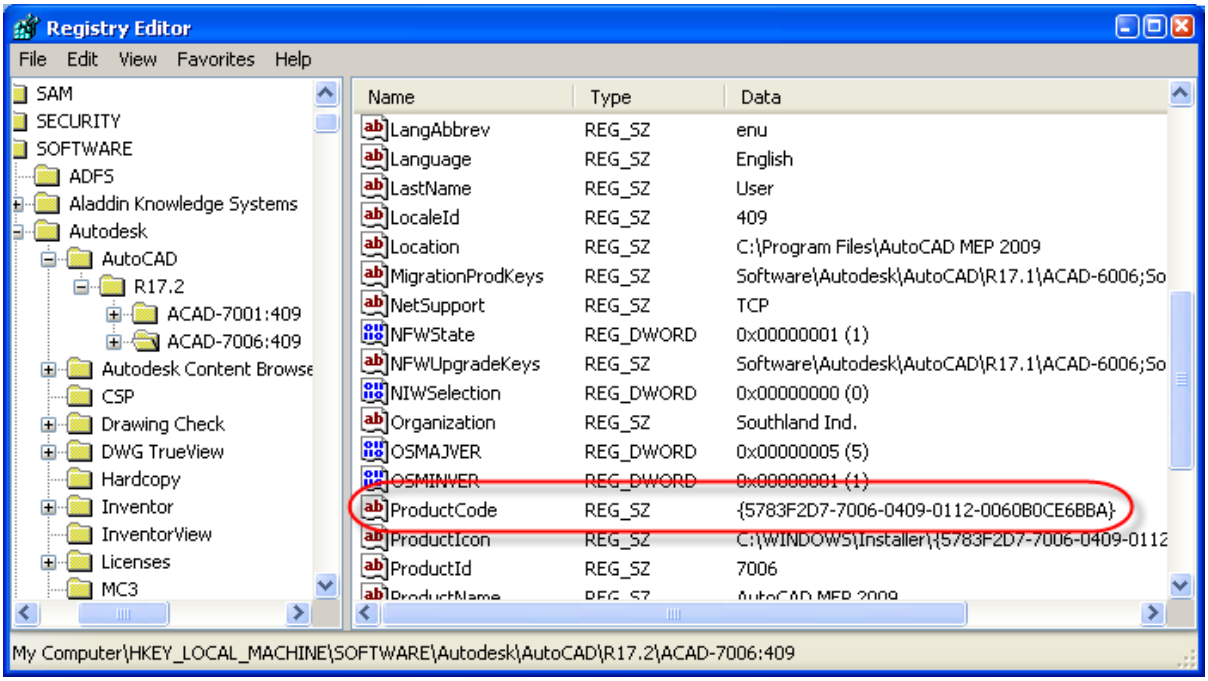
1. Calling Msiexec.Exe and specifying the product code (Global Unique Identifier or GUID) along with any other command switches that are appropriate…

   Msiexec /x{5783F2D7-6001-0409-0002-0060B0CE6BBA} /qb

   The product code can be found by looking in the registry for the ProductCode value under the registry key…

   HKEY_LOCAL_MACHINE\SOFTWARE\Autodesk\<product>\…

   The following image shows the ProductCode value for AutoCAD MEP 2009.



2. Calling Msiexec.Exe and specifying the cached local MSI file along with any other command switches that are appropriate…

   Msiexec /xC:\Windows\Installer\1878910.msi /qb

   Note that the name of this cached local MSI file is not consistent, across computers or even the same computer if the application is installed, uninstalled and installed multiple times. As

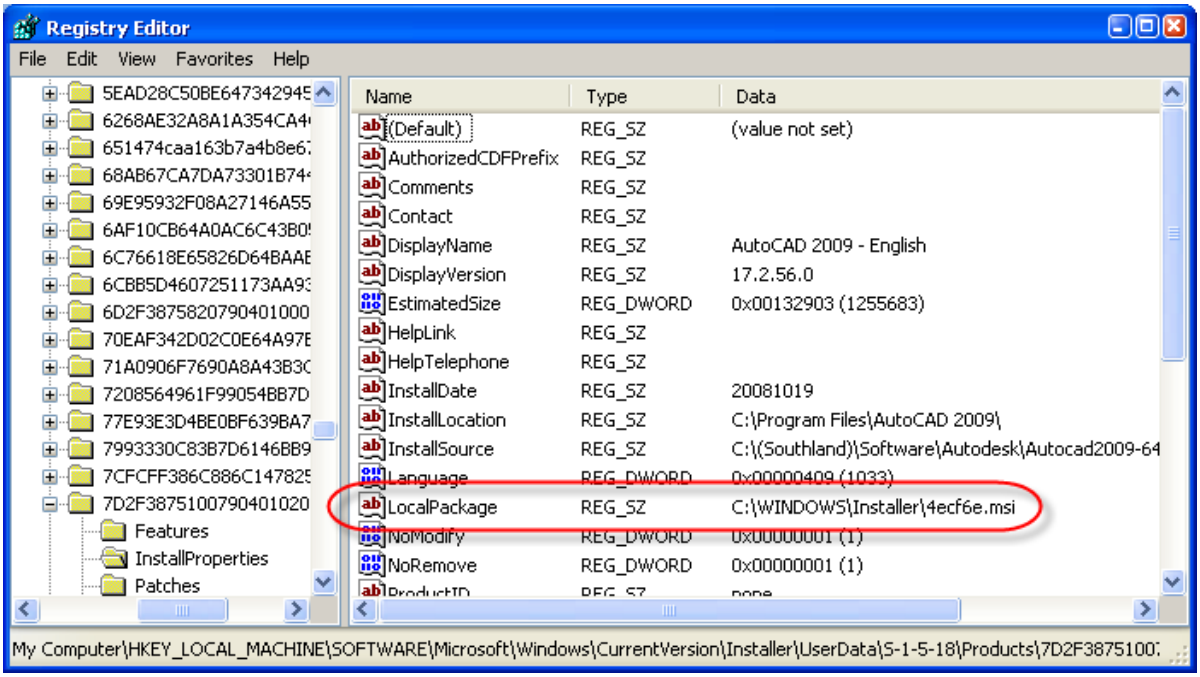
such, it's not recommended for enterprise wide software removal. None the less, it can still be used if you find a need to do so.

Also note, the cached local MSI is typically found in the C:\Windows\Installer folder. This folder is a System Hidden folder and may not be directly browsed to graphically.

The cached local MSI file can be found by looking in the registry for the LocalPackage value under the registry key…

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData

Once at this registry key, search for the name of the product as it appears in "Add/Remove Programs". Under this key, look for the value LocalPackage as shown in the following image…



3. Calling Msiexec.Exe and specifying the deployment MSI file along with any other command switches that are appropriate…

Msiexec /x\\server\deploy\AdminImage\Acad.Msi /qb

This is perhaps the more reliable and easiest method to uninstall your application but does require that you have access to the CD/DVD media or the administrative image.

Repair / Reinstall Installations

In addition to installing and removing software, you can also repair or reinstall installations. This can be a nice way to have your users troubleshoot their own installation if they suspect trouble. The following line uses the "/f" switch to repair a Dwg TrueConvert installation, specifying

several options “aums” for how the repair should happen as well as an additional flag “/norestart” to suppress the “Restart your computer Yes/No” prompt.

Msiexec /faums DwgTrueConvert.Msi /norestart

Update/Service Pack Installation

While updates and service packs can be included in the deployments for major products, they can also be applied separately. Using the MSP with the MSIEXEC program, they can be installed using the “/update” command line switch along with any other switches you desire. The following line would update AutoCAD 2009 with Service Pack 1 using a basic, progress bar and cancel button dialog for an unattended installation.

e.g. msiexec /update acad2009sp1.msp /qb

Many of the updates and service packs are distributed as EXE files. The MSP that was used above can typically be extracted using the “/e” command line switch. There’s often a “/q” switch to quiet the display of the confirmation dialog at the end of the installation if you prefer to install directly from the EXE file. However, the MSP will be needed to apply the update or service pack to a deployment or to install the MSP using MSIEXEC with command line switches that you define. Consult the readme file that comes with the update/service pack for complete information on the various switches that may or may not be used with the update/service pack and how they are used.

MSI Modification

While we covered in detail how to modify the INI files to create a custom installation, you can also modify the MSI file itself in some cases to achieve even more specific results. Modifying an MSI file isn’t a trivial matter, the MSI structure and intent is complex and typically a task for a software packaging engineer or developer. However, with a little trial and error and careful looking, you can leverage it to your advantage.

To accomplish this task, you can use a free tool from Microsoft called Orca. Information in downloading and installing Orca can be found at http://support.microsoft.com/kb/255905. You can also search on the term Orca from Microsoft’s web site and find a wealth of information related to this utility.

The example we’re use for using Orca involves Autodesk’s TrueView application. It’s a free DWG/DXF viewer. As most CAD Administrators likely have found, installing TrueView across an enterprise can be problematic because it doesn’t offer full profile support like AutoCAD. This means, if you want to point TrueView to a network location for Plotter configurations or Plot Style tables, you’ll need to manually configure those locations.

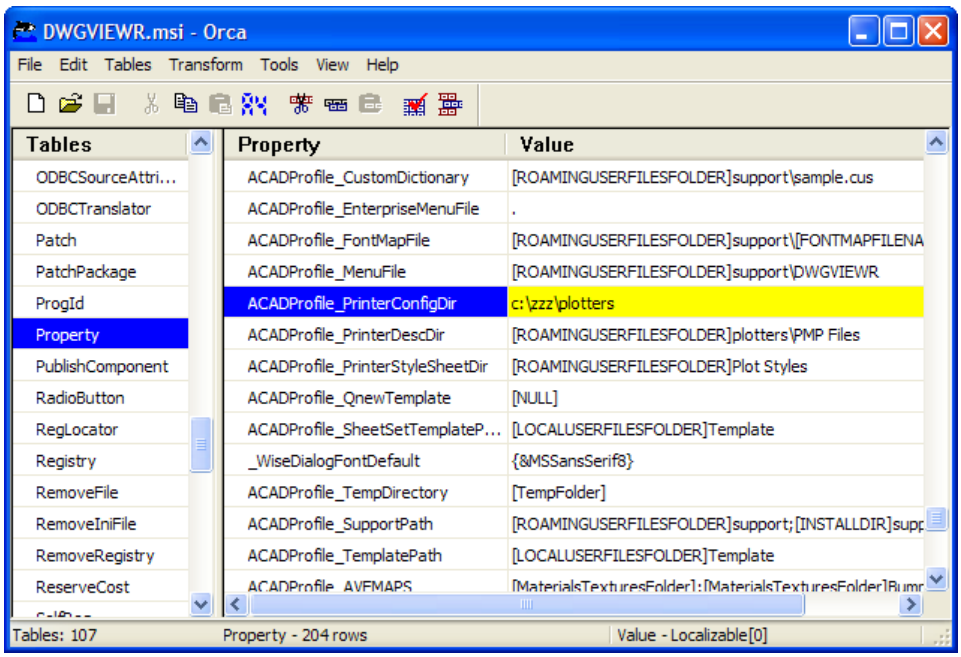
As if this wasn’t bad enough, any different user login on that system would have to also manually make these changes. Each time a new user launches TrueView for the first time, it launches a secondary installer which sets the profile specific information for TrueView for that user. This includes the Plotter and Plot Style location.

To work around this, some people use a registry file to modify these settings directly in the registry. Your registry file might look list…

[HKEYHKEY_CURRENT_USER\Software\Autodesk\DWG TrueView\R6\DWGVIEWR-7001:409\Profiles\<<Unnamed Profile>>\General]
"PrinterConfigDir"="\\\\Server\\share\\directory\\PLOTTERS"
"PrinterDescDir"="\\\\Server\\share\\directory\\PLOT STYLES"
"PrinterStyleSheetDir"="\\\\Server\\share\directory\\PLOT STYLES"
"Support"="%appdata%\\Autodesk\\DWG TrueView 2009\\R6\\enu\\support;%programfiles%\\DWG TrueView 2009\\support;%programfiles%\\DWG TrueView 2009\\fonts;%programfiles%\\DWG TrueView 2009\\help;\\\\Server\\share\\directory\\;\\\\Server\\share\\fonts\\;"
"DefaultConfig"="\\\\Server\\share\\directory\PLOTTERS\\Myplotter.pc3"
"PlotToFilePath"="\\\\Server\\share\\plotfiles\\"

The trouble with this approach is that it requires you to again, manually import the registry file or to import it automatically with a login script. The trouble with a login script, is that it may import the registry before the user ever runs TrueView. If this happens, TrueView might think it already setup all the user profile specific information and not add the other settings it needs. You'd need to add some logic in your login script to only import the registry file if it determined that TrueView was already started once and the profile specific information was already there. It would then import the registry modifying those incorrect settings TrueView defaults to. Even this means that the very first launch of TrueView doesn't have the custom settings, unless you logged back off and on, then restarted TrueView a second time.

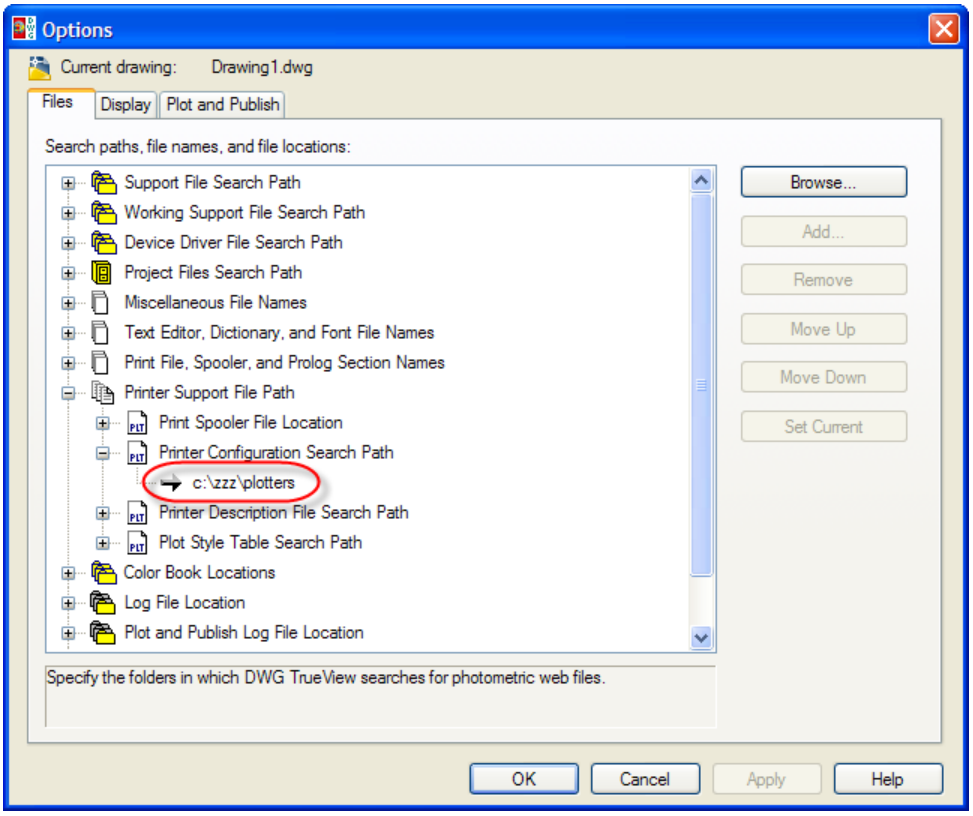
This is where Orca comes into play. Launching Orca, you can then use the File->Open command and select the TrueView MSI. Next, we'll search the MSI tables for settings that reference the "Plotters" folder. It shows up many places but under the "Property" table, we find the settings we're looking for. In the following image, you see where we edited the "ACADProfile_PrinterConfigDir" property to reference out network location for plotter configurations, "n:\mycad\plotters". There are other properties as well that we haven't edited yet for PMP file location as well as Plot Style tables and others.



Have a look around and see the various things in the MSI. Most should not be edited, and for a major product like AutoCAD where you can create a deployment, it's recommended you use other processes than using the Orca utility if at all possible. But be aware of it's existence and use for those cases where it makes sense like with TrueView.

Once you've finished editing the MSI, save the changes and exit the utility. When you now install TrueView, it will install with your configured network Plotter, Plot Style, and other paths pre-configured. No login scripts importing registry files or manual configuration for each user.

You should note, however, that this edit to the MSI does NOT eliminate the creation of the default Plotters folder TrueView creates under your Windows user profile. And it does NOT create the folder in the location you specified. This edit only changes the location the TrueView configuration points to, whether it exists or not. We want result because the complexity of an MSI database and our lack of knowledge in what the developers of this MSI had in mind, we want to perform as minimally invasive edits as possible.



Installation Scripts (VBScript) and Batch Files

Now that you've learned how to hack at a deployment and setup INI files to make a single click automated installation (or uninstall/repair) the next step is to tie all those things together to perform the type of installation you want, which may be as complex as uninstalling an old version, installing a new version, adding service packs, hot fixes, other utilities, and even copying custom files or creating custom shortcuts.

This type of work can be performed with VBScript files, batch files or even a combination of both. Learning VBScript or Batch file programming is beyond the scope of this course, but it's not hard to pick up the basics and even those who aren't familiar with it, should be able to understand what's going on.

The key difference between VBScript and Batch files is that batch files use old DOS commands. Despite being an older process, some very powerful things can be done with batch files. Like reading a text files and processing the various contents of it. VBScript on the other hand is like an ASCII text version of Visual Basic. It's a lot more powerful, but harder to learn with more complex syntax.

On method to perform installations with either process is to create one batch file or VBScript that does everything you need. If you're only going to be doing one install, this is likely the best approach. Here's the a sample batch file that uninstalls AutoCAD 2007, TrueView2007, TrueConvert 2007, and installs AutoCAD 2008, SP1, and TrueView 2008…

```
REM -- Uninstall AutoCAD 2008 ---
Msiexec.exe /x{5783F2D7-5001-0409-0002-0060B0CE6BBA} /qb /l*v %temp%\uninstac07.log
REM --- Intstall AutoCAD 2008 ---
\\l5375\au\software\Acad2008\deploy\AdminImage\Setup.exe \\l5375\au\software\Acad2008\deploy\AdminImage\AcadAU07.ini
REM --- Install AutoCAD 2008 SP1 ---
Msiexec.exe /update "\\l5375\au\software\Acad2008\acad2008sp1.msp" /qb
REM --- Uninstall Trueview 2007 ---
Msiexec.exe /x{2CD6BBA0-17C8-4789-9B9B-B36F7E815F6A} /qb /l*v %temp%\uninsttv07.log
REM ---Uninstall Trueconvert 2007 ---
Msiexec.exe /x"\\l5375\au\software\DWG True Convert\Disk1\DWGTrueConvert.Msi" /qb /l*v %temp%\uninsttc07.log
REM --- Install DWG TrueView 2008 ---
"\\l5375\au\software\Trueview 2008\Disk1\Setup-silent.exe"
```

And one using VBScript (watch for word wrap)…

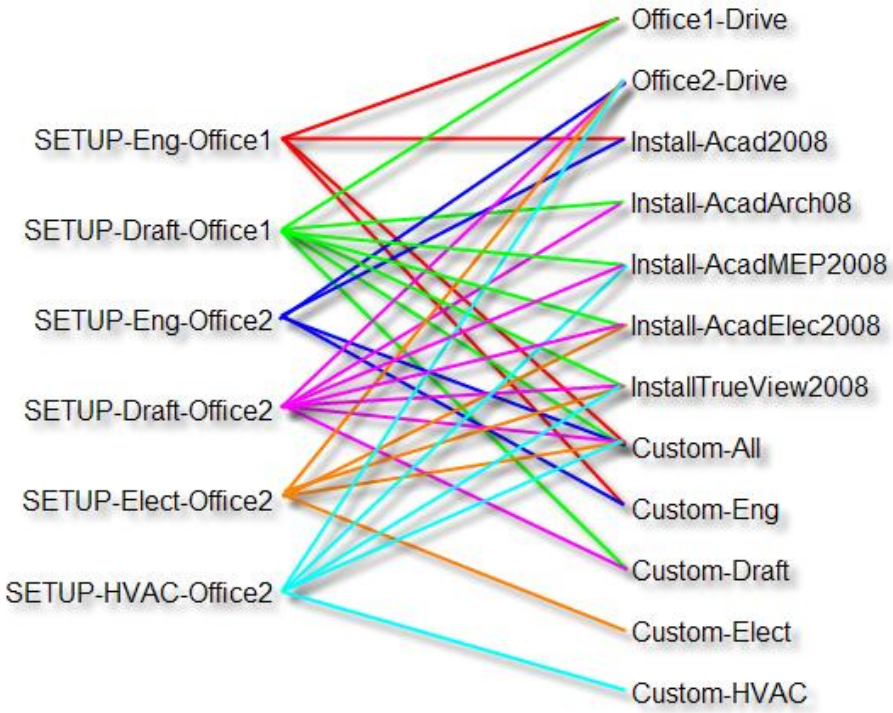
```
' --- Create WScript Shell Object ---
Set objWS = WScript.CreateObject("Wscript.Shell")
' -- Uninstall AutoCAD 2008 ---
objWS.Run("Msiexec.exe /x{5783F2D7-5001-0409-0002-0060B0CE6BBA} /qb /l*v %temp%\uninstac07.log"), 0, TRUE
' --- Intstall AutoCAD 2008 ---
objWS.Run("\\l5375\au\software\Acad2008\deploy\AdminImage\Setup.exe
\\l5375\au\software\Acad2008\deploy\AdminImage\AcadAU07.ini"), 0, TRUE
' --- Install AutoCAD 2008 SP1 ---
objWS.Run("Msiexec.exe /update \\l5375\au\software\Acad2008\acad2008sp1.msp /qb"), 0, TRUE
' --- Uninstall Trueview 2007 ---
objWS.Run("Msiexec.exe /x{2CD6BBA0-17C8-4789-9B9B-B36F7E815F6A} /qb /l*v %temp%\uninsttv07.log"), 0,TRUE
' ---Uninstall Trueconvert 2007 ---
objWS.Run("Msiexec.exe /x" & """\\l5375\au\software\DWG True Convert\Disk1\DWGTrueConvert.Msi""" & "/qb /l*v
%temp%\uninsttc07.log"), 0, TRUE
' --- Install DWG TrueView 2008 ---
objWS.Run("""\\l5375\au\software\Trueview 2008\Disk1\Setup-silent.exe"""), 0, TRUE
```

Perhaps a better method, if you need multiple software deployments of varying types, would be to break up your VBScript or batch file into separate files. It's a technique I typically use because I may need to install TrueView and AcadLT for a group of people, AutoCAD and TrueView to others, maybe a vertical like Inventor Series or Revit along with TrueView to others. Now what happens if there's a service pack for TrueView? Do I add the various VBScripts and batch files?

Instead, what I do is make separate VBScript and batch files for each product. Not only does the VBScript/batch files for each product install the product, I also have it do service packs (if they aren't part of the deployment, as well as hot fixes or updates related to that product. I then create "master" VBScript or batch files that call the other product specific VBScript or batch files in the order I want, in the combination I want. Structuring your installations VBScript or batch files in this way makes if very quick and easy to make a custom master installation on demand.

But don't get too carried away. Maybe you have 5 different AutoCAD deployments. Some are stand alone installations; others networked but reference different license servers or distributed servers in a different order. Maybe the customization within the various deployments is different between groups of users. There's no need to create separate VBScript or batch files for each AutoCAD deployment. You can call a single VBScript/batch file for the AutoCAD deployment from a master and pass as a parameter or argument the name of the deployment you want to use. This keeps the number of files you need to manage smaller and still allows you to only need to update one file when a service pack is added.

The last thing to consider is installation location. If you have offices in other locations with their own server, you can install the same software at different locations, and have your VBScript/batch files use a server local to computer being installed to. One thing I did in all my master VBScript/batch files was to calls one of several other special VBScript/batch file that mapped a drive letter to a server. The master VBScript/batch file, then called the appropriate script to map the drive properly depending where the computer was located. Here's an example…



Using the techniques, you can automate the install and uninstall of all of your various Autodesk products along with service packs and other items. One thing to watch for with VBScript or a batch file, is spaces in file names and paths. They require double quotes surrounding them. And when mixed with other parameters that need double quotes, you will often have nested double quotes, especially with VBScript. It is best to use paths/names with no spaces and techniques that minimize use of quotes where possible. It'll save you a lot of troubleshooting time.