

# Advanced Edge Client Installation for Windows&ndash;The Mysteries of Windows Installer Revealed



El\_Jefe, 2012-02-08

---

In many small to medium sized companies (and even some fairly large ones), when you need a new piece of software installed on your desktop, you call the IT guy, or put in a trouble ticket, and the IT guy shows up with a DVD or CD, and installs your software. If you're lucky, he remote controls your PC, and uses a shared network drive to install software to your PC. This works fine for an office of 10 – 50 people. But let's say you're a large multinational corporation with 50,000 or 100,000 desktops. To roll out software to that number of desktops and laptops becomes a very costly investment if you are doing it manually. You'd need to clone Billy the desktop tech a whole lot of times to get enterprise-wide rollouts done in a timely fashion.

Microsoft thought this too, and back in 1999-2000, they partnered with a company called Veritas who had a product called WinInstall. They came up with a standard for deploying software that, with some revisions, is still in use today. They called it the Microsoft Installer, later it became known as Windows Installer. We call them MSI files generally, from their extension, .msi. You can think of the file as a flat-file database, very much like an MS Access file. There are a number of tables in it that contain the actual compressed files that are deployed to your PC, scripts that are used in the run time installation, and properties that are presented to the user in a GUI install. In addition to all this goodness, the MSI files actually do health checks on the application by using key files and components (think files, registry settings, COM objects, stuff like that), and if the installation becomes damaged somehow either by the user or some other process overwriting a key piece, upon launch of the advertised shortcut, it will actually self-heal the application. Conversely, if you need to remove a particular piece of software from the environment, Windows Installer allows for the orderly clean up and removal of applications using the same database file.

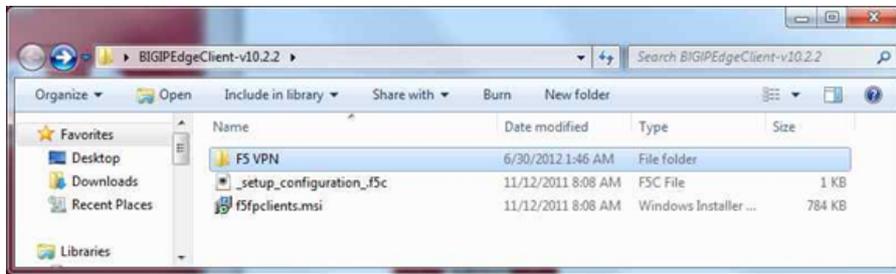
The neat thing is that if you have a little bit of know-how as to how it all works together, you can roll out the same software application to 50,000 PCs, and each install will have the same characteristics as all the others, because you set the parameters for the install. If you want to you can install the applications in the background, right under the user's noses, and all the while, they're none the wiser, you can do that to with some command line switches. It's one of those jobs in IT where if no one knows anything is going on, you're doing your job right. Or to steal a line from The Wizard of Oz, "Pay no attention to the man behind the curtain."

So, how does this affect me? Well, most of us reading this are network folks, and don't have a lot to do with enterprise desktop management. However, you may have a large APM Install base at your company, and you need to know how to roll out the Edge Client to all your user's laptops, via Active Directory, SCCM, or some other enterprise desktop management platform. Or, maybe the enterprise desktop folks need a little guidance on how to roll out the Edge Client. Hopefully, this article will clear all that up. So, let's start by looking at the BIG-IP Edge Client.

When you download the BIG-IP Edge Client to your desktop, you notice something right away. The icon for the BIGIPEdgeClient.exe doesn't look like a normal application (.exe) file icon. It looks like this –



What that tells me is that this is no ordinary executable, but one that is a wrapper to call a Windows Installer File, that is very commonly used in applications that are put together using InstallShield. If you use a tool like 7-Zip or WinRAR, you can extract the contents of this file, into a folder. Here, we can see the contents –

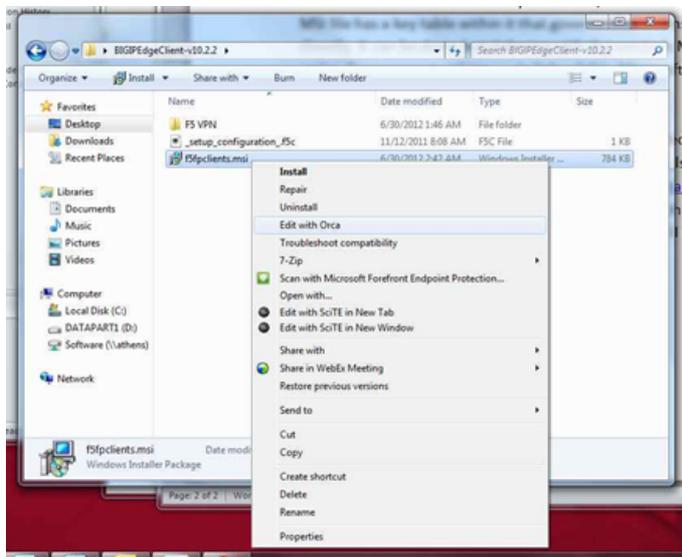


All of these components are necessary for the installation of the Edge client. If you are doing a large scale rollout of BIG-IP Edge Client, this is what you use as the source directory for your install. In the case of the Edge Client, the MSI file is a database, and instead of using the MSI file to store the file components, the MSI file uses the F5 VPN folder to place the contents of it on the end user's PC.

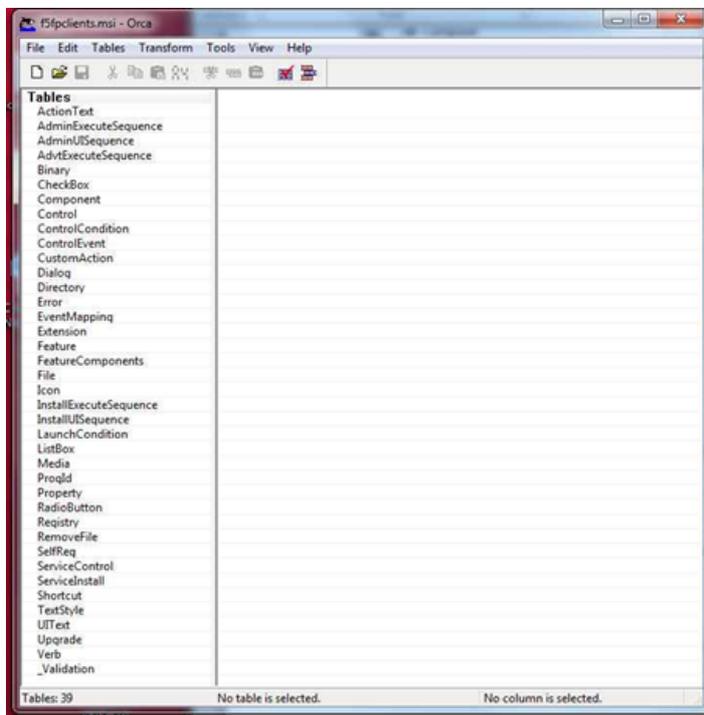
In order to customize and modify the install, we need to learn a few things. First, understand that the MSI file has a very important table within it that governs how it is installed. However, you don't want to edit it directly. It can be done, but tinkering with a vendor's MSI file can make it behave erratically, and makes for a support nightmare. In light of this, Microsoft has provided us a couple of ways to deal with this, and we are going to look at both.

The table in the MSI file that governs how it is installed is the Property table. Microsoft has provided us a tool called Orca to not only examine the MSI, but also to create transforms. You can download Orca here - <http://www.microsoft.com/en-us/download/details.aspx?id=3138> Orca is a very, very small part of the SDK, and can be found here - C:\Program Files\Microsoft SDKs\Windows\v7.0\Bin\Orca.msi, after the SDK is installed. Personally, I saved the orca.msi elsewhere, and uninstalled the rest of the SDK, because Orca is really all I needed. I then installed the Orca Windows Installer file by double clicking on it, and I have the tool I need.

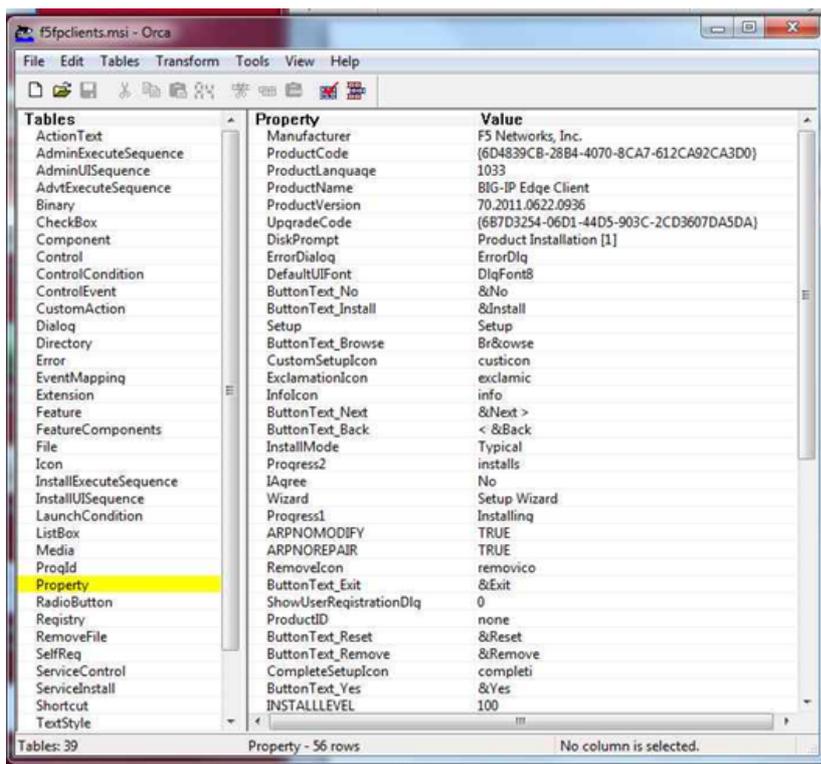
Then, when I browse back to the expanded archive, if I right click on the f5fpclients.msi file, I have a new option – Edit with Orca, as shown below -



Choose Edit with Orca, and you can then explore inner workings of the MSI.



This is a listing of all the tables within the MSI database file. In here, you can add, delete, modify anything in the MSI that you would need. You can store new custom procedures to kick off other processes, register the software, just about anything you want. Choose the Property table.



This is where all the neat stuff is kept, and where you can change the way the application installs. Notice some properties are written in all caps, and others in upper and lower case. The ones in all caps are public properties. The public properties in the Edge Client MSI are –

- **ARPNOMODIFY** – default is TRUE - the user has the option to modify the program from the Add/ Remove Programs Interface
- **ARPNOREPAIR** – default is TRUE - the user has the option to repair the program from the Add/ Remove Programs Interface
- **INSTALLLEVEL** – default is 100 - the initial level at which features are selected "ON" for installation by default.
- **ARPHHELPINK** – default is <http://askf5.com> - Add/ Remove Programs Interface displays this as a help link for the program

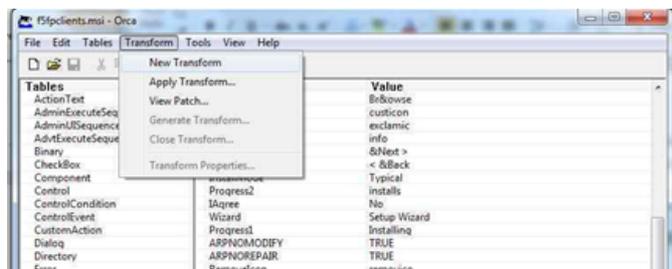
- **PROMPTROLLBACKCOST** – default is P - the action the installer takes if rollback installation capabilities are enabled and there is insufficient disk space to complete the installation. P = prompt the user with a dialog asking to rollback or not.
- **ALLUSERS** – default is 1 - configures the installation context of the package. 1 is a per-machine install, 0 is a per user install. If the user lacks rights to do a per-machine install, if this property is set to 1, it will fall back to a per user install.
- **ARPPRODUCTICON** – default is icon.ico - Specifies the icon used in the Add/ Remove Programs Interface.
- **STARTAPPWITHWINDOWS** – default is 0 - The Edge Client has to be started manually by the user. If set to 1, Edge client loads with windows.

One interesting thing to note about public properties is that they can be set at package load through the command line interface of the msixec installer. This is useful if you only want to change public properties, you can set your command line execution of your enterprise desktop management system to set these properties while installing the F5 supplied MSI. There are some examples on how to do this at the end of the article.

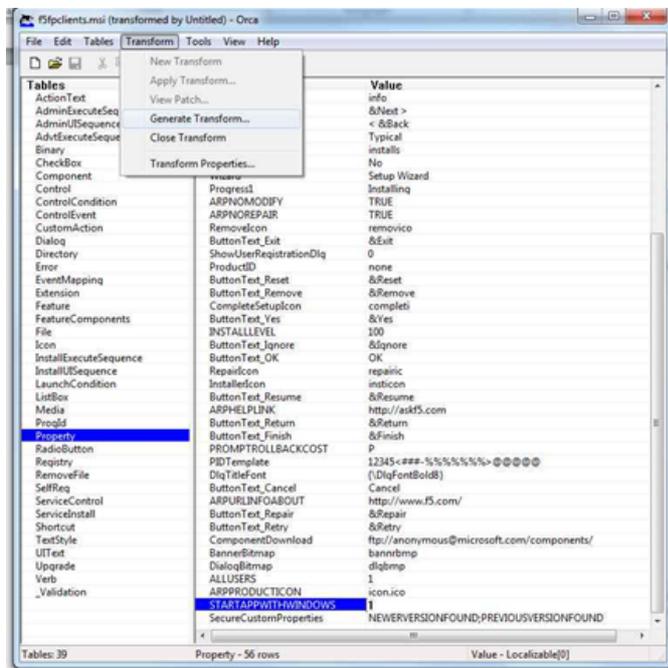
The properties with upper and lower case letters are private properties. These are properties that are protected and therefore cannot be set at the command line at runtime. Some of the more interesting ones in the Edge client MSI are –

- **ProductCode** – this is a GUID used to identify this particular MSI. This is very helpful when uninstalling a program from your PC's. running `msiexec.exe /x {GUID}` will uninstall the application from the desktops. For example, if you had a bunch of older Edge Clients you wanted off the PC's before you rolled out the new version, you could get the product code from the old MSI, and run `msiexec` with the `/x` option to do the cleanup.
- **Iagree** – this is the checkbox that you click to accept the End User License Agreement (EULA), No is unchecked, Yes is checked
- **InstallMode** – Tells the installer how to install the application. Default is Typical, which includes all the usual options. Some other choices are Custom or Complete.

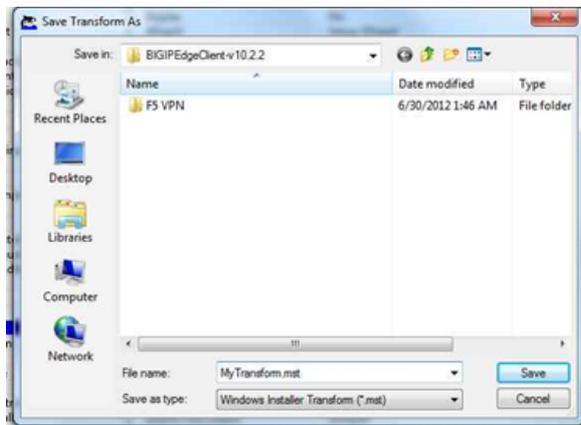
Since these are private properties, the only means by which we modify them is by a transform file that is saved and included in the source directory, and called when installing the MSI. Also, you can modify public properties via a transform, just be aware that the public properties set at the command line generally take precedence. Let's say we want to modify our Edge Client MSI to start with Windows, and we want to make this happen via a transform. Since we have Orca open, click the Transform tab, and choose New Transform.



Then, make the changes you need. In our case, we go down to the `STARTAPPWITHWINDOWS` property, and change that value to 1. Then go back to the Transform tab, and choose Generate Transform.



We can then save our new transform file to be used later. In our case, we save it as MyTransform.mst



Congratulations. You have just created your first transform. We can then call this from the command line using the following example –

- **msiexec.exe /i f5fpclients.msi TRANSFORMS=MyTransform.mst /qn!**

This installs the f5fpclients MSI file with the MyTransform.mst transform quietly, with no option to cancel. To set the STARTAPPWITHWINDOWS public property in the command line, we could use this example –

- **msiexec.exe /i f5fpclients.msi STARTAPPWITHWINDOWS="1" /qn!**

This installs the f5fpclients MSI file with the STARTAPPWITHWINDOWS property set to 1, quietly with no option to cancel.

Here are some of the more common MSIEXEC command line options you might find useful –

- **Install Options**
  - /i - normal installation
  - /a - administrative install
  - /x - uninstall the package
- **User Display Options**
  - /quiet - quiet mode (there is no user interaction)
  - /passive - unattended mode (the installation shows only a progress bar)
  - /q - set the UI level:
    - n - no UI
    - b - basic UI
    - r - reduced UI

- f - full UI

- **Restart Options**

- /norestart - the machine will not be restarted after the installation is complete
- /promptrestart - the user will be prompted if a reboot is required
- /forcerestart - the machine will be restarted after the installation is complete

If you want to learn more about MSI files and how to modify them, here are a few links to get you started –

- MSIEXEC command line - <http://technet.microsoft.com/en-us/library/cc759262%28WS.10%29.aspx>
- ItNinja (formerly AppDeploy) - <http://www.itninja.com/> Large blogsite, lots of MSI resources
- InstallSite – <http://www.installsite.org> Discussion Community and Windows Installer Info

---

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](http://f5.com)

F5 Networks, Inc.  
Corporate Headquarters  
[info@f5.com](mailto:info@f5.com)

F5 Networks  
Asia-Pacific  
[apacinfo@f5.com](mailto:apacinfo@f5.com)

F5 Networks Ltd.  
Europe/Middle-East/Africa  
[emeainfo@f5.com](mailto:emeainfo@f5.com)

F5 Networks  
Japan K.K.  
[f5j-info@f5.com](mailto:f5j-info@f5.com)