

Customize Client Center Installation

There are different ways to install Client Center for Configuration Manager
(<https://sccmclctr.codeplex.com/>):

- Click Once (<http://sccmclctr.codeplex.com/releases/clickonce/SCCMCliCtrWPF.application?localhost>)
- Browser based Version (IE is used to display the UI, the Tool is not running on the Web-Server)
- Windows Installer Installation
- chocolatey.org (cinst SCCMCliCtr)
- ...

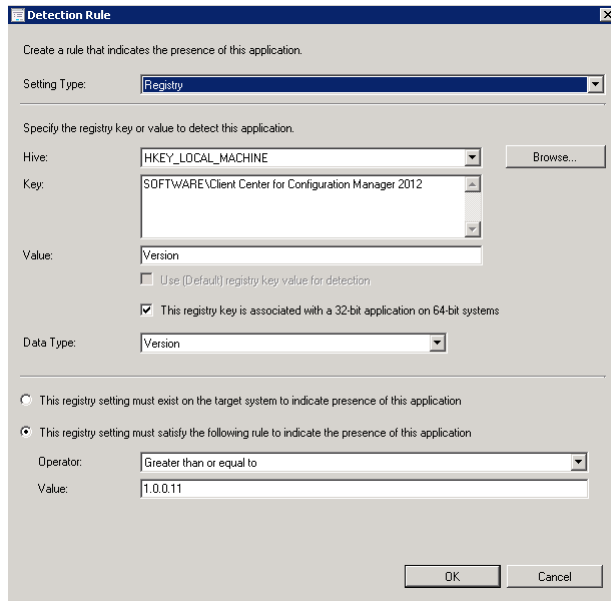
In a corporate environment, it's mostly difficult to access resource on the Internet, so ClickOnce and the chocolatey installer may fail.. The Browser Version is often limited because of the dependency to IE (Security, UAC, RunAs Administrator and x86 Problematic). So for most enterprise environments, the Windows Installer Package may be the only proper way to distribute Client Center.

1 Contents

2	CM12 MSI Deployment Type.....	2
3	Create a Transform File.....	2
4	Place Configuration files outside of the MSI	3
5	SCCMCliCtrWPF.exe.config.....	4
6	Hide the License start screen	5
7	Disable Plugins (Donators only)	5
8	Integrate License key (Donators only)	5
9	Change source path for Self-Update (Donators only)	6

2 CM12 MSI Deployment Type

Creating a Configuration Manager Application for the Tool shouldn't be hard. By default, CM12 is using the MSI Product Code as detection method for the Application. Product Codes can or will change ([http://msdn.microsoft.com/en-us/library/aa370854\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa370854(v=vs.85).aspx)) so it can be dangerous to use it as detection method. For Client Center, a Registry-Key should be used to detect the Product and Version:

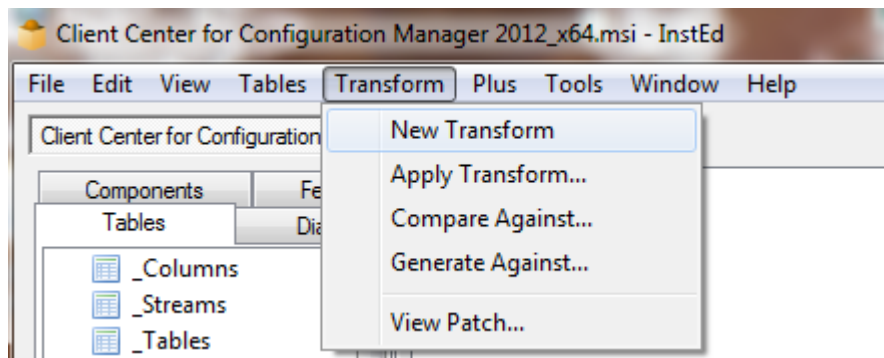


(It's always an 32Bit key; HKEY_LOCAL_MACHINE\SOFTWARE\Client Center for Configuration Manager 2012)

3 Create a Transform File

Some Enterprise related Settings will require a Transform-File to "transform" some settings in the Windows installer package. By using a Transform-File, the original MSI is not modified and can (mostly) be replaced with newer releases of the File.

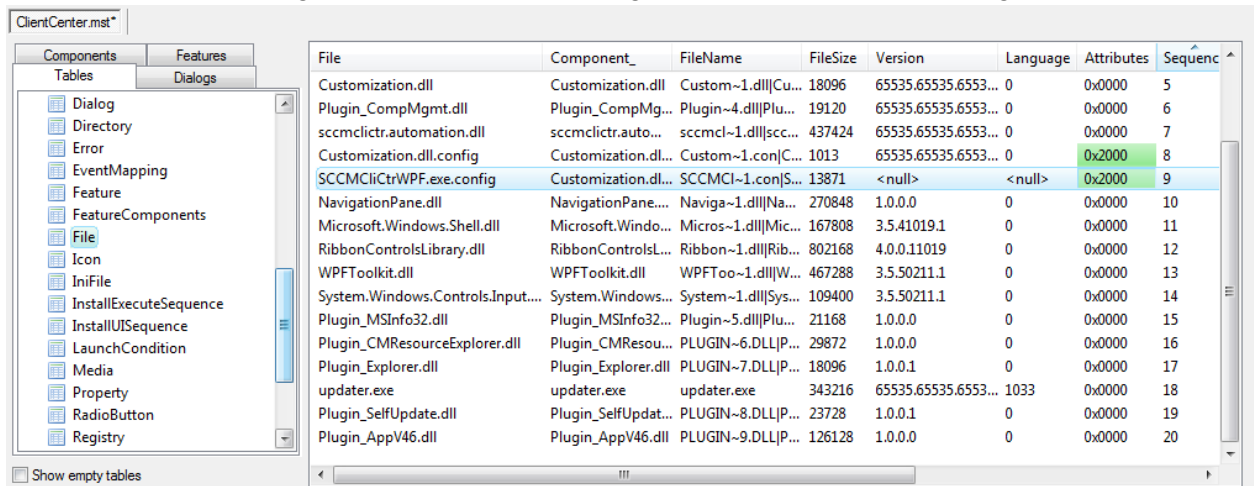
1. Open "Client Center for Configuration Manager 2012_x64.msi" with InstEd (<http://www.instdedit.com/>)
2. Create a new Transform (e.g. ClientCenter.mst)



4 Place Configuration files outside of the MSI

Placing the .config files outside of the MSI allows easy modification of these files.

3. In the Files table, change the Attributes to 0x2000 (NonCompressed) on all Files you want to manipulate (e.g. sccmclctrwpf.exe.config and customization.dll.config)



File	Component_	FileName	FileSize	Version	Language	Attributes	Sequence
Customization.dll	Customization.dll	Custom~1.dll Cu...	18096	65535.65535.6553...	0	0x0000	5
Plugin_CompMgmt.dll	Plugin_CompMg...	Plugin~4.dll Plu...	19120	65535.65535.6553...	0	0x0000	6
sccmclctr.automation.dll	sccmclctr.auto...	sccmcl~1.dll scc...	437424	65535.65535.6553...	0	0x0000	7
Customization.dll.config	Customization.dl...	Custom~1.con C...	1013	65535.65535.6553...	0	0x2000	8
SCCMClctrWPF.exe.config	Customization.dl...	SCCMCl~1.con S...	13871	<null>	<null>	0x2000	9
NavigationPane.dll	NavigationPane...	Naviga~1.dll Na...	270848	1.0.0.0	0	0x0000	10
Microsoft.Windows.Shell.dll	Microsoft.Windo...	Micros~1.dll Mic...	167808	3.5.41019.1	0	0x0000	11
RibbonControlsLibrary.dll	RibbonControlsL...	Ribbon~1.dll Rib...	802168	4.0.0.11019	0	0x0000	12
WPFToolkit.dll	WPFToolkit.dll	WPFToo~1.dll W...	467288	3.5.50211.1	0	0x0000	13
System.Windows.Controls.Input....	System.Windows...	System~1.dll Sys...	109400	3.5.50211.1	0	0x0000	14
Plugin_MSInfo32.dll	Plugin_MSInfo32...	Plugin~5.dll Plu...	21168	1.0.0.0	0	0x0000	15
Plugin_CMResourceExplorer.dll	Plugin_CMResou...	PLUGIN~6.DLL P...	29872	1.0.0.0	0	0x0000	16
Plugin_Explorer.dll	Plugin_Explorer.dll	PLUGIN~7.DLL P...	18096	1.0.0.1	0	0x0000	17
updater.exe	updater.exe	updater.exe	343216	65535.65535.6553...	1033	0x0000	18
Plugin_SelfUpdate.dll	Plugin_SelfUpdat...	PLUGIN~8.DLL P...	23728	1.0.0.1	0	0x0000	19
Plugin_AppV46.dll	Plugin_AppV46.dll	PLUGIN~9.DLL P...	126128	1.0.0.0	0	0x0000	20

4. Change also the Sequence of the Files to a number that is higher than the highest original value (in the case, the highest number is 20 -> the Files will get at least 21 and 22 or use higher numbers for the case that new files will be added in the future)

WPFToolkit.dll	WPFToolkit.dll	WPFToo~1.dll W...	467288	3.5.50211.1	0	0x0000	13
System.Windows.Controls.Input....	System.Windows...	System~1.dll Sys...	109400	3.5.50211.1	0	0x0000	14
Plugin_MSInfo32.dll	Plugin_MSInfo32...	Plugin~5.dll Plu...	21168	1.0.0.0	0	0x0000	15
Plugin_CMResourceExplorer.dll	Plugin_CMResou...	PLUGIN~6.DLL P...	29872	1.0.0.0	0	0x0000	16
Plugin_Explorer.dll	Plugin_Explorer.dll	PLUGIN~7.DLL P...	18096	1.0.0.1	0	0x0000	17
updater.exe	updater.exe	updater.exe	343216	65535.65535.6553...	1033	0x0000	18
Plugin_SelfUpdate.dll	Plugin_SelfUpdat...	PLUGIN~8.DLL P...	23728	1.0.0.1	0	0x0000	19
Plugin_AppV46.dll	Plugin_AppV46.dll	PLUGIN~9.DLL P...	126128	1.0.0.0	0	0x0000	20
Customization.dll.config	Customization.dl...	Custom~1.con C...	1013	65535.65535.6553...	0	0x2000	21
SCCMClctrWPF.exe.config	Customization.dl...	SCCMCl~1.con S...	13871	<null>	<null>	0x2000	22

5. In the Media tab, create a new entry where the LastSequence id is the highest id of the custom files

Field Name	Value
DiskId: Number (2), Key, Required	2
LastSequence: Number (4), Required	22
DiskPrompt: String (64), Localizable	<null>
Cabinet: String (255)	<null>
VolumeLabel: String (32)	LABEL
Source: String (32)	<null>

Result should look like:

DiskId	LastSequence	DiskPrompt	Cabinet	VolumeLabel	Source
1	20	<null>	#disk1.cab	<null>	<null>
2	22	<null>	<null>	LABEL	<null>

- Copy the Files from an installed instance of the Tool into the directory where the MST and MSI exists.

Name	Date modified	Type
Client Center for Configuration Manager ...	15.05.2014 15:19	Windows Installer ...
ClientCenter.mst	20.05.2014 10:34	MST File
Customization.dll.config	19.05.2014 13:44	XML Configuratio...
SCCMCliCtrWPF.exe.config	20.05.2014 09:55	XML Configuratio...

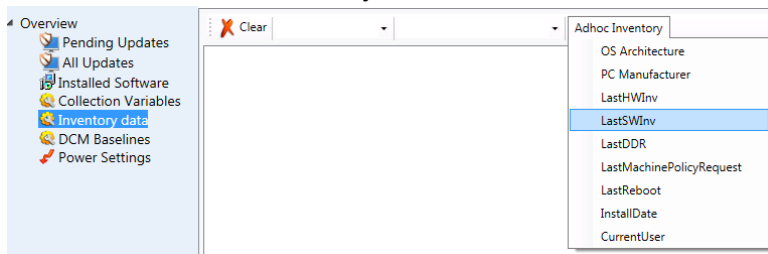
You can now modify these Files based on your requirements.

5 SCCMCliCtrWPF.exe.config

There are some sections in the config file that should be reviewed before the tool is distributed:

- ☐ AgentInstallIPS : The PowerShell Code to Install the CM12 Agent
- ☐ AgentInstallSiteCode : Default Site-Code for Agent Installation
- ☐ AgentInstallIMP : Default Management Point for Agent Installation
- ☐ NoLocalAdminCheck : to prevent checking if the user is local (the case with UAC).

☐ AdhocInv: Custom Inventory classes



- ☐ WinRMPort : Default = 5985 (non SSL)
- ☐ WinRMSSL : Enable SSL for WinRM connections , you have to set WinRMPort to 5986 (<http://support.microsoft.com/kb/2019527>)

6 Hide the License start screen

Donators over 10\$ will get a LicenseKey to hide the start screen and to activate some more goodies. But It's not a license contract, it's just a gift...



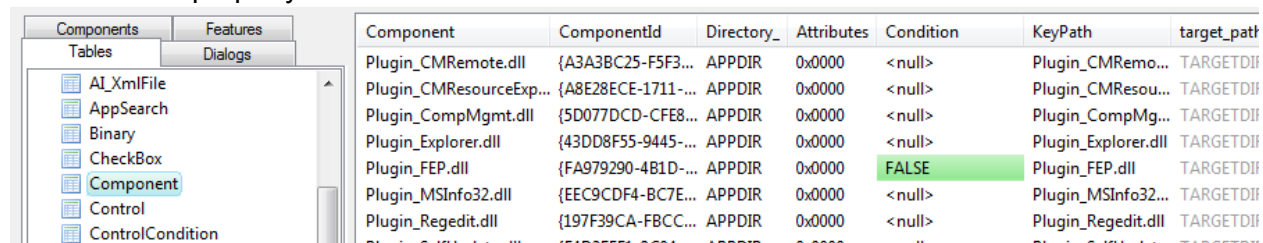
https://www.paypal.com/cgi-bin/webscr?cmd=_s-xclick&hosted_button_id=TLTFJHYA69VHU

The name of the Donator will be visible in the Title of the Tool. If you want to see your company name, the donation should be more generous.

7 Disable Plugins (Donators only)

The Tools-Items are loaded on startup from all Plugin_*.dll Files. If you don't want to have some specific plugins, make sure that the corresponding Plugin_*.dll files is not installed.

In the Component tab, select the Plugin's to disable and insert a "FALSE" in the Condition property.

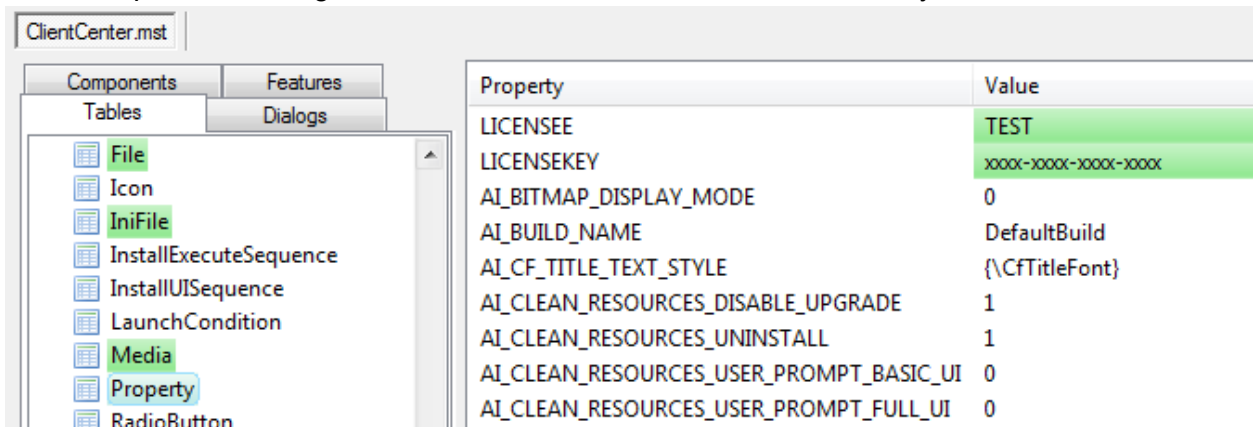


*) in this example, the FEP plugin will not install

8 Integrate License key (Donators only)

1. The License key for the donators can be added as MSI parameters like: `msiexec /i clictr.msi LICENSEE="test" LICENSEKEY="xxxx-xxxx-xxxx-xxxx"`.

2. Another option is to integrate the LICENSEE and LICENSEKEY directly into the Transform



3. Or as 3rd way, modify Customization.dll.config

```
<applicationSettings>
  <SCCMcliCtr.Properties.Settings>
    <setting name="Licensee" serializeAs="String">
      <value>test</value>
    </setting>
    <setting name="LicenseKey" serializeAs="String">
      <value>xxxx-xxxx-xxxx-xxxx</value>
    </setting>
  </SCCMcliCtr.Properties.Settings>
</applicationSettings>
```

Note: If you enter the Key as Parameter or over Licensee/LicenseKey property, the key will be stored in the registry and will not remove during uninstall or update. The File Customization.dll.config can be overwritten from newer versions.

9 Change source path for Self-Update (Donators only)

The latest Versions of client center can do a self-update. If you don't want to update over the Internet, you can change the source path to point to an internal Web-Server, or disable the Plugin (See Section "Hide the License start screen").

Note:

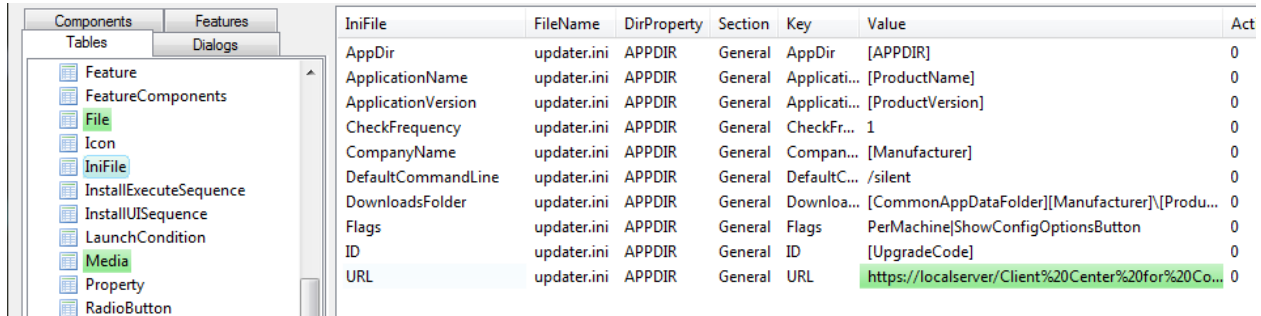
Self-Update cannot handle external files like Transform-Files, you have to integrate the changes into a new Windows Installer File.

I would recommend disabling the Self Update Plugin in enterprise environments and distribute updates with Configuration Manager.

1. Download the update definition text-file from https://sccmclictr.azurewebsites.net/Client%20Center%20for%20Configuration%20Manager_Updates.txt to an internal Webserver.

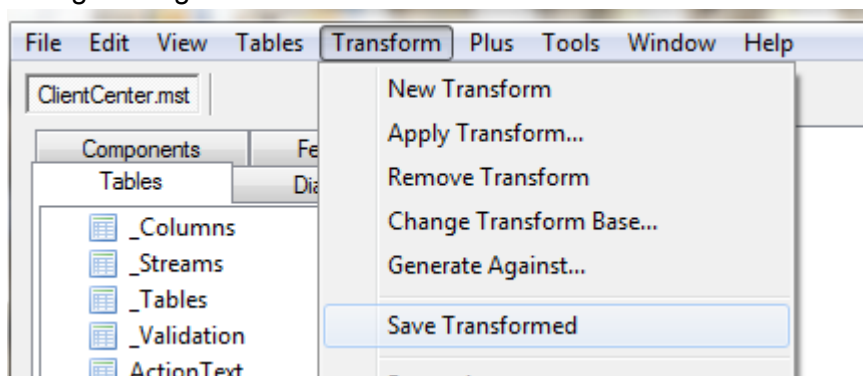
Thursday, June 12, 2014

2. In the IniFile Section change the URL of updater.ini to the update definition text file on your internal infrastructure



IniFile	FileName	DirProperty	Section	Key	Value	Act
AppDir	updater.ini	APPDIR	General	AppDir	[APPDIR]	0
ApplicationName	updater.ini	APPDIR	General	Applicati...	[ProductName]	0
ApplicationVersion	updater.ini	APPDIR	General	Applicati...	[ProductVersion]	0
CheckFrequency	updater.ini	APPDIR	General	CheckFr...	1	0
CompanyName	updater.ini	APPDIR	General	Compan...	[Manufacturer]	0
DefaultCommandLine	updater.ini	APPDIR	General	DefaultC...	/silent	0
DownloadsFolder	updater.ini	APPDIR	General	Downloa...	[CommonAppDataFolder][Manufacturer][Produ...	0
Flags	updater.ini	APPDIR	General	Flags	PerMachine>ShowConfigOptionsButton	0
ID	updater.ini	APPDIR	General	ID	[UpgradeCode]	0
URL	updater.ini	APPDIR	General	URL	https://localhost/Client%20Center%20for%20Co...	0

3. Self-Update cannot use a Transform File, so you have to generate a new MSI with all the changes integrated.



4. Calculate the MD5 checksum of the new MSI with PowerShell:

```
$newMSIFilePath = "Client Center for Configuration Manager 2012_x64_new.msi"
$md5 = new-object -TypeName System.Security.Cryptography.MD5CryptoServiceProvider
$hash = [System.BitConverter]::ToString($md5.ComputeHash([System.IO.File]::ReadAllBytes($newMSIFilePath)))
$hash.replace('-', '').tolower()
```

Within the update definition Text-File, change the URL of the msi File to the internal location that hosts the updated msi and the new md5 checksum.

```
;aiu;
[Update]
Name=Client Center for Configuration Manager 2012
Category=minor Update
ProductVersion=1.0.0.11
URL=https://localhost/Client Center for Configuration Manager 2012_x64_new.msi
Size=2659328
MD5=db67d3448776fae5dd29115b50c5bf38
ServerFileName=Client Center for Configuration Manager 2012_x64_new.msi
Flags=Critical|NoCache|Os64
RegistryKey=HKLM\SOFTWARE\Client Center for Configuration Manager 2012\Version
Version=1.0.0.11
UpdatedApplications=Client Center for Configuration Manager 2012[1.0-1.0.0.11)
Feature=new App-V 4.x Plugin
AutoCloseApplication=[APPDIR]SCCMcliCtrWPF.exe;Client Center for CM12
```

The yellow marked Version strings must match with the Version of the Product.