# SCCM AddApp Tool

Version: 2.10.1

Written by: Joel Chettle

Last modified: 9/29/2021

**Contents**

[SCCM AddApp Tool 1](#__RefHeading___Toc1809_651615580)

[Description 1](#__RefHeading___Toc1811_651615580)

[Requirements 2](#__RefHeading___Toc1813_651615580)

[Configuration 2](#__RefHeading___Toc1815_651615580)

[How-to Launch 5](#__RefHeading___Toc1817_651615580)

[How-to Use 6](#__RefHeading___Toc1819_651615580)

[Example 10](#__RefHeading___Toc2726_651615580)

[Troubleshooting 11](#__RefHeading___Toc1821_651615580)

[Credits 12](#__RefHeading___Toc1823_651615580)

## Description

If consistent packaging standards and naming conventions are in place, this tool allows the packager to automatically create the Active Directory group, Collection, Application, and Deployment in SCCM for a package by providing minimal input.

For instance, if using a package with the name of “Joes\_MostGreatSoftware\_1.0” and all options were selected, the following would be created by the tool:

* **AD Group** – Joes MostGreatSoftware 1.0
* **SCCM Collection** – Joes\_MostGreatSoftware\_1.0-Install
  + linked to AD Group
* **SCCM Application** – Joes\_MostGreatSoftware\_1.0
* **SCCM Deployment** – Joes\_MostGreatSoftware\_1.0
  + linked to Collection

In many cases this automates ALL, if not most of, the steps required.

## Requirements

* Microsoft .Net Framework 2.0+
* Microsoft PowerShell 3.0 (minimum); 4.0+ (recommended)
* Active Directory and SCCM modules, which are included automatically if both AdminTools and Configuration Manager are installed. The SCCM cmdlet library may get updated by Microsoft, in which commands within the script may become depreciated or could no longer function. Version tested, at the time, is 5.1710.1059.1003.

**Files included:**

* SCCMAddAppTool.exe
* SCCMAddAppSettings.xml

## Configuration

All custom settings are stored in the settings XML. This is so the core script does not have to be modified by the user.

Prior to launching the tool, make sure these settings are defined for the site.

|  |  |
| --- | --- |
| **Tool Settings** | |
| SelectAll | Create all components (Active Directory group, Collection, Application, and Deployment). True/false. |
| CreateAD | Create an Active Directory group for the package. Ignored if SelectAll is true. True/false. |
| CreateCollection | Create a device Collection. If AD group is enabled, this will also create a query rule linking the two. Ignored if SelectAll is true. True/false. |
| CreateDeployment | Create a Deployment to the Collection. Collection must be enabled. Ignored if SelectAll is true. True/false. |
| *\*The above 4 settings can be changed within the tool itself. But it is provided within the XML as an option to always start the tool with a custom selection. For instance if a site did not utilize AD for software distribution, this could be set to false and AD would always start off as disabled when launching the tool rather than always having to manually deselect it.* | |
| **AD Settings** | |
| ADDomain | Active Directory domain where the Distribution groups will be created |
| ADPath | The full path to the OU where the groups will be created. In form-  OU=,OU=,DC=,DC=,DC= |
| ADGroupScope | The Security type defined for the AD group. Valid values are DomainLocal, Global, and Universal. |
| ADDescription | Text entered in the Description field for the AD group. This can be left blank. |
| **SCCM Settings** | |
| Sitecode | SCCM sitecode |
| LimitingCollection | The Limiting Collection to be used for the Device Collections |
| CollectionFolder | The folder in SCCM under Device Collections that the new Collections will be moved to. Tool will create folder if it doesn’t exist. |
| RefreshInterval | The interval type used to refresh the Collection. Valid values are Minutes, Hours, or Days. |
| RefreshIntCount | The count used in conjunction with the Interval type. Must be numeric. |
| ApplicationFolder | The folder in SCCM that new Applications or Packages will be moved to. Tool will create folder if it doesn’t exist. |
| PkgPrefix | Text to prefix to the Package Name in SCCM. This can be left blank. |
| AllowTaskSeqInstall | Application setting. Allow this application to be installed from the Install Application task sequence action without being deployed. True/false. |
| PrestageDP | Application Distribution setting. Prestaged distribution point. Automatically download content when the packages are assigned to distribution points, Download only content changes to the distribution point, or Manually copy the content in this package to the distribution point. Valid values are AutoDownload, DeltaCopy, or NoDownload. |
| DPGroup | The Distribution Group used to distribute content. Must use a pre-existing group and not an individual DP. |
| AllowBranchCache | Application Deployment Type setting. Allow clients to share content with other clients on the same subnet. True/false. |
| InstallBehavior | Application Deployment Type setting. Valid values are InstallForUser, InstallForSystem, or InstallForSystemIfResourceIsDeviceOtherwiseInstallForUser. |
| maxDeployTime | Application Deployment Type setting. Maximum allowed run time (minutes). Enter numeric value between 15 and 720. |
| SendWakeup | Application Deployment setting. Send wake-up packets. True/false. |
| UserNotification | Application Deployment setting. User notifications. Display in Software Center and show all notifications, Display in Software Center and only show notifications for computer restarts, or Hide in Software Center and all notifications. Valid values are DisplayAll, DisplaySoftwareCenterOnly, or HideAll. |
| PkgrTestCollection | If defined, will create an additional Available deployment to this Collection. This is intended to be used for package testing. |
| **Package settings** | |
| PkgDelimiter | A non-alphanumeric character to separate the Package name into Manufacturer Product and Version. Example “-“ if the package names are Manufacturer-Product-Version. Some characters are not allowed, such as  / \ : | \* ? < > “ . |
| PkgFilesFolder | The source folder that packages are stored. The package must be located in/under this folder. |
| MSIargs | Arguments to pass in to msiexec.exe along with the install command. This is if using a MSI install type. |
| UninstallArgs | Arguments to pass in to msiexec.exe along with the uninstall command. This is if using a MSI install type. |
| LogOption | The log level to use with msiexec. This is if using a MSI install type. Example: l\*v |
| LogPath | The log path to use with msiexec. This is if using a MSI install type. |
| ScriptInstallCMD | The install command to enter in the SCCM Application if using a Script install type. |
| ScriptUninstallCMD | The uninstall command to enter in the SCCM Application if using a Script install type. |
| **Testing** | |
| TestMachines | Tool can populate test machines to add to the AD group if same one(s) are used per packager. Define the user ID as <Tester User=””> followed by the machine name(s) as the value. Multiple can be entered separated by , or ;. |

\*Any True/False settings that do not have a True/False value entered will default to “False”.

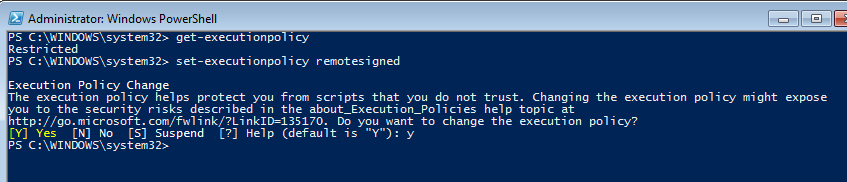
After these are set, they will remain consistent for all packages created for the site.

## How-to Launch

To run the tool, simply double-click the EXE file or right-click the file and select “Run As”. The tool must be run with an admin account that has access to make modifications to the relevant areas of both Active Directory and SCCM.

It may take a moment before the GUI loads. This is normal.

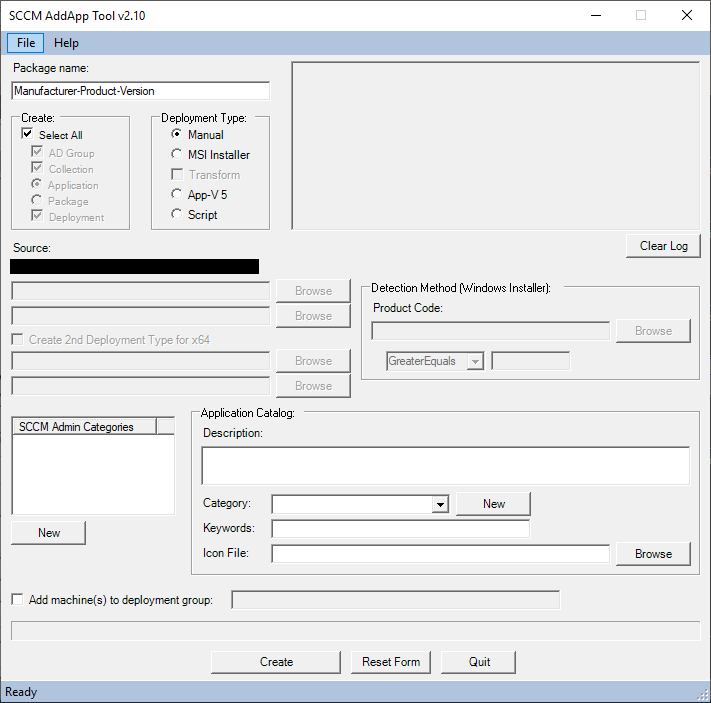
If the script fails to run, first check the Execution Policy on your machine. – Run Powershell.exe and enter “get-executionpolicy”. If it returns Restricted then custom scripts cannot execute. It’s recommended to change this to RemoteSigned by typing “set-executionpolicy remotesigned”. Then hit Y to confirm. (Note, you will not have to do this every time. Setting the policy is permanent.)



Then try launching the tool again.

## How-to Use

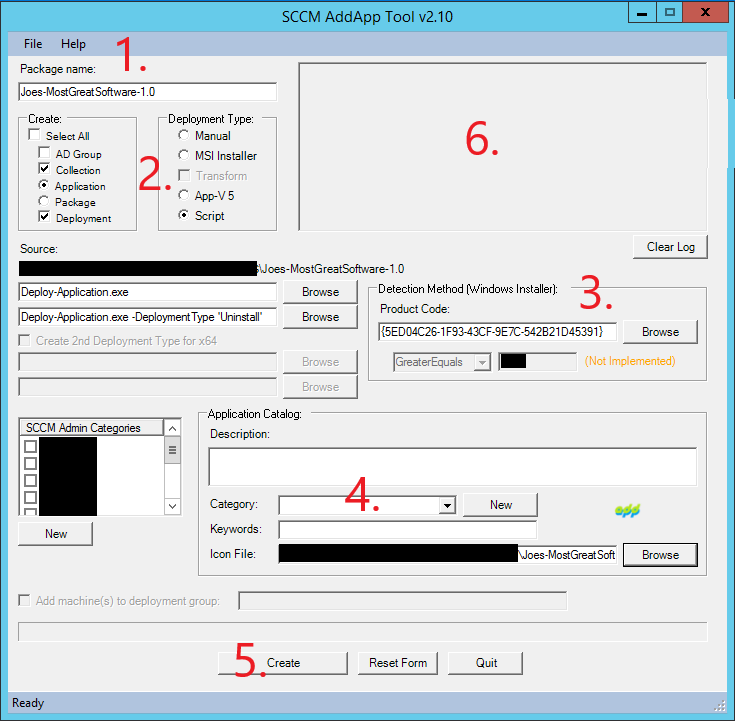
Interface:



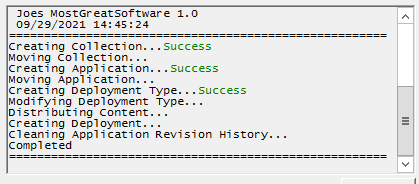
|  |  |
| --- | --- |
|  | Enter the package name in the textbox. Must be in the format shown. The delimiter (In this case “-”) may be different depending on the entry in the settings XML. The max length for this text field is 50 when selecting a Package or 64 when selecting an Application. |
|  | Define which options you would like the tool to create or Select All to create all. When selecting between Application or Package type, note that the tool was built for Applications. But some functionality is there for Packages. (It will create an empty Package with no Programs.) |
|  | Choose a Deployment type option.   * Manual. This is for situations where the tool cannot create the Deployment type by the other means. You will “manually” create it after the tool sets up everything else. * MSI. This is for packages that are simple MSI’s only, with/without a Transform. Install/Uninstall commands, etc. are built automatically. * App-V 5. This is for App-V5 packages. * Script. This creates a script-based installer in which you can define the install/uninstall commands. |
|  | The Source folder will be pulled from the Settings XML + the Package name. The source files must be located in or under this location.  When selecting MSI type, the top box will be for the MSI. The 2nd box for the MST.  When selecting AppV5, the top box will be for the AppV.  When selecting Script, the top box will be for the Install command. The 2nd box will be for the Uninstall command. This will prepopulate with settings from the XML file.  Browse allows you to navigate to a file. |
|  | If the checkbox for a 2nd Deployment Type is selected for x64, this will allow selection of two sets of MSI/MST’s. Note that the path is changed slightly to include a “x86” and “x64” folder. But Browse is available to change it.  NOTE: This will create two MSI Deployment Types in SCCM. But it will not set the Operating System Requirements. That will have to be a manual setting afterward. |
| *\*This may confuse some people. But I used to work at a place with a mixture of 32 and 64-bit systems. Many times we created a package that contained both versions of installers for a single application. So this created a single App with 2 Deployment Types. I have not used this in quite some time. But decided to leave the feature available in the tool.* | |
|  | Detection method will be enabled if selecting a Script type. Will only accept a product code for detection at this time.  Browse allows you to navigate to a MSI and import the code.  Version information is shown, but not currently incorporated in the detection.  Must be in the form of {00000000-0000-0000-0000-000000000000} using only Hexadecimal numbers. |
|  | This displays current Admin categories pulled from SCCM. Any or multiple can be selected for the App.  New button dynamically creates a new category in SCCM and will display it at the bottom of the list. |
|  | |
| Fields used for Application Catalog settings.  Description and Keywords are text fields.  Category is a dropdown list. It also pulls current categories defined in SCCM, and like the admin category, allows you to create New. Only one can be selected though.  Icon file is the file to use for the application icon. This can be a .ico or .exe. It cannot use .dll’s. Browse allows you to navigate to the file. Once selected, you will see the image of the icon appear on screen. When selecting an exe, it extracts the image into a png format to %temp%. After the application is created, this file is removed during cleanup. | |
|  | |
| If you wish to add machine names to the newly created AD group, check the box and enter them in the text field.  If Testers are defined in the settings XML, this will prepopulate with those entries.  This is disabled unless AD Group is checked. | |
|  | Create button starts the process after all selections have been made. Once complete, the form will reset.  Reset Form resets all the fields and selections to the form’s original state.  Quit closes the form. |
|  | This box is read-only and will display messages as progress completes. Certain actions will perform validation checks in which will display Pass or Fail messages. Others will just continue on.  Clear Log button will reset this display to its original state. |

## Example

This may look complicated. But an Application can be created with as little as 3 button clicks. Here’s an example with only 5…

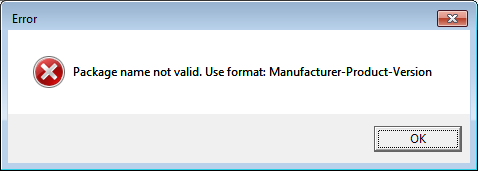


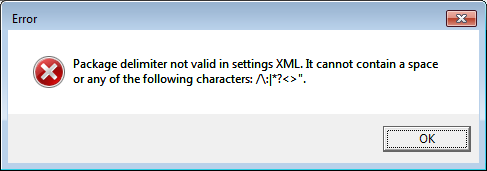
1. Enter package name.
2. Select Script.
   1. In this example we are using the PowerShell Application Deployment Toolkit (which is also another great free tool) for our package. Install and Uninstall commands are set automatically, imported from the XML.
3. Within the package we have an MSI. Browse to this to import the Product Code for Detection.
   1. *I know we could have simply chosen MSI for the Deployment Type. But I wanted to show a better example, as most cases with packaging, it’s not always simple.*
4. Here we chose to only import an icon.
5. Create.
6. Output is logged as the tool runs.



## Troubleshooting

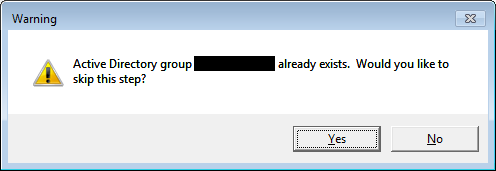
Additional validation checks are in place to check user input. If any fail, an error message will appear on the screen. Such as.





If a step errs during creation of the App, that error should also be displayed in a message window with the full error details.

Some pre-validation checks are also performed at various steps, and if that feature already exists there will be a prompt to Skip or exit. This allows the packager to continue using the tool even if some steps were already completed. Or cancel out if it was unintentional.



## Credits

* PS2EXE-GUI by Markus Scholtes (a rework of PS2EXE by Igor Karstein) – Used to compile the AddApp ps1 script into a more user-friendly executable.