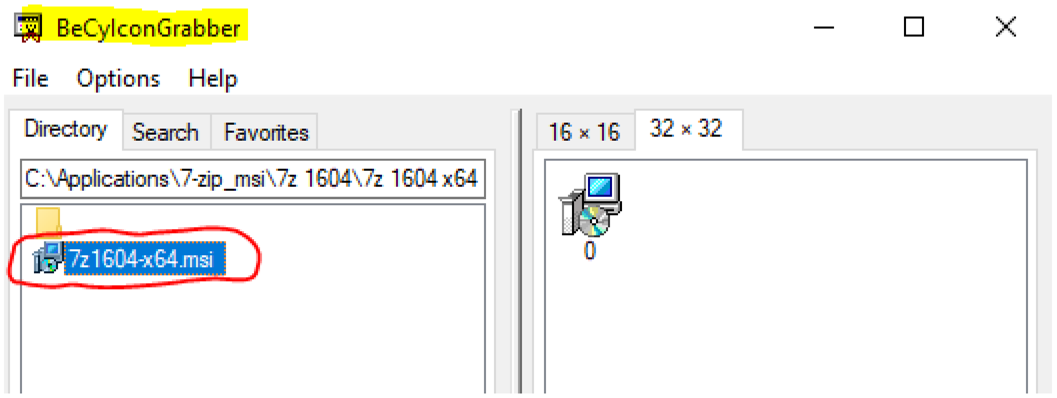
Creating an Application from an MSI

Part 1

In this lecture, we will create an application using an **MSI**. An MSI is a Windows Installer.

- Creating an application from an MSI file is probably one of the easiest applications that you can create within Config Manager because most of the installation command line, including parameters, are pre-filled and are part of the MSI.
- Now let's go ahead and create our application
  From the console, click Software Library, then click Application Management, then click Applications, then right-click create application.
  From the **General screen,** we are going to create our **deployment type** for this application.
- So what is a deployment type? The concept is simple, for example, Lets say we have 100 computers in our office, and we want to deploy the 7-Zip application to all 100 computers. 50 of these computers run Windows 10 64 bit O/S, and the other 50 run Windows 10 the 32 bit version of the O/S. You would create one deployment type for the 64 bit O/S. Then you would go back and create another deployment type for the other 50 computers running Windows 10 32 bit. So in this case, the deployment type is based upon the version of Windows 10.
- In that example, we can create one application, and by creating two deployment types, one for 32 bit windows and one for 64 bit windows, we can successfully deploy that single application to all 100 computers.
  That's the point of deployment types, that an Application can contain multiple deployment types that customize the installation behavior of the application.
- **Now** we want to choose the default MSI file and browse out to our software location. So in this case, that will be \\SASCCM01\Software\7-zip_msi\7z 1604\7z 1604 x64.
- Now click Open, then click next. If you get a warning that saying the installer file could not be verified that means that the file is **not digitally signed. What that means is that 7 zip** is **open source** application. It's pretty common for open source software to not digitally sign their installer files, so I'm just going to click Yes and **ignore** that error. Now click next
- Here is the **General Information** for the software application.
  This is considered the meta data. This is just information about the software, like the Name, Publisher software version. I'll go ahead and quickly complete this form. This information will be stored in the Configuration Manager Database.
- Notice that the MSI installer has pre-filled the install command line with the /I for install and the /q parameter, which is a quiet install.
- For the **Install behavior**, click the down arrow and choose **Install for system**. What this setting does is that the installation will run for all users, even if the user that initiates the installation is not an administrator.

- Now click next , then next again. Now the application creation part has completed. Click close.
- From the console, we see our **application**, right click and click **properties**
- From the **general** screen, we will want to check to allow this application to be installed from a **task sequence.**
- Now click the **Application Catalog**
  We can add categories, documentation links and the privacy URL, which is the license.txt file for the 7-zip program. Categories help to organize your applications in the clients software center. **If you want to create or add a category, just click the edit button.**
  Now let's add an icon to the application. Icons dress up the application in the software center. You can get these icons from the msi or use an Icon Extractor like Be-Cy-IconGrabber. Using this program, you can click the msi file then save the icon.



Now to add the icon click browse and browse out to the icon file, choose the file, click ok, then click apply. Then click ok

Now from the console click the **Deployment types tab** at the bottom

Right click on the deployment type, and click properties, and lets rename the Deployment Type. Considering that we will create another deployment type, I want the name to be a little more descriptive.

- Now we will change the name of the deployment type to **7-Zip 1604 MSI_x64**
- The **content** tab shows the location of our MSI x64 software that we have already defined.
- Now click the **Programs** tab and here we see the pre-filled command line Install and Uninstall command lines taken from the MSI's product code.
- The next tab is the **Detection Method**. The detection method determines if the software has already been installed.

- On the left, click the arrow, then click edit clause. Here we see the **product code**. SCCM uses the product code from the MSI file to detect if the **same version of the software has already been installed**. Click ok.
- Now click **User Experience**, we need to **verify** these settings. **Install for System** and **whether or not a user is logged on**.
- Max allowed runtime could be set to 15 minutes. Because 7-Zip is a very small program. Estimated installation time set to 2 minutes.
- Restart behavior, based upon the return codes of the installation, which could be a Success, or a restart needed, based upon the return code. We will choose the default

- Now Click **Requirements**. What this means is that there could be some requirements that needs to be met in order for this deployment type to be functional
- Considering that this is a 64-bit install, we are going to create a **condition** based upon the **Operating System**. Then for **Operator,** in this case, we will choose **Windows 10 64 bit**.
  So what that means is that this application that we have created would never show up in the clients Software Center for install unless the Operating System of that client was Windows 10 64 bit.
- Click return codes – Here, you can see the default return codes. We won't be adding any return codes.
- Click **Dependencies** – There are no dependancies needed. I will explain dependencies later. Click ok.
- Now back to the console
- Here we have created our 64-bit application installer or deployment type, now we will create our 32-bit installer or deployment type, and we will do that in the next lecture.