

Problem: During capture you get an error: You do not have enough disk space to complete the installation process, or the installation content have been moved, renamed, or deleted. Error: 80070643



- As you can see the error above does not really give you have path to travel down. So, where do we go next?
- I like to look into %temp% and see if application has any logs. This app had good logging. So, I opened the log and found this

#### **Scheduled Task Action (Articulate 360 Installer Service): C:\Program Files**

#### **(x86)\Articulate\360\Desktop Application\Articulate 360 Installer Service.exe**

CustomActionProperties.Get(ScheduledTaskDescription) = Allows non-admin installation or updates of Articulate 360 software. If this task is disabled or stopped, installing or updating Articulate 360 software will require admin privileges. This task uninstalls itself when Articulate 360 is uninstalled.

#### **Exception in RegisterScheduledTaskAction: System.UnauthorizedAccessException: Access is denied. (Exception from HRESULT: 0x80070005 (E\_ACCESSDENIED))**

at Microsoft.Win32.TaskScheduler.V2Interop.ITaskFolder.RegisterTaskDefinition(String Path, ITaskDefinition pDefinition, Int32 flags, Object UserId, Object password, TaskLogonType LogonType, Object sddl)

at Microsoft.Win32.TaskScheduler.TaskFolder.RegisterTaskDefinition(String path, TaskDefinition definition, TaskCreation createType, String userId, String password, TaskLogonType logonType, String sddl)

at Articulate.Bootstrapper.Application.TaskScheduler.RegisterLocalSystemTaskDefinition(String taskName, TaskDefinition taskDefinition)

at Articulate.Bootstrapper.Application.TaskScheduler.RegisterScheduledTask(String taskName, String taskDescription, IEnumerable`1 triggers, IEnumerable`1 actions)

at Articulate.Bootstrapper.Application.CustomActions.RegisterScheduledTaskAction(Session session)

CustomAction RegisterScheduledTaskAction returned actual error code 1603 (note this may not be 100% accurate if translation happened inside sandbox)

- This tells me that FlexApp is interfering with the app creating a **Scheduled Task** and its crashing the install.
- Looking back at the above log, I also saw this -> C:\ProgramData\Package Cache\{D39E6472-95B5-484A-8400-82AC9C55484E}\v1.38.21861.0\Articulate.360.Package.msi

- Another trick is, to start an installer EXE's and find where they unpack all their files for install. Once you locate them, you might be able to break up the installer. Once I found the installer; My question was who is creating the scheduled task and the window services I saw. If I ran the MSI vs the EXE, what do I lose if anything?
- So, I ran the MSI on its own without capture and found that the MSI is creating scheduled task and the window service.
- So, now what if I try and capture only the MSI install? So, we're back to; How do we get the MSI to NOT install the scheduled task?

As MSI's are editable, let's see if we can edit its default behavior. I downloaded OrcaMSI, with OrcaMSI I was able to find the section that creates the scheduled task and remove it from the MSI directly.

Tables	Action	T...	Source	Target	ExtendedT...
AdminExecuteSequence	SetINSTALLFOLDER	51	INSTALLFOLDER	[DESKTOPAPP_INSTALLFOLDER]	
AdminUISequence	WixFailWhenDeferred	1025	WixCA	WixFailWhenDeferred	
AdvtExecuteSequence	WixCloseApplications	1	WixCA	WixCloseApplications	
Binary	WixCloseApplicationsDeferred	3073	WixCA	WixCloseApplicationsDeferred	
Component	WixCheckRebootRequired	85	WixCA	WixCheckRebootRequired	
CustomAction	CreateRegistryValuesAction.SetProperty	51	CreateRegistryValuesAction	RegistrySettingsLocation=HKKEY_LOCAL_MACHINE\Software\Articulate\Common\Settings;SsoOnly={SSOON...}	
Directory	CreateRegistryValuesAction	3073	BootstrapperCA	CreateRegistryValuesAction	
Feature	RegisterScheduledTaskAction.SetProperty	51	RegisterScheduledTaskAction	InstallLocation=[DESKTOPAPP_INSTALLFOLDER];FileName=Articulat360_Installer_Service.exe;RegistrySettings...	
FeatureComponents	RegisterScheduledTaskAction	3073	BootstrapperCA	RegisterScheduledTaskAction	
File	DeleteScheduledTaskAction.SetProperty	51	DeleteScheduledTaskAction	RegisterScheduledTaskAction	
Icon	DeleteScheduledTaskAction	3073	BootstrapperCA	DeleteScheduledTaskAction	
InstallExecuteSequence					
InstallUISequence					
LaunchCondition					
Media					
MsiFileHash					
Property					
Registry					
Shortcut					
Upgrade					
WixCloseApplication					
_Validation					

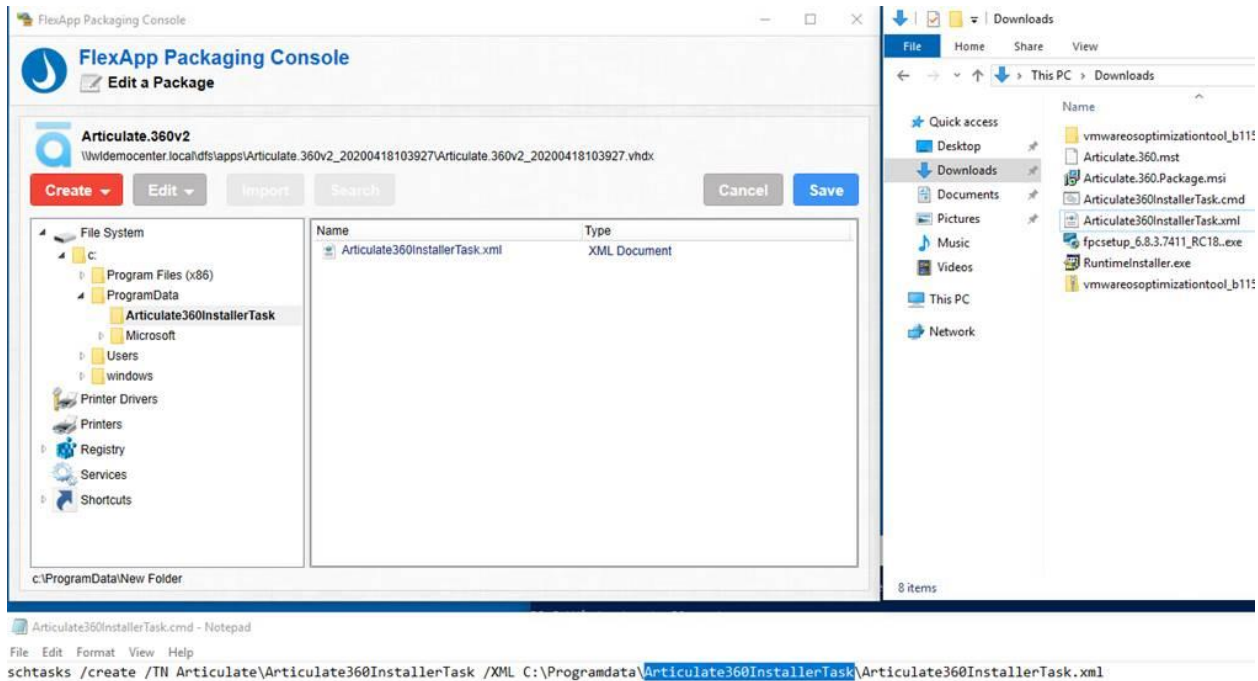
After the MSI was edited directly, I was able to capture the app. This proved that the real issue was, the MSI could not create the scheduled task while we were in capture mode. So, the real issue and above error did not relate at all to the actual error message the application generated.

- But what about the scheduled task? I would like to not deviate from the what the app wants installed just in case it needs it.
- If you remember above, I installed the app natively for testing reasons. Well during that process I exported the scheduled task while I was testing.
- Now, since I have the scheduled task Exported I can script the import.

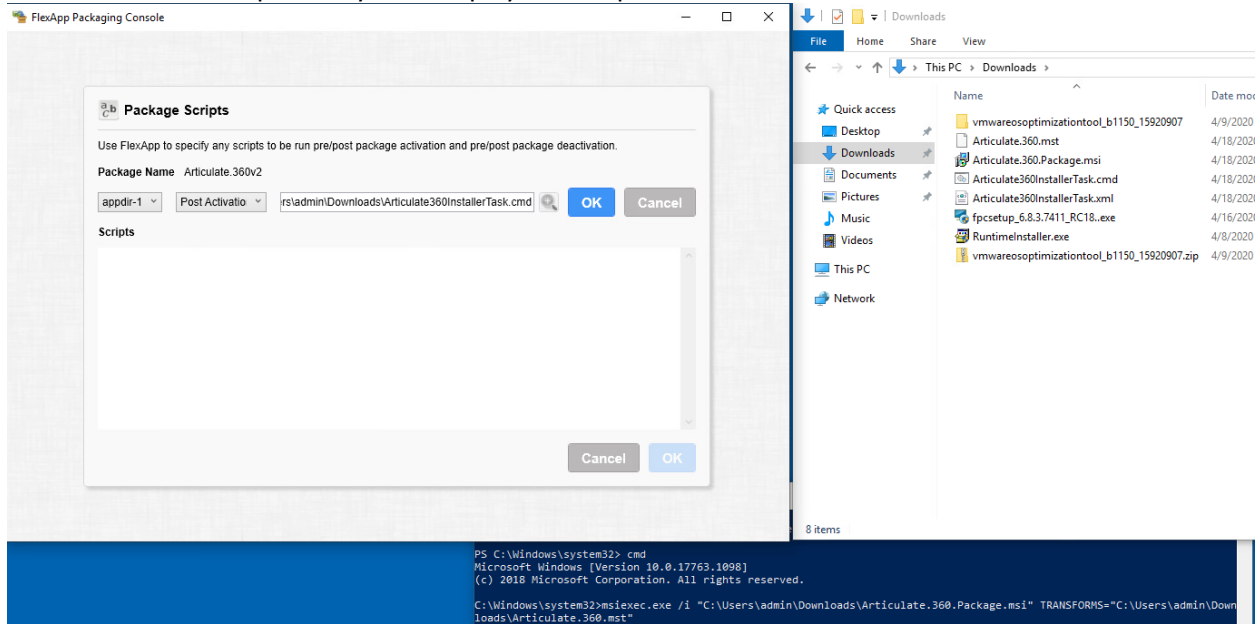
So, to import the scheduled task Post playback we need a 2 things.

1. A place in the layer we can store the scheduled task XML.
2. A script to be played back post that will import the scheduled task.

Post capture, I opened the layer with the editor and created the folder path in the layer and pasted in the XML file to match what my script will want.



Then I added the script the layer to be played back post.



The last thing to think about here is what would the final recipe look like to the customer. So, with that in mind, I didn't like that I had to edit the MSI directly. So, I used the OrcaMSI Tool to record my edits of the MSI, into a MST file. This transform file allows you to change the behavior of the MSI without editing it directly.

### High level outline:

1. Run articulate-360.exe and don't click anything.
2. Go to C:\ProgramData\Package Cache\{D39E6472-95B5-484A-8400-82AC9C55484E}\v1.38.21861.0\Articulate.360.Package.msi and copy the MSI out.
3. Cancel the install
4. Put our Articulate.360.mst, Articulate360InstallerTask.xml and Articulate360InstallerTask.cmd where you can access them from the packager.
5. Open an elevated cmd and have your MSI command line ready.
  - a. `msiexec.exe /i "C:\Users\admin\Downloads\Articulate.360.Package.msi" TRANSFORMS="C:\Users\admin\Downloads\Articulate.360.mst"`
    - i. Edit the pathing as needed.
6. Create a new FlexApp layer and start capture.
7. Run the above MSI command.
8. Save the layer
9. Open the layer for editing.
  - a. Create C:\Programdata\Articulate360InstallerTask
  - b. Copy and paste Articulate360InstallerTask.xml Into the new directory  
Articulate360InstallerTask
10. Save your edits
11. Playback the layer and add a script post playback
  - a. Articulate360InstallerTask.cmd
12. Save the layer.