# Preparing for a staging release:

* Set the CloudApiPublic version, release and build numbers.
* Use git to push everything to master, and record the 7 digits of the hash of the commit you just pushed.
* Add a section to CloudApiPublic\WindowsSdkReleaseNotes.txt that documents all of the changes since the last release. Include the name of the download file to be produced here, including the 7 digit hash you recorded above.
* Use git to push the above change to master, and record the 7 digits of the hash of the commit you just pushed, overwriting the hash you saved previously.
* Change the define in CLDefinitions from cliff servers to staging servers
* Change properties in CloudApiPublic, CloudSetupSdkSyncSample, and CloudSetupSdkSyncSampleSupport to add code-signing with the CloudPlatformCodeSigning.pfx.
  + For CloudApiPublic and CloudSetupSdkSyncSampleSupport, in project properties, use the signing tab. For CloudSetupSdkSyncSample, click Number 6, select Releases, click “SingleImage” on the left, then click the Signing tab. Set the Digital Certificate File, Certificate Password, and set Sign Output Files to “Setup.exe and Windows Installer Package”. Press Ctrl+Shift+S to save all.

# Staging release procedure:

* I have CloudPlatformCodeSigning.pfx in the directory C:\CertBackup\CloudSigning which is used in commands below; so if you change it, then change the commands accordingly
* Select the Debug solution configuration, then Build/Clean Solution. Check that the Clean succeeded.
* Select the ReleaseSampleAppOnly solution configuration, then Build/Clean Solution. Check that the Clean succeeded.
* Select the Release64 solution configuration, then Build/Clean Solution. Check that the Clean succeeded.
* Stay in Release64 solution configuration
* Build BadgeCOM project, check for build success
* Delete BadgeCOM.dll from 3rdParty\bin\release
* Switch to ReleaseSampleAppOnly solution configuration
* Build BadgeCOM project, check for build success
* Open a Visual Studio 2012 Developer Command Prompt “as Administrator”.
* Change directory to ~\3rdParty\bin\Release in command prompt
* Run these commands in command prompt (copy and paste from here):
  + tlbimp.exe BadgeCOM.dll /delaysign /publickey:C:\CertBackup\CloudSigning\CloudPlatformCodeSigning.pub /out:BadgeCOMLib.dll
  + sn.exe -R BadgeCOMLib.dll C:\CertBackup\CloudSigning\CloudPlatformCodeSigning.pfx
    - requires password to certificate
  + copy BadgeCOMLib.dll ..\..\..\CloudSdkSyncSample\bin\release\BadgeCOMLib.dll
  + copy BadgeCOMLib.dll ..\..\..\CloudSdkSyncSample\bin\debug\BadgeCOMLib.dll
* In the CloudApiPublic References, change the BadgeCOMLib reference Specific Version to True.
* Build CloudApiPublic project, check for build success
* Obfuscate CloudApiPublic binary in CloudApiPublic\bin\Release and copy from CloudApiPublic\bin\Release\Obfuscated to CloudApiPublic\bin\Release, CloudSdkSyncSample\bin\Release, and CloudSdkSyncSample\bin\Debug
* Switch to Debug solution configuration
* Build CloudSdkSyncSample project, check for build success
* Switch to ReleaseSampleAppOnly solution configuration
* Build CloudSdkSyncSample project, check for build success
* Build CloudSetupSdkSyncSampleSupport project, check for build success
* Build CloudSetupSdkSyncSample setup project, check for build success
* Copy CloudSdkSetup.exe from the setup project output to a convenient location like C:\
  + Copy from C:\CloudSetupSdkSyncSample\Express\SingleImage\DiskImages\DISK1
* Resource hack the copied CloudSdkSetup.exe:
  + Change the OriginalFilename field to an empty string in Version Info -> 1 -> 1033 and compile the script.
  + Change the Icon Group -> 100 -> 0 by replacing resource with ~\Artwork\cloudForInstallShield.ico
  + Save changes as CloudSdkSetup.exe in the same “copied to” location
* In the previous VS Developer Command Prompt windows, change directory to where you placed the “copied to and modified” CloudSdkSetup.exe.
* Run these commands in command prompt (replace <password> with the certificate password):
  + signtool remove /c CloudSdkSetup.exe
  + signtool sign /f C:\CertBackup\CloudSigning\CloudPlatformCodeSigning.pfx /p <password> CloudSdkSetup.exe
* Check that the CloudSdkSetup.exe file is properly signed:
  + In Explorer, right-click the file and select Properties.
  + Select the Digital Signatures tab.
  + Click the “Citrix Systems, Inc.” item in the signature list.
  + Click the Details button.
  + Under Digital Signature Information it should say “The digital signature is OK.”
* Zip CloudSdkSetup.exe into a zip file with the naming convention CloudSDK-v0.1.<hash>.zip where the 0.1 is the version and can be incremented, and the hash is the first 7 characters of the current git commit that you recorded earlier.
* The zip file is the completed release.
* @@@@@@@@@@NOTE: DO NOT CHECK-IN THESE CHANGES TO GITHUB!!!!!! Including .pfx files copied to the projects where you changed the settings (below). These files should be ignored by gitignore, but remove them anyway. Procedure:
  + In GitExtensions, look at the commit list.
  + Reset the CloudApiPublic.csproj file if it contains only signing and BadgeCOMLib.dll changes.
  + Reset the CloudSetupSdkSyncSample.isl file if it contains only signing changes.
  + Reset the CloudSetupSdkSyncSampleSupport.csproj file if it contains only signing changes.
  + Reset the CLDefinitions.cs file if it contains only the switch to the STAGING servers.
  + C:\Source\Projects\win-client\CloudApiPublic\CloudPlatformCodeSigning.pfx
  + C:\Source\Projects\win-client\CloudSetupSdkSyncSampleSupport\CloudPlatformCodeSigning.pfx

# Smoke Test:

* Unzip the CloudSdkSetup.exe program.
* Check that the .exe file is properly signed:
  + In Explorer, right-click the file and select Properties.
  + Select the Digital Signatures tab.
  + Click the “Citrix Systems, Inc.” item in the signature list.
  + Click the Details button.
  + Under Digital Signature Information it should say “The digital signature is OK.”
* Install CloudSdkSetup.exe.
* Select the option to show the Readme file.
* Click Finish. You should see the HTML docs.
* On the start menu, show the Cloud.com\Cloud folder. Then:
  + Test “Show SDK Documentation”
  + Test “Show SDK Release Notes”
  + Test “VS CloudSdkSyncSample Solution”
* In VS2012, build the app and test it as documented below for the binary (CloudSdkSyncSample.exe).
* Test CloudSdkSyncSample.exe:
  + On the start menu, click “Start CloudSdkSyncSample”.
  + Fill in the required info on the main window.
  + Fill in the trace info on the Advanced Options dialog.
  + Click Save Settings on the main window.
  + Click Reset Sync on the main window.
  + Clear the cloud folder.
  + Click Install Badging.
  + Click Start Syncing.
  + Click Sync Status…
  + Verify that badges appear on the cloud folder Explorer window.
  + Verify that the expected behavior occurs on the sync status window.
  + Verify that the sync status window eventually shows “synced”.
  + Close the sync status window.
  + Click Stop Syncing on the main window.
  + Click Uninstall Badging.
  + Close the application.