Preparing for a staging release:

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, 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.

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

Staging release procedure:

In Debug solution configuration, clean solution

In ReleaseSampleAppOnly solution configuration, clean solution

In Release64 solution configuration, clean solution

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:

* 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:\

Resource hack the copied CloudSdkSetup.exe to change the OriginalFilename field to an empty string in Version Info -> 1 -> 1033 and compile the script; then 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

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.

The zip file is the completed release.

DO NOT CHECK-IN TO GITHUB!!!!!! Including .pfx files copied to the projects where you changed the settings. These should be ignored by gitignore, but remove them anyway.

* C:\Source\Projects\win-client\CloudApiPublic\CloudPlatformCodeSigning.pfx
* C:\Source\Projects\win-client\CloudSetupSdkSyncSampleSupport\CloudPlatformCodeSigning.pfx