This is an engineering release to evaluate the Intel optimized WebRTC for UWP stack for Video calling feature. This release is based on open source project ‘WebRTC for UWP’, release m66 at commit ‘f0ba12’. Please refer to the Open source project for more details at <https://github.com/webrtc-uwp/webrtc-uwp-sdk>

This document lists the prerequisites and instructions to compile and run the open source PeerCC application with Intel library. Please refer to Release notes document for known issues.

**Prerequisites:**

1. Windows RS4 or newer with Intel graphics driver version of 23.20.16.4979 or newer. Tested on RS5 and Intel KabyLake (7300U) platform with driver version 25.20.100.6373
2. Built-in and/or external camera, microphone and speaker to make a video call
3. Visual studio 2017
4. Windows SDK 16299 (available from archive page <https://developer.microsoft.com/en-us/windows/downloads/sdk-archive>)

**Note1:** When installing the SDK, include the feature "Debugging Tools for Windows" which is required to run prepare.bat. Note that the SDK install as part of Visual Studio does not include this feature.

**Note2:** Please make sure to uninstall any other Windows SDK version above 16229 if it is installed

1. Python 2.7(64 bit) and make sure it is in the system PATH environment variable
2. Strawberry Perl(version 5.22.1002)

**Clone and Build WebRTC for UWP SDK:**

1. **Clone the upstream source and checkout m66**
   1. $ git clone --recursive <https://github.com/webrtc-uwp/webrtc-uwp-sdk.git>
   2. $ cd webrtc-uwp-sdk
   3. $ git checkout releases/m66
   4. $ git reset --hard **f0ba12**

(<https://github.com/webrtc-uwp/webrtc-uwp-sdk/commit/f0ba12a2097ca6f75702e5faf0abfa354a8e3c6f>)

* 1. $ git submodule update

1. **Compile upstream m66 source**
   1. $ bin\prepare.bat
   2. Load webrtc\windows\solutions\WebRtc.sln in VS 2017
   3. Build PeerConnectionClient.WebRtc client project from VS ‘solution explorer’ under ‘Samples\PeerCC\Client’ for ‘**Release x64’** configuration
   4. Build peerconnection\_server server project from VS ‘solution explorer’ under ‘Samples\PeerCC\Server’ for ‘**Release x86’** configuration

**Note:** It seems like the server only supports x86 build configuration

1. **Build PeerCC application using intel binaries**

Please make sure to follow above steps under “Compile upstream m66 source” before the below steps

* 1. Create ‘bin\Release\_x64’ folder under ‘webrtc-uwp-sdk\common\windows\samples\PeerCC\Client\’ folder
  2. Copy the intel binaries ‘Org.Webrtc.dll’ and Org.Webrtc.winmd’ to ‘webrtc-uwp-sdk\common\windows\samples\PeerCC\Client\bin\Release\_x64’ folder
  3. Load webrtc-uwp-sdk\webrtc\windows\solutions\WebRtc.sln in VS 2017
  4. Add reference to ‘PeerConnectionClient.WebRtc’ client project from VS ‘solution explorer’ under ‘Samples\PeerCC\Client’
     1. Right click on the ‘PeerConnectionClient.WebRtc’ project and select ‘Add->Reference’
     2. Go to ‘Projects’ page and Uncheck Org.Webrtc.Uwp project
     3. Go to ‘Browse’ page, browse to Intel binaries folder (see path in step b), select ‘Org.Webrtc.winmd’ file and click ‘Add’ button
     4. Select the ‘Org.Webrtc.winmd’ in ‘Reference Manager/Browse’ page and click ‘Ok’ button
  5. Build ‘PeerConnectionClient.WebRtc’ client project for ‘**Release x64’** configuration
  6. Verify that the Intel binaries are used
     1. Right click on ‘webrtc-uwp-sdk\common\windows\samples\PeerCC\Output\PeerCC.Client\WebRtc\bin\x64\Release\Org.Webrtc.dll’ and select properties.
     2. Select ‘Digital Signatures’ tab and check the signer is Intel

1. **Create application package**
   1. Remove **Assets\wide310x150Logo.png** from **Package.WebRtc.appxmanifest.** Click onPackage.WebRtc.appxmanifest file in VS solution explorer and go to **Visual Assets** Tab and remove **Assets\wide310x150Logo.png** from Wide Tile line and save
   2. Delete the ‘PeerConnectionClient.WebRtc\_TemporaryKey.pfx’ from PeerConnectionClient.WebRtc project in VS. This is to avoid certificate expired error message during package creation
   3. Right click on PeerConnectionClient.WebRtc project, select **Store** then select **Create App Packages…**
   4. Select “**I want to create packages for side loading**” and then press **Next** button.
   5. Select ‘Release x64’ configuration and then press **Create** button.

**Notice the Package path:** webrtc\_uwp-sdk\common\windows\samples\PeerCC\Client\AppPackages\

**Install PeerCC Application:**

**Note:** Install the UWP app on **two** RS4 or newer OS systems **(E.g. System1 and System2)**

1. Copy the UWP app package from ‘webrtc-uwp-sdk\common\windows\samples\PeerCC\Client\AppPackages’ folder to the desired folder (e.g.: C:\Users\<xxx>\Desktop) on the deployment system1
2. Browse to the app package directory
3. Install certificate
4. Right click on PeerConnectionClient.WebRtc.xxx.x64.appxbundle ->properties -> Digital Signatures tab ->select the signature -> Click ‘Details’ ->View Certificate

->Install Certificate -> select ‘Local Machine’ -> click ‘Next’ -> select ‘place all certificates in the following store’ option

-> click ‘Browse’ and select ‘Trusted Root Certification Authorities’ -> click ‘Ok’ -> Next -> Finish

1. Install PeerConnectionClient.WebRtc.xxx.x64 package
2. Double click on PeerConnectionClient.WebRtc.xxx.x64.appxbundle and click Install.
3. Repeat above steps I, II, III, IV on system2

**Run Server and Client applications**

1. Copy **webrtc\_uwp-sdk\webrtc\xplatform\webrtc\out\win\_x86\_release\peerconnection\_server.exe** to the desired folder (e.g.: C:\Users\<xxx>\Desktop) on system1
2. Go to C:\Users\<xxx>\Desktop and run **peerconnection\_server** application by double clicking on it
3. Launch the PeerConnectionClient.Webrtc application from the windows ‘Start’ menu on system1
4. Configure the application settings by providing server IP address, codec, resolution and frame rate by selecting ‘setting’ icon

**Note: Please select h264 codec to evaluate Intel optimizations**

1. Press **Connect** button.

If you see a ‘failed to connect to server’ message then follow below steps to enable loopback network connection to run server and client on same system

* 1. Run these commands from PowerShell

$ get-appxpackage -name "\*PeerConnection\*" to get the package family name

$ checknetisolation loopbackexempt -a -n="packagefamilynamehere". Please replace the “packagefamilynamehere” with actual name

1. Repeat above steps III, IV and V on system2. It will display peer system’s name in **Available Peers** list if and only if the peer system is already connected to the same server IP.
2. Select the peer’s name from the **Available Peers** list and press call button. Please wait for the call to be established. Please check your firewall settings if you don’t see video stream within ~10 sec, you might have to disable firewall.
3. To disconnect the call, press ‘disconnect’ button