Building the ASCOM Platform

# Prepare the Build Environment

A 64bit Windows OS version 10 or later is required for the build environment.

This build process was validated in a fully updated VirtualBox VM running Windows 11 64 bit. Please install the following components to create the ASCOM Platform Windows build environment.

Never install the ASCOM Platform in this instance. The purpose of this is to ensure that only project references are used within the solution and that there are no unintended dependencies on pre-installed ASCOM components.

## Windows

* Make sure that .NET 3.5 support is enabled on the PC. This must be turned on manually in Windows 10/11 through:
  + Win 10: the “Turn Windows features on and off” option in “Uninstall or change a program”.
  + Win 11: the “Additional features " link at the bottom of the “Enable features” menu.

## VS 2019

A VS 2019 install is required only in order to get the .NET 3.5 and 4.0 targeting packs.

* Download the VS2019 installer and just install the editor with no workloads.
* In addition, install these individual components:
  + .NET Framework 3.5 development tools
  + .NET Framework 4 targeting pack

## VS 2022

* Download the Visual Studio 2022 Community installer from [https://www.visualstudio.com](https://www.visualstudio.com/)
* Run the installer and select these workloads:
  + .NET Development
  + Desktop Development with C++” workloads
  + Visual Studio Development
* Launch VS2022 and sign in with your Microsoft account.
* Close VS2022

## Notepad ++

* Download from <https://notepad-plus-plus.org>
* Run the installer and take default options.

## Git For Windows

* Download from <https://git-scm.com/download/win>
* Run the installer and select Notepad ++ on the editor selection page
* Take other options at default

## GitHub Desktop

* Download from <https://desktop.github.com/>
* Run the installer with default options
* Sign in to GitHub.com with your GitHub credentials
* Close GitHub Desktop

## InstallAware X15

InstallAware is a purchased product and is only licensed for use by one developer at a time. However, our “Studio” license entitles us to install as many “build machines” as required. The components below represent a build machine install and no other IA components should be installed.

* Request the installer from Bob Denny or Peter Simpson who will also supply the access password.
* Run the installer and select "Studio" (not “Studio Admin”) as the product type.
* Select Custom Install and deselect all Runtime options except for:
  + "Windows installer"
  + "Windows installer (x64)"

## MS Build Community Tasks

* Download the version 1.5 MSI installer from <https://github.com/loresoft/msbuildtasks/releases>
* Run the installer with default options.

## ASCOM Platform Build Script

* Create the directory C:\ASCOM Build on the build machine.
* Copy the contents of the Repository “Build Process” folder to the C:\ASCOM Build folder.
* Edit the:
  + <GitCheckoutBranch> variable near the head of the BuildPlatform.msbuild file to select the develop or master branch as appropriate.
  + BuildType PropertyGroup content to be Release or ReleaseCandidate
  + VS compiler build to be Debug or Release

## Sandcastle Help File Builder

* Download the guided installer from <https://github.com/EWSoftware/SHFB> making sure that the version matches the version last used to edit the Help file sources.
* Run the installer:
  + Install the HTML Help 1 Compiler from the MS web site when prompted, Ignore the message about already having a newer version of the Help
  + Go back to the SHFB installer and click the “Search again” button on the HTML Help 1 page, which will now find the HTML Help 1 compiler.
  + Click the “Install SHFB” button on the Sandcastle Help File Builder and Tools page.
  + Install the Visual Studio package.
  + Don’t install the MAML snippets.
  + Move through remaining dialogues and close the installer script.
* Reboot the PC or VM.

## VSIX Sign Tool

* Install the dotnet “Sign” tool as described here: <https://learn.microsoft.com/en-us/visualstudio/extensibility/signing-vsix-packages?view=vs-2022>

## Code Signing Certificate

* Export your code signing key, including its private key into a PFX file.
* Copy the PFX file to a folder of your choice on the build machine.
* Right click the PFX file and select “Install PFX”
* Select “Current User”
* Click “Next” twice.
* Enter your certificate’s password but do not enable string private key protection because this will continually interrupt the build process prompting for the key password
* Click “Next” twice and then “Finish” to import the certificate
* You must also create the following USER environment variables:
  + SIGN\_CERTIFICATE="Full path to your code signing PFX file"
  + SIGN\_CERT\_HASH=certificate’s SHA 256 hash
  + SIGN\_PASSWORD=AnyPasswordRequiredToUseYourCertificate

# How to Build the ASCOM Platform

## Process

* Open a Visual Studio 64bit command prompt
* Change directory to C:\ASCOM Build
* Run the Build.cmd command file.
* At the end of a successful build the Platform installers, other installable or publishable components and the build log will be found in the C:\ASCOM Build folder.