* **Tools**

These are the tool which we have used to build the msi installer.

* WiX Toolset
* Jdk 8.

Tools required for signing msi

* Signtool

Follow the below steps to get signtool

- Download and install windows SDK from micresoft.

- Note the full path to the signtool.exe application, located to the bin directory of the Windows SDK installed  
(path will be something like 'C:\Program Files (x86)\Windows Kits\10\bin\10.0.15063.0\x64\')

* **Environment Variables**

You need to set the path environment variable with WiX Toolset path for compiling and linking your code.

* In **Start**, search for **Control Panel** and then select it.
* Click the **Advanced system settings** link.
* Click **Environment Variables.** In the section **System Variables,** find the PATH environment variable and select it. Click **Edit**. If the PATH environment variable does not exist, click **New.**
* In the **Edit System Variable** (or **New System Variable**) window, specify the value of the PATH environment variable <Path to WiX Toolset\bin>. Click **OK**. Close all remaining windows by clicking **OK**.
* Reopen Command prompt window, and run your code.
* **Packaging firmware binaries**

Refer README present at "~\installer\dist\Tomcat\_Server\webapps\firmware\" folder for creating metadata json and packaging latest firmware binaries.

* **Build msi**
* To sign jar and war files, create msi and sign msi, run make\_all.bat (/installer/Windows/WIX) file.

To run this file follow the below steps

* Edit the file using any editor
* Set the below variables with appropriate values  
  *PathToCACerticate  
  CertificatePassword  
  TimeStampURL  
  AliasName  
  MSIFilePath  
  SignToolPath  
  ProductVersion*
* Save and close the file.
* Run the file.
* To create msi without signing, just run the setup\_make.bat ( /installer/Windows/WIX/) file.

Options:   
-p : Specify the path to the base folder of the source. Absolute paths can be used.

-v : Specify the ProductVersion.

Example: setup\_make.bat -v 3.5.1.0 -p ..\

Once the build is successful it will create .msi file in “installer/Windows/build” folder.

Run this .msi file for installation.

* **Description of code files**

To build an MSI , the compiler processes the source files (.wxs , .wxl and .wxi) and produces object files (.wixobj). These objects files are then consumed by the linker, which produces Windows Installer file(.msi).

* **BuildVars.wxi :** This BuildVars.wxi file is same as .h files for C/C++. In this file we have defined macros, which is used in another files by including with the <? include BuildVars.wxi ?> element.
* **setup.wxs :** This setup.wxs is the top level source file, which contains the product information, structure for the installation, actual components, features, custom actions and properties.
* **UserInterface.wxs :** This file contains the information related to each dialog page.
* **ProgressText.wxs :** This file contains all the timing relative actions for InstallExecuteSequence, InstallUISequence, AdminExecuteSequence and AdminUISequence.
* **ErrorText.wxs :** This file contains the information about the errors.
* **Common.wxs :** This file contains the commonly used UI text.
* **English-US.wxl :** This English-US.wxl file contains the set of strings used for localizing a product into a specified culture.
* **sign\_jar\_war.bat :** This file is used to sign the jar and war files.
* **setup\_make.bat :** This file is used for building .msi
* **sign\_msi.bat :** This file is used to sign .msi with CA certificate.
* **make\_all.bat :** This is the master script which calls the sign\_jar\_war.bat, setup\_make.bat and sign\_msi.bat sequentially.
* **Logging**

For creating an installation log, you can use a command line which looks like this:

msiexec /i "<PathToPackage>\Example.msi" /L\*V "C:\log\example.log"