Developing Tizen Apps
with the Tizen Studio

* This document is based on Tizen Studio 1.0
Table of Contents (1/2)

Prerequisites for the Tizen Studio 4
  System Requirements for Tizen Studio 5
  System Requirements for Tizen Emulator 7
  JDK Requirements 9

Tizen Studio Installation 12
  Downloading the installer 13
  Installing the Tizen Studio 14
  Installing Additional Packages 18
  Confirming and Removing Packages 25
  Uninstalling the Tizen Studio 26
# Table of Contents (2/2)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running Applications on the Emulator</td>
<td>27</td>
</tr>
<tr>
<td>Launching the Tizen Studio</td>
<td>28</td>
</tr>
<tr>
<td>Creating the Application Project</td>
<td>37</td>
</tr>
<tr>
<td>Generating the Author Certificate</td>
<td>42</td>
</tr>
<tr>
<td>Creating the Emulator Instance</td>
<td>51</td>
</tr>
<tr>
<td>Running the Application on the Emulator</td>
<td>55</td>
</tr>
<tr>
<td>Running Applications on the Target Device</td>
<td>51</td>
</tr>
<tr>
<td>Installing the Certificate Extension</td>
<td>52</td>
</tr>
<tr>
<td>Connecting the Target Device</td>
<td>57</td>
</tr>
<tr>
<td>Generating the Author &amp; Distributor Certificates</td>
<td>62</td>
</tr>
<tr>
<td>Running the Application on the Target Device</td>
<td>78</td>
</tr>
<tr>
<td>Appendix</td>
<td>79</td>
</tr>
</tbody>
</table>
Prerequisites for the Tizen Studio
The following table lists the operating systems supported by the Tizen Studio.

<table>
<thead>
<tr>
<th>OS</th>
<th>Version</th>
<th>Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ubuntu</td>
<td>16.04/14.04</td>
<td>32 and 64</td>
</tr>
<tr>
<td>Microsoft Windows®</td>
<td>10/8/7</td>
<td>32 and 64</td>
</tr>
<tr>
<td>Mac OS® X</td>
<td>10.11 (El Capitan) / 10.10 (Yosemite) / 10.9 (Mavericks) / 10.8 (Mountain Lion)</td>
<td>64</td>
</tr>
</tbody>
</table>

For more instructions on how to check the system specifications, see Appendix: Checking System Specifications (pages 83–84).
The following table lists the processor and memory requirements for the Tizen SDK.

<table>
<thead>
<tr>
<th>Component</th>
<th>Minimum</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor Speed</td>
<td>2 GHz</td>
<td>3 GHz and above</td>
</tr>
<tr>
<td>Processor Type</td>
<td>Intel Dual-Core</td>
<td>Intel Core i5 and above</td>
</tr>
<tr>
<td>Memory</td>
<td>3 GB</td>
<td>4 GB and above</td>
</tr>
<tr>
<td>Disk Space</td>
<td>6 GB free</td>
<td>6 GB and above</td>
</tr>
</tbody>
</table>

**Note**

The Tizen Studio can be installed on AMD processors, but may not perform properly due to absence of the Intel Hardware Accelerated Execution Manager (HAXM), which accelerates the speed of Tizen application emulation.
System Requirements for Tizen Emulator

The following table lists the CPU and screen resolution requirements for the Tizen Emulator.

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Recommended: support for Intel VTx* (Virtualization Technology)</td>
</tr>
<tr>
<td>Screen resolution</td>
<td>Recommended: 1280 x 1024</td>
</tr>
</tbody>
</table>

* For more instructions on how to check for Intel VTx support, see Appendix: Checking CPU VTx Support (pages 85–86).
The following table lists the Graphic Card requirements for the Tizen Emulator.

<table>
<thead>
<tr>
<th>Brand</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>GeForce 8300 GS, GeForce 8500 GT, GeForce GT 220, GeForce GT 430, GeForce GT 530, GeForce GT 330M, GeForce GTX 550Ti, Quadro NVS 290 and later versions</td>
</tr>
<tr>
<td>ATI</td>
<td>RADEON HD 4850, RADEON HD 5450 and later versions</td>
</tr>
</tbody>
</table>

For more instructions on how to update the graphic card driver, see Appendix: Checking and Updating the Graphic Card Driver (pages 87–88).
JDK Requirements

Tizen Studio requires the JDK (Java Development Kit) installation for using the Java Runtime (VM).

Java 8 or higher is required for the Tizen Studio to work properly, and it is recommended to install Oracle JDK 8 for the Tizen.

Do not install Open-JDK.

---

**Note**

Tizen Studio is a development tool based on the Eclipse IDE, which essentially requires the JDK installation as well.

---

For instructions on how to check and uninstall the existing JDK in your system, see [Appendix: Checking and Uninstalling the JDK](page 89).

1. Select Accept License Agreement.
2. Download the installer (.exe file) according to your system environment.

Choose the right version, 32bits (x86) or 64bits (x64).
JDK Requirements

Install the JDK by following the instructions.

Do not change the installation folder from the default location.

Choose the Tizen Studio with IDE installer according to your system environment.

For more information about the IDE and CLI options, see Appendix: Tizen Studio Installation Options – IDE and CLI (page 90).
Double-click on the installer (.exe file), and the installer starts verifying the system requirements prior to the installation of Tizen Studio.
If the JDK is not installed on your computer, the following pop-up appears. Click **OK** and install the JDK before installing the Tizen SDK.

For more information about the JDK requirements and how to install JDK, see **JDK Requirements** (page 7–9).
When the installer window appears, start installation by following the steps:

1. Accept the **Software License Agreement**.
2. Confirm the **Installation Location**. The directory must be empty.
3. Click the **Install** button on the right to continue.
The installer installs the basic packages (IDE for Web application development) for the Tizen Studio.

When the installation is completed, click **Finish**, and Package Manager is launched.

**Tizen Studio Installation**

**Installing the Tizen SDK**
Run the Package Manager to install additional packages required for the development of Tizen applications. Later, you can launch the Update Manager from Windows Start Menu > All Programs > Tizen Studio > Tools > Update Manager.
Package Manager has 3 tabs: **Main SDK**, **Extension SDK**, **Progress**.
Main SDK tab shows all platforms and tools that can be installed.

![Tizen Studio Package Manager](image)
Extension SDK tab shows the list of extra tools and packages for Tizen Studio.
Tizen Studio Installation

Progress tab shows the detail installation status of each package.

Installing Additional Packages
Click the arrow on the left and expand to view the components of the package.
The description of each component is provided at the bottom of the window.
For the purpose of this tutorial, install all components of the 2.3.2 Wearable package by clicking install.
Do not switch off the Package Manager while it is downloading and installing packages. Downloading and installing may take a few minutes.

Select the **Progress** tab to view the detailed progress of the installation.
The install button changes to delete when the packages are installed.
Check **View installed packages** to confirm the installed packages. You can also remove a package if you want.
To uninstall the entire Tizen SDK, launch the Uninstaller from Windows Start Menu > All Programs > Tizen Studio > Tools > Uninstaller.
Running Applications on the Emulator
Running Applications on the Emulator

Launch Tizen Studio from Windows **Start Menu > All Programs > Tizen Studio**. The Tizen Studio is launched successfully as below.
Tizen IDE is composed of the following sections.

For more information about each section, see the Appendix: Tizen IDE Views (page 99).
In this tutorial, a Tizen Web application project is created.

Go to **File > New > Tizen Project.**
Running Applications on the Emulator

The Project Wizard is used to create the basic application skeleton with the required folder structure and mandatory files. Select Template and click Next. Then select profile and version.

For more information about each project type, see Appendix: Tizen Studio Project Types (pages 100–102).
Running Applications on the Emulator

Select Web Application and click Next. Then select the application template.
Running Applications on the Emulator

Change the project name, if you want. Click More properties, if you want to change more properties. Leave all other fields in the Project Wizard to their default values, and click Finish.
Running Applications on the Emulator

The new project is created and shown in the **Project Explorer**.
Running Applications on the Emulator

To install and run applications on devices, you must sign the application package by generating a vendor-specific certificate and registering it in the IDE. (The certificate policy may vary between different vendors and/or models.)

Since an emulator is a pseudo device as well, applications to be installed and run on the emulator must also be signed with certificates generated from the Tizen SDK.

**Security profile** is a tool for managing certificates and it helps generating, registering, and activating certificates in Tizen SDK.
Running Applications on the Emulator

Open **Tizen Tool > Certificate Manager** to set a security profile for generating certificates.
Running Applications on the Emulator

In the Certificate Manager, click the plus icon (+) to create a new profile.
Running Applications on the Emulator

Enter a profile name and click **Next**.
Running Applications on the Emulator

Create a new author certificate or use a previously created author certificate and click Next. Then, fill in the required information.
Running Applications on the Emulator

You can use the default Tizen distributor certificate or another distributor certificate if you have one. In general, the default Tizen distributor certificate is used and you do not need to modify distributor certificates.

A distributor issues a certificate of its own to developers. And the distributor checks if an application submitted to its app store includes its distributor certificate.

- **Use the default Tizen distributor certificate**
  
  The Tizen SDK offers a default Tizen distributor certificate, which you can use to submit your application to the Tizen Store.

- **Select a distributor certificate for another app store**
  
  If you own a distributor certificate for another app store, and want to submit your application to that store, add that distributor certificate to the certificate profile.
Running Applications on the Emulator

Launch an Emulator instance.

Launch the **Emulator Manager** by clicking the icon in the **Connection Explorer**.
Running Applications on the Emulator

Select **wearable 2.3.2 circle**.
Click the Launch button to launch the emulator instance.
Running Applications on the Emulator

The Emulator is launched in its own window, and the structure of the emulator is shown in the **Connection Explorer**.
Running Applications on the Emulator

Right-click on the project title, and select **Run As > Tizen Web Application** while the Emulator display of the is switched on.
Running Applications on the Emulator

The application is launched in the Emulator successfully.
Running Applications on the Target Device
Running Applications on a Target Device

The **Certificate Extension** is an extra package of Tizen Studio. It is required for generating an author certificate for **commercial devices**, such as Samsung Gear S2, and registering it in the IDE.

To install the Certificate Extension, launch **Package Manager** from **Start > All Programs > Tizen Studio > Tools > Package Manager**, or **Tools > Package Manager** in the Tizen Studio menu.
Running Applications on a Target Device

In Extension SDK tab, select the Certificate Extension, and click install.
Click the **Accept** to accept the EULA (End User License Agreement), then Certificate Extension starts to install.
Running Applications on a Target Device

Close the Package Manager when the Certificate Extension is installed.

![Diagram of Package Manager](image)

- **Installing the Certificate Extension**
- **Connecting the Target Device**
- **Generating the Author & Distributor Certificates**
- **Running the Application on the Emulator**
Running Applications on a Target Device

For Gear S2, you must connect the device to the computer using Wi-Fi. For devices that can be connected using USB cable, you can simply connect the device to the computer with an USB cable.

For connecting a Gear S2 device using Wi-Fi:

1. Switch on the Debugging mode in Settings > Gear info.

![Switch on Debugging mode](image)
Running Applications on a Target Device

2. Switch off Bluetooth in Settings > Connections.

3. Switch on Wi-Fi in Settings > Connections and note the IP address. The device and the computer must be connected to the same Wi-Fi network.
Running Applications on a Target Device

Launch the **Remote Device Manager** by clicking the icon in the **Connection Explorer** and click +.

Installing the Certificate Extension

Connecting the Target Device

Generating the Author & Distributor Certificates

Running the Application on the Emulator
Running Applications on a Target Device

Fill in the information about the device and click **Add**.

Select the device to connect and click **Connect** button.

---

**Installing the Certificate Extension**

**Connecting the Target Device**

**Generating the Author & Distributor Certificates**

**Running the Application on the Emulator**
In the first attempt, the connection can fail. In this case, a pop-up appears in the device.

Click the check button to allow Gear to install apps in the device manually.

In the **Connection Explorer**, note that the device is connected successfully.
Running Applications on a Target Device

Select **Tools > Certificate Manager** to generate and register a developer certificate.
Running Applications on a Target Device

Click puls icon (+) to create security profile.
Running Applications on a Target Device

Samsung profile appears because Certificate Extension is installed. Select Samsung profile.
Running Applications on a Target Device

Select device type. Here we choose mobile/wearable, then click Next for create a new certificate profile.
Running Applications on a Target Device

Select Create a new author certificate.
Fill the required information.
Running Applications on a Target Device

Sign in with your Samsung account.

Installing the Certificate Extension
Connecting the Target Device
Generating the Author & Distributor Certificates
Running the Application on the Emulator
Running Applications on a Target Device

Author certificate is created. Click **Next** for creating a distributor certificate.
Running Applications on a Target Device

Select **Create a new certificate** to create a **distributor certificate**.

![Create Certificate Profile](image)

- **Device Type**
- **Certificate Profile**
- **Author Certificate**
- **Distributor Certificate**

**Step 4. Distributor certificate**

- **Create a new distributor certificate**
- **Select an existing distributor certificate**

- **Certificate Path**
- **Password Confirm**
Running Applications on a Target Device

Enter the Device ID and click Next.
Running Applications on a Target Device

The distributor certificate is generated.
Running Applications on a Target Device

You can see generated certificates in the certificate manager.
Right-click on the target device in the Connection Explorer, and go to Permit to install applications.
Running Applications on a Target Device

Click **OK** when the upload is completed.
Running Applications on a Target Device

Right-click on the project title, and select **Run As > Tizen Web Application** while the Emulator display is switched on.
You can view a summary of important information about your computer and operating system by clicking Windows Start menu, right-clicking Computer, and clicking Properties.
You can check the available disk space by clicking the Windows Start Menu > Computer.
Checking the CPU VTx Support


- VT supported by CPU but locked off by mainboard
- CPU VT not supported and HAXM NOT available
- CPU VT supported and HAXM available
- VT supported but locked on by BIOS (BIOS setup required)
You have to set the VT option to **enabled** in the BIOS menu. The entrance point of the BIOS menu depends on the type of the PC mainboard.

### Checking the CPU VTx Support

- **CPU VT supported and controllable in BIOS**
- **CPU VT supported but NOT controllable in BIOS**
You can check your graphic card driver in your system by clicking Windows **Start Menu** ☐, right-clicking **Control Panel**, and clicking **Adjust screen resolution > Advanced Settings**.
You must update to the latest vendor-provided version of the graphic card driver for OpenGL ES acceleration to use the Tizen Emulator.

• For Microsoft Windows® 8/7, check and install the necessary drivers in Control Panel > System and Security > Windows Update.
Confirm the JDK installation in **Control Panel > Programs > Programs and Features**.
The Tizen IDE (Integrated Development Environments) provides a development tool with based on the Eclipse IDE.

- Provides GUI editor and analysis tools with an excellent UX
- Easy for general users and developers
- Requires relatively high system resources

The Tizen CLI (Command Line Interfaces) provides an environment for creating, building, and packaging projects by using commands on a terminal.

- Suitable for high-level developers with automated build system
- Requires relatively low system resources
Tizen SDK Installation Options

You can change the package server, upload a custom SDK image, change the proxy settings, and add extra repositories in the Advanced Configuration.
Tizen SDK Installation Options

You can install or update the Tizen SDK by using one of the following options.

• Configuring the Package Server
  1. In the combo box at the top, select **Package Server**.
  2. In the text box, type an available repository address and click **Refresh**.
  3. In the **Distribution** box, select the distribution you want to download.
  4. Click **Confirm**.

![Tizen SDK Manager Configuration](image-url)
You can install or update the Tizen SDK by using one of the following options.

- Configuring the SDK Image
  1. In the combo box at the top, select **SDK Image**.
  2. Click the folder icon, browse to the SDK image file, and click **OK**.
  3. Click **Confirm**.

![Tizen SDK Manager Configuration](image)
You can configure a proxy to connect to a repository server.

1. Click the **Proxy Settings** tab.
2. Select the configuration you want to use.
3. If you select Manual Proxy Configuration, enter HTTP Proxy and Port in the fields.
4. Click **Confirm**.
The Tizen SDK supports extension packages, called extra repositories, contributed by external developers and companies.

To add an extra repository, click the Extra Repository tab and click Add.

When the Add Repository dialog appears, enter the Name and Repository in the fields. The repository indicates the external server address.
To remove an extra repository, click the Trash icon for the repository in the table, and click **Confirm**.

If the external repository is removed, the packages from the external repository are deprecated.
To modify an extra repository, click the target repository and click **Modify** in the table.

When the **Modify Repository** dialog box appears, modify the **Name** and **Repository** fields. The repository indicates the external server address. You can also click the folder icon on **Repository** to select a local image file or local server location.
To activate or deactivate an extra repository, select the check box of the repository to be activated, or clear the check box of the repository to be deactivated.
Tizen IDE Views

- The **Project Explorer** view provides a hierarchical view of all resources in the workspace. You can open files for editing, or select resources for operations, such as building, packaging, signing, or validating an application.

- The **Connection Explorer** view shows the devices connected to the system. A list of connected devices is available after connecting a device or launching a Tizen Emulator with a tree view.

- The **Outline** view shows the structure outline and Object hierarchy of C/C++ and Javascript.

- The **Property** view shows the property of selected files on **Project Explorer**, such as file path, size, editability, and name.

- The **Problem** view shows the status of build and packaging error or warning.

- The **Console** view shows the target or emulator log when applications are run.

- The **Edit** view is the code editor area.

- The **Toolbar** provides various shortcut icons for building and debugging.
Tizen SDK Project Type

The **Template** tab provides a list of templates with basic structure the developers can use to create their own Web application project.
The **Online Sample** tab provides a list of sample applications demonstrating the usage of various APIs and UI design which developers can refer to.
Right-click on the Emulator screen to view more options on the Emulator. Click **Control Panel**.
In the **Emulator Control Panel**, developers can create and use any data required during application execution and imitate environmental conditions for the device sensors.
Screen Timeout on the Emulator

You can adjust the screen timeout of the emulator in Settings > Display > Screen timeout.

2:14 PM
Mon 4 Jan

Swipe-up
Troubleshooting

You can troubleshoot your Tizen SDK installation by referring to the installation log file. The log file is located in a different directory based on your operating system:

- **Windows® 7**: `%LOCALAPPDATA%\installmanager\install-log`
- **Windows® XP**: `%USERPROFILE%\Local Settings\Application Data\installmanager\install-log`
- **Ubuntu, MacOS®**: `$HOME/.installmanager/install-log`
If the Install Manager fails to connect to the SDK package server:

- Check whether you can access the Tizen Web site using the Web browser on the system where you are installing the SDK.
- If not, install the SDK using the SDK image.
- Check whether you are using a network proxy. If yes, run the Install Manager. Click **Settings** and select the desired option. If the proxy server needs authentication, an ID and password dialog appears.
- The proxy option is not provided anymore.
When installing the Tizen SDK using the SDK image, you may see the **Cannot open SDK image file** dialog box.

The box is displayed if you are using the JDK1.6 updates 26 JDK. To avoid this problem, ensure that you are using the latest version of Java.

See **JDK Requirements** (pages 7–9) for the JDK installation guide.
Troubleshooting

Crashing of Java during SDK Installation

During the Tizen SDK installation, the Java application can crash unexpectedly, and the following window is displayed.

To avoid this problem:

• Ensure that you are using the latest version of the error DLL file, such as MSVCR100.dll. If the error file version is not the latest, upgrade it.
• Update your system with all the Windows-related updates using Windows Update.
Troubleshooting

Install Manager Not Appearing

If the Install Manager does not appear, an error may have occurred in the JDK installation.

You can check your problem by following these directions:

• Run the Install Manager with the –help options.

```bash
./install-manager.exe -help
```

• If you see the following message, the Java libraries are not suitable to Java executables.

"Registry key 'Software\JavaSoft\Java Runtime Environment\CurrentVersion' has value 'XXX', but 'XXX' is required."
The cause is that Java files in System32 do not match the Java Runtime Environment in the registry.

To avoid this problem, copy the `java.exe` file to the System32 directory.

If Tizen Emulator does not appear when you launch it, you may face the same trouble log at `<TIZEN-SDK-DATA>/emulator/vms/<Emulator name>/logs/emulator.log` file.
If the Emulator does not work properly, there might be an error in the HAXM.

The Intel Hardware Accelerated Execution Manager (Intel HAXM) is a hardware-assisted virtualization engine (hypervisor) that uses Intel Virtualization Technology (VT) to speed up Tizen application emulation on a Windows® or Mac® host machine.

HAXM is supposed to be installed automatically as part of the Tizen SDK installation.

However, to install HAXM separately, you can download the Windows installer package using the link below:

http://download.tizen.org/sdk/haxm/1.1.9/win/intelhaxm-tizen.exe
HAXM supports the following Windows® versions:
- Windows® 7 (32/64-bit)
- Windows® Vista (32/64-bit)
- Windows® XP SP2 or later (32-bit only)

Run the installer and accept the UAC prompt, if you are running the stand-alone installer.
Troubleshooting

When the first HAXM install screen is displayed, click **Next**.

You can access the documentation at any time by clicking **Intel HAXM Documentation**.
Troubleshooting

Read and accept the Intel HAXM End-User License Agreement (EULA).
You are prompted to adjust the amount of RAM to be allocated to Intel HAXM.

Note

The installer also functions as a configuration tool for Intel HAXM. To change the memory settings, run the installer again.
The next screen confirms your Intel HAXM memory allocation settings. Click **Install** to proceed.
When the Intel HAXM installation is finished, click **Finish** to exit the installer.
To verify that Intel HAXM is running, open a Command Prompt window and execute the command: `sc query intelhaxm`

If Intel HAXM is working, the command will show a status message indicating that the state is 4 \textbf{RUNNING}.
Starting and Stopping HAXM

To stop or start Intel HAXM, open a Command Prompt window with administrator privileges and execute one of the following commands:

- Stop HAXM: `sc stop intelhaxm`
- Start HAXM: `sc start intelhaxm`

Adjusting Intel HAXM Memory Allocation

To change the amount of memory allocated to Intel HAXM, run the installer again.

Note

Changes to Intel HAXM memory settings take effect when Intel HAXM is restarted. The currently running emulators continue to use the previous memory setting.
Removing HAXM

**Warning**

Close all instances of the Tizen emulator before removing Intel HAXM.

To uninstall Intel HAXM, run the installer again or use the Control Panel.

**Important**

Removing Intel HAXM disables the acceleration of all Tizen emulators, but the Tizen emulator still functions. Installing Intel HAXM again re-enables the emulator acceleration.
Troubleshooting HAXM (BIOS/OS Setting)

Hardware feature required by HAXM may be disabled by BIOS or OS. If your installation fails because of a hardware requirement, please check BIOS/OS as followed suggestion:

- Make sure VT is enabled in BIOS.
- Make sure Execute Disable Bit is enabled in BIOS.
- Make sure Data Execution Prevention is enabled in Windows. On the Windows platform, click **Control Panel > System > Advanced system settings > Advanced tab > Performance section**, **Options** (or **Settings**) button > **Data Execution Prevention**. Make sure that DEP is enabled.