# Prepare your development environment

This document describes how to prepare your development environment to use the Microsoft Azure IoT Hub device management library for C. It describes how to prepare a development environment both in Windows using Visual Studio and in Linux.

# Running simple\_sample in Windows

1. Install [Visual Studio 2015](https://www.visualstudio.com/). You can use the free Community Edition if you meet the licensing requirements. Be sure to include Visual C++.
2. Install [cmake](http://www.cmake.org/). Use version 2.8 or above. Verify that cmake is properly installed and in your path by typing:

**cmake -version**

1. Search for “VS2015 x86 Native Tools command prompt” app and run it. A command prompt will open. Use this command prompt.
2. Create a folder called **c:\dm**. Extract "DeviceClient.zip" into **c:\dm\.** Note you can extract the files into a different folder location of your choice, but make sure you update the paths in the below instructions appropriately. The following steps assume you extracted the files into the c:\dm folder.
3. After you extract all the files, make a "projects" folder in the "c:\dm\azure-iot-sdks\c\iotdm\_client" folder and navigate to that folder using the following commands:

**mkdir projects**

**cd projects**

1. Run the **cmake** command in the project folder as follows:

**cmake ..**

1. Open the "**project.sln**" file using Visual Studio 2015.
2. Go to the Solution Explorer in VS and in the tree view navigate to iotdm\_simple\_sample 🡪 Source Files 🡪 iotdm\_simple\_sample.c
3. Locate the following code in the file:

static const char\* connectionString = "[device connection string]";

1. Insert your device connection string between the double quotes. This is the connection string that was output the CreateDeviceIdentity project. Save the changes.
2. In Solution Explorer, right-click on the **iotdm\_simple\_sample** project and select **Set as StartUp Project** to set the **iotdm\_simple\_sample** application your startup project.
3. Build and run the **simple\_sample** application. When the sample runs, the device is ready to receive and process requests from the service. Notice that when the device connects to IoT Hub, the service will automatically start to observe resources on the device. The device libraries will invoke the device callbacks to retrieve the latest values from the device.

You can use one of the **simple\_sample** or the **iotdm\_app\_template** application in this solution as a starting point for creating your own client applications.

# Running simple\_sample in Linux

This section shows you how to set up a development environment for the Azure IoT device SDK for C on Ubuntu.

**Note:** this setup process requires **cmake** version 2.8.11 or higher and **gcc** version 4.9 or higher.

* You can verify the current version installed in your environment using the cmake --version command.
* You can verify the current version of **gcc** installed in your environment using the gcc --version command. For information about how to upgrade your version of gcc on Ubuntu 14.04, see <http://askubuntu.com/questions/466651/how-do-i-use-the-latest-gcc-4-9-on-ubuntu-14-04>.

## Downloading the Azure IoT SDK to your linux device

* From your linux device can use the wget command to download the zip to your device and then use the unzip command to expand the zip file.

wget <uri of "azure-iot-sdks.zip"> –O azure-iot-sdks.zip

unzip azure-iot-sdks.zip

* Or you can unzip "azure-iot-sdks.zip" on your PC / Mac / Linux desktop then transfer the files over the network to your linux device using [FileZilla](https://filezilla-project.org/) or SCP.

## Building the Azure IoT device management libraries and running the sample

You will build a sample application which relies on the SDK. You first need to update the credentials in the sample app to include a valid Azure IoT device connection string. Edit "/c/iotdm\_client/samples/iothub\_client/samples/iotdm\_edison\_sample/iotdm\_edison\_sample.c" in the following way using vi or other text editor. Replace the "connectionString" string placeholder with the connection string for your device, when you are finished the result should look like the below connection string with your own credentials instead of the placeholders in brackets.

**IMPORTANT**: If you don’t replace items in brackets with your own credentials the sample will not function.

static const char\* connectionString = "Your device connection string";

* In the ssh terminal session type the following commands:

mkdir projects

cd projects

cmake ..

make

* To run sample type:

./samples/iotdm\_simple\_sample/iotdm\_simple\_sample

The linux device is now ready to receive and process requests from the service.