# Installing TensorFlow on Windows 10

## Requirements

1. Windows 10 (64-bit)
2. Anaconda Python 3.7.1

### Pre-req Installation Steps

1. Install Microsoft Visual C++ 2015 Redistributable Update 3 <https://visualstudio.microsoft.com/vs/older-downloads/>
2. Required Packages
   * 'absl-py >= 0.7.0',
   * 'astor >= 0.6.0',
   * 'gast >= 0.2.0',
   * 'google\_pasta >= 0.1.2',
   * 'keras\_applications >= 1.0.6',
   * 'keras\_preprocessing >= 1.0.5',
   * 'numpy >= 1.14.5, < 2.0',
   * 'six >= 1.10.0',
   * 'protobuf >= 3.6.1',
   * 'tensorboard >= 1.13.0, < 1.14.0',
   * 'tensorflow\_estimator >= 1.13.0rc0, < 1.14.0rc0',
   * 'termcolor >= 1.1.0'
3. [Install MSYS2](https://www.msys2.org/)
4. Install Bazel
   * [Download Bazel for Windows from GitHub](https://github.com/bazelbuild/bazel/releases).
   * Look for bazel-<version>-windows-x86\_64.exe, for example bazel-0.16.1-windows-x86\_64.exe.
   * **Tip:** Rename the binary to bazel.exe and move it to a directory on your %PATH% (for example to c:\bazel), so you can run Bazel by typing bazel in any directory.
   * Edit environment variables.
   * Open the “Environment Variables” dialog box from Control Panel or Start Menu, and add or edit the following variables under the “User variables” section:
   * **Edit Path**. Add new entries to the beginning of the list:
   * The directory of bazel.exe. (Example: c:\bazel).
   * The usr\bin directory of MSYS2. (Example: c:\msys64\usr\bin).
   * If you will build **Python** code: the directory of python.exe. (Example: c:\python3).
   * **Add BAZEL\_SH**. Its value must be the path to MSYS2 Bash. Example: c:\msys64\usr\bin\bash.exe
   * **Add JAVA\_HOME** (if you will build **Java** code). Its value must be the directory where you installed the Java JDK 8, for example C:\Program Files\Java\jdk1.8.0\_152. In order to use this with the default local\_jdk javabase, it must be installed on a volume which Windows considers to be **local**, network mounted filesystems will not work.
   * **None of these paths should contain spaces or non-ASCII characters.**

## GPU Requirements

1. NVIDIA® GPU card with CUDA® Compute Capability 3.5 or higher. See the list of CUDA-enabled GPU cards.
2. NVIDIA® GPU drivers —CUDA 10.0 requires 410.x or higher.
3. CUDA® Toolkit —TensorFlow supports CUDA 10.0 (TensorFlow >= 1.13.0)
4. CUPTI: ships with the CUDA Toolkit
5. cuDNN SDK (>= 7.4.1) *TensorFlow will not load without the cuDNN64\_7.dll file.*Install form here: <https://developer.nvidia.com/rdp/cudnn-download>

* + **[4.3. Installing cuDNN on Windows](https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html" \l "installwindows)**
  + *Navigate to your <installpath> directory containing cuDNN.*
  + *Unzip the cuDNN package.*
  + *cudnn-9.0-windows7-x64-v7.zip*
  + *or*
  + *cudnn-9.0-windows10-x64-v7.zip*
  + *Copy the following files into the CUDA Toolkit directory.*
  + *Copy <installpath>\cuda\bin\cudnn64\_7.dll to C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\bin.*
  + *Copy <installpath>\cuda\ include\cudnn.h to C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\include.*
  + *Copy <installpath>\cuda\lib\x64\cudnn.lib to C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\lib\x64.*
  + *Set the following environment variables to point to where cuDNN is located. To access the value of the $(CUDA\_PATH) environment variable, perform the following steps:*
  + *Open a command prompt from the****Start****menu.*
  + *Type Run and hit****Enter****.*
  + *Issue the control sysdm.cpl command.*
  + *Select the****Advanced****tab at the top of the window.*
  + *Click****Environment Variables****at the bottom of the window.*
  + *Ensure the following values are set:*
  + *Variable Name: CUDA\_PATH*
  + *Variable Value: C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0*
  + *Include cudnn.lib in your Visual Studio project.*
  + *Open the Visual Studio project and right-click on the project name.*
  + *Click****Linker > Input > Additional Dependencies****.*
  + *Add cudnn.lib and click****OK****.*

### Step 1

Add the CUDA, CUPTI, and cuDNN installation directories to the %PATH% environmental variable. For example, if the CUDA Toolkit is installed to C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0 and cuDNN to C:\tools\cuda, update your %PATH% to match:

SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin;%PATH%

SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\extras\CUPTI\libx64;%PATH%

SET PATH=C:\tools\cuda\bin;%PATH%

# Installing Keras

Keras is compatible with: Python 2.7-3.6.