

# How to install Dart on Windows?

## Install Dart on Windows

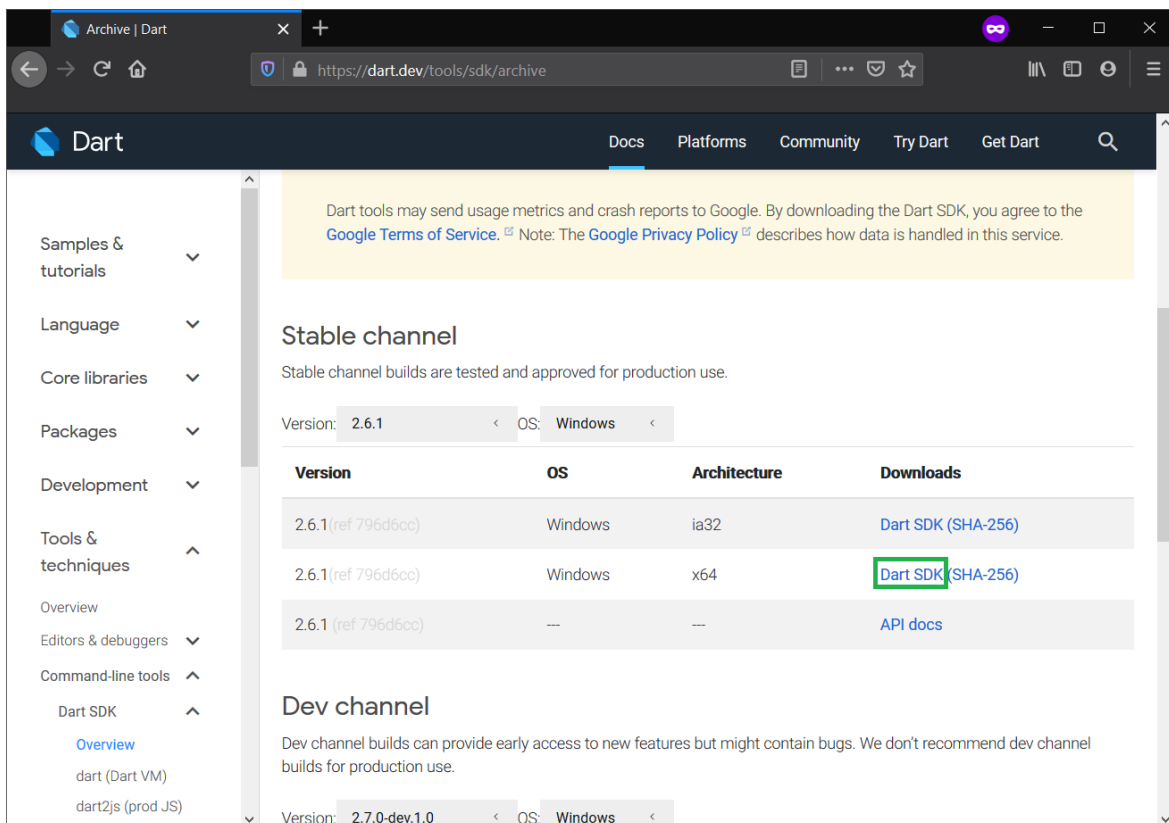
In this tutorial, we will provide a step by step process to get Dart installed on Windows.

Dart SDK comes as a pre-compiled version. So, download and extracting is all you need to do.

### Step 1: Download Dart SDK

Go to Dart SDK archive page.

The URL is <https://dart.dev/tools/sdk/archive>.



Click on the Dart SDK link. The download will start immediately and a zip file will be downloaded.

### Step 2: Extract zip file

Extract the contents of Dart SDK zip file. The contents of the folder would be as shown below.

Name	Date modified	Type	Size
bin	11-11-2019 18:01	File folder	
include	11-11-2019 17:55	File folder	
lib	11-11-2019 17:55	File folder	
dartdoc_options.yaml	11-11-2019 17:55	YAML File	1 KB
LICENSE	11-11-2019 17:48	File	2 KB
README	11-11-2019 17:48	File	1 KB
revision	11-11-2019 17:55	File	1 KB
version	11-11-2019 17:55	File	1 KB

### Step 3: Run Dart

You can run Dart command. Yeah!

Open command prompt and navigate to the bin folder.

Run the command `dart .`

```
Administrator: Command Prompt
D:\software\dart-sdk\bin>dart
Usage: dart [<vm-flags>] <dart-script-file> [<script-arguments>]

Executes the Dart script <dart-script-file> with the given list of <script-arguments>.

Common VM flags:
--enable-asserts
  Enable assert statements.
--help or -h
  Display this message (add -v or --verbose for information about all VM options).
--package-root=<path> or -p<path>
  Where to find packages, that is, "package:..." imports.
--packages=<path>
  Where to find a package spec file.
--observe[=<port>[/<bind-address>]]
  The observe flag is a convenience flag used to run a program with a set of options which are often useful for debugging under Observatory. These options are currently:
    --enable-vm-service[=<port>[/<bind-address>]]
    --pause-isolates-on-exit
    --pause-isolates-on-unhandled-exceptions
    --warn-on-pause-with-no-debugger
  This set is subject to change. Please see these options (--help --verbose) for further documentation.
--write-service-info=<file_name>
  Outputs information necessary to connect to the VM service to the specified file in JSON format. Useful for clients which are unable to listen to stdout for the Observatory listening message.
--snapshot-kind=<snapshot_kind>
```

```
snapshot-kind <snapshot_kind>
--snapshot=<file_name>
These snapshot options are used to generate a snapshot of the loaded
Dart script:
  <snapshot-kind> controls the kind of snapshot, it could be
                    kernel(default) or app-jit
  <file_name> specifies the file into which the snapshot is written
--version
Print the VM version.

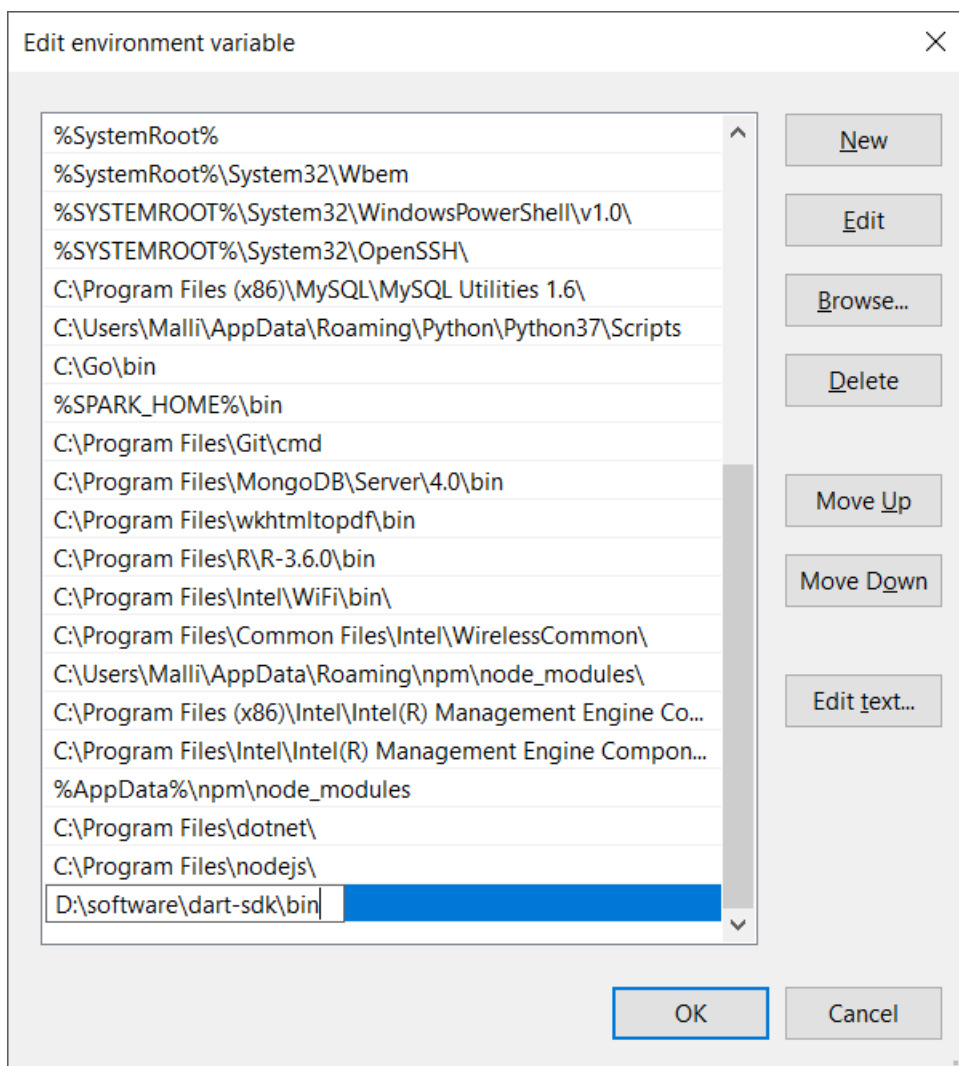
D:\software\dart-sdk\bin>
```

## Step 4: Add Dart Path to PATH Environment Variable

As of now, you can run dart command only from bin folder of dart sdk. To run dart command from anywhere in your file system, add dart bin path to PATH environment variable.

Open **Environment Variables**. Under **System variables**, click on **Path** and click **Edit** button.

**Edit environment variable** window appears. Click on New and paste the dart sdk bin path as shown below.

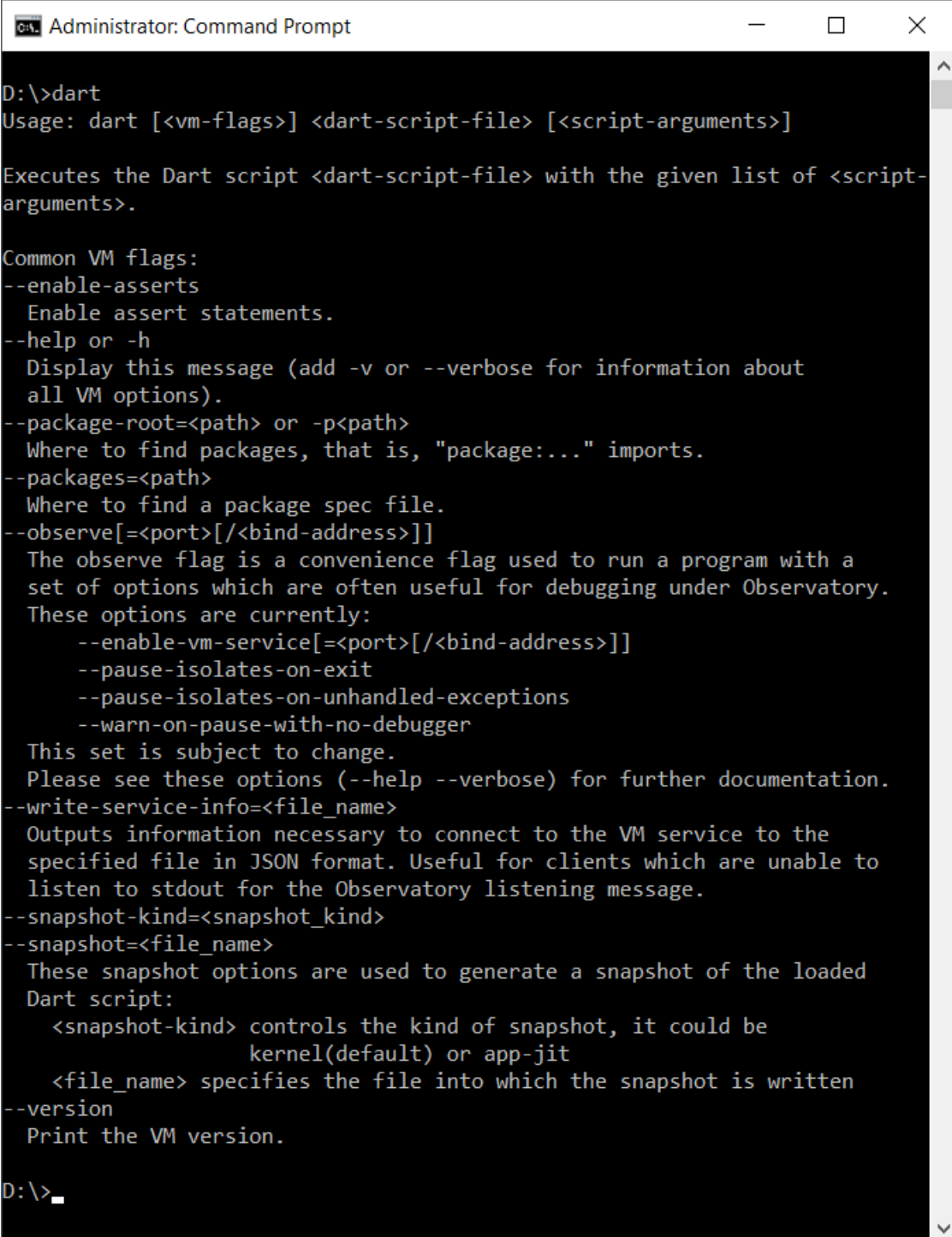


Click on OK. The changes will take effect if you restart your command prompt.

## Step 5: Restart Command Prompt

Close the existing command prompt window and open a new command prompt.

Just run the `dart` command from any working directory. We shall run from `D:\`.



```
Administrator: Command Prompt
D:\>dart
Usage: dart [<vm-flags>] <dart-script-file> [<script-arguments>]

Executes the Dart script <dart-script-file> with the given list of <script-arguments>.

Common VM flags:
--enable-asserts
  Enable assert statements.
--help or -h
  Display this message (add -v or --verbose for information about all VM options).
--package-root=<path> or -p<path>
  Where to find packages, that is, "package:..." imports.
--packages=<path>
  Where to find a package spec file.
--observe[=<port>[/<bind-address>]]
  The observe flag is a convenience flag used to run a program with a set of options which are often useful for debugging under Observatory. These options are currently:
    --enable-vm-service[=<port>[/<bind-address>]]
    --pause-isolates-on-exit
    --pause-isolates-on-unhandled-exceptions
    --warn-on-pause-with-no-debugger
  This set is subject to change.
  Please see these options (--help --verbose) for further documentation.
--write-service-info=<file_name>
  Outputs information necessary to connect to the VM service to the specified file in JSON format. Useful for clients which are unable to listen to stdout for the Observatory listening message.
--snapshot-kind=<snapshot_kind>
--snapshot=<file_name>
  These snapshot options are used to generate a snapshot of the loaded Dart script:
    <snapshot-kind> controls the kind of snapshot, it could be
      kernel(default) or app-jit
    <file_name> specifies the file into which the snapshot is written
--version
  Print the VM version.

D:\>_
```

## Conclusion

In this [Dart Tutorial](#), we learned how to install Dart on Windows, to work with Dart programming.

## **Dart Programming**

◆ Dart Tutorial

⇒ **Install Dart on Windows**

◆ Dart - Hello World

◆ Dart - Variables

◆ Dart - Comments

◆ Dart - If Else

◆ Dart - For Loop

## **Dart String Operations**

◆ Dart - Concatenate Strings

◆ Dart - Split String

◆ Dart - Replace Substring in String

◆ Dart - Find Substring of String

◆ Dart - String Length

◆ Dart - Trim String

## **Dart Exception Handling**

◆ Dart - Try Catch

## **Dart List Operations**

◆ Dart - List

◆ Dart List - Iterate

◆ Dart - Check if List is Empty

◆ Dart - Check if List Contains Element

◆ Dart Reverse List

◆ Dart Join Lists

◆ Dart - Check Equality of Two Lists