



ROS on Windows installation

From: <http://wiki.ros.org/Installation/Windows>

Windows Operating System

- ROS for Windows requires 64-bit Windows 10 Desktop or Windows 10 IoT Enterprise.
- Please ensure that you have Powershell installed and in the system path.
- Exclude `c:\opt` (and later your workspace folder) from real-time virus Scanners, as they can interfere with install and development.

Reserve space for the installation

- Clean and back up any existing data under `c:\opt` before proceeding.
- `c:\opt` is the required install location. Relocation is not currently enabled.
- Please ensure you have 10 GB of free space on the C:\ drive for the installation and development.

Install Visual Studio 2019

Building a ROS project for Windows requires Visual Studio and the Microsoft SDKs for Windows.

- [Download Visual Studio 2019](#)
 - Vcpkg is used for managing dependent libraries. It requires that the English language pack be installed.
 - Include "Desktop development with C++" workload.
 - If you already have Visual Studio 2019 installed, you can [Modify Installation](#)

Install Windows Package Manager

Chocolatey is a package manager for Windows. It is used to make it easy to install tools and libraries needed for building and running ROS projects. The following instructions redirect the chocolatey install location into the c:\opt, so that you can clean or move a ROS environment from that one location.

1. In the Start Menu, find the "x64 Native Tools Command Prompt for VS 2019" item.
2. Right Click, select More then "Run as Administrator"
3. Copy the following command line:
4. `@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"`
 - Paste it into the command window.
 - Approve any prompts
 - Once it has completed, close the command prompt to complete the install.
5. Install Git:
 - Reopen the Visual Studio Command Window as described above.
 - Please install Git using the command here, even if you have it installed as an application:
 - `choco upgrade git -y`
 - Close and Reopen the Visual Studio Command Window as described above.
 - Ensure Git is now available in the Visual Studio command window:
 - `git --version`

Binary Package Installation

To set up ROS for Windows follow these recommended steps:

ROS Last Known Good (LKG) Build Installation

To get things started, install the recommended `desktop_full` metapackage. A Metapackage is a collection of other packages. The Desktop-Full metapackage refers to a number of other packages needed to build, run, debug and visualize a robot.

Open the Visual Studio Command Prompt as Administrator as described above.

```
mkdir c:\opt\chocolatey
set ChocolateyInstall=c:\opt\chocolatey
choco source add -n=ros-win -s="https://aka.ms/ros/public" --priority=1
choco upgrade ros-noetic-desktop_full -y --execution-timeout=0
```

Create a ROS Command Window shortcut

In order to use ROS on Windows, the ROS setup script needs to be called in each command Window. In order to not forget in the future, it is helpful to have a ROS shortcut which does this automatically.

- Create an Administrative command line shortcut for Visual Studio:
- Right click in a Windows Explorer folder, select New > Shortcut
- In the shortcut path, copy the highlighted command line from the following options, depending on the Visual Studio install above:
 - If you are using Community:

```
C:\Windows\System32\cmd.exe /k "C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\Tools\VsDevCmd.bat" -arch=amd64 -host_arch=amd64&& set ChocolateyInstall=c:\opt\chocolatey&& c:\opt\ros\noetic\x64\setup.bat
```

- ~~If you are using Professional:~~

```
C:\Windows\System32\cmd.exe /k "C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional\Common7\Tools\VsDevCmd.bat" -arch=amd64 -host_arch=amd64&& set ChocolateyInstall=c:\opt\chocolatey&& c:\opt\ros\noetic\x64\setup.bat
```

- ~~If you are using Enterprise:~~

```
C:\Windows\System32\cmd.exe /k "C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise\Common7\Tools\VsDevCmd.bat" -arch=amd64 -host_arch=amd64&& set ChocolateyInstall=c:\opt\chocolatey&& c:\opt\ros\noetic\x64\setup.bat
```

- Name the shortcut "ROS"
- Set that shortcut as Administrator
 - Right Click on the shortcut and choose "Properties".
 - Select the Shortcut Tab if not already selected.
 - Press the Advanced button
 - Check the button "Run as Administrator".
 - Press OK on the Advanced properties dialog.
 - Press OK on the "ROS Properties" shortcut dialog.

Using the new Windows Terminal

Microsoft released a new [open source terminal for Windows](#), which includes many improvements over the built in command line, including tabs and appearance customization. You can install it from the [Microsoft Store](#).

To set up the terminal for ROS:

- Find the Windows Terminal from the start menu, right click and select 'Run as Administrator'
- Select settings from the drop down arrow next to the Add Tab (+) Button.
- In the list array in the "profiles" object, add a new block for ROS.

```
"profiles" :
{
  list:
  [
    ...
    {
      "commandline" : "C:\\Windows\\System32\\cmd.exe /k \"C:\\Program Files (x86)\\Microsoft Visual Studio\\2019\\Community\\Common7\\Tools\\VsDevCmd.bat\" -arch=amd64 -host_arch=amd64 && set ChocolateyInstall=c:\\opt\\chocolatey&& c:\\opt\\ros\\noetic\\x64\\setup.bat",
      "guid" : "{xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx}",
      "icon" : "ms-appx:///ProfileIcons/{0caa0dad-35be-5f56-a8ff-afceeeaa6101}.png",
      "name" : "ROS Noetic",
      "startingDirectory" : "c:\\ws"
    },
  ],
}
```

- from a Visual Studio command window, use the command `uuidgen` to generate a globally unique identifier (aka universally unique identifier).
- copy the guid (select the text, then right click to copy)
- Replace `xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx` with the text copied above.
- (Optionally) Set this guid as the "defaultProfile"

```
"alwaysShowTabs" : true,
"copyOnSelect" : false,
"defaultProfile" : "{xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx}",
....
```

When launching the new Windows terminal, please remember to Run as Administrator, by right clicking on the Windows Terminal and Select *Run as Administrator*. There is a [Always Run Terminal elevated feature request](#) that needs to be implemented before this requirement is lifted.

Alternatively, Ctrl+Shift+clicking on the terminal icon in either the start menu or task bar is a handy shortcut to run as administrator.

Stay Up to Date

If you want to update your ROS install, use Chocolatey's upgrade feature.

Open the ROS Command Prompt created above and approve the administrative elevation if not already opened.

Run the following command:

```
set ChocolateyInstall=c:\opt\chocolatey
choco upgrade all -y --execution-timeout=0
```

It is recommended to add `--execution-timeout=0` to accommodate a chocolatey install failure due to slow network.

Uninstall

1. Before the uninstallation, make sure no ROS system or program is running on your system.
2. In a command prompt, run the following command:

```
rmdir /s /q c:\opt
```

Feedback

In case you run into any upgrade/install/uninstall issues, you are encouraged to ask a question on answers.ros.org and tag `windows`.

Consuming a ROS Package


The ROS community has many thousands of [packages](#) which cover different aspects of building a Robot. Many packages have been ported by the community or build on Windows without modification.

Is the package already available?

To consume a ROS package, we recommend the following workflow:

Binary Installation

Determine if there is a binary release of the ROS package.

- Using [ROS Wiki](#) , locate the binary release name and attempt to install using Chocolatey.

- If this succeeds, then you are all set!
- `choco install ros-melodic-<package_name>`

Source Installation

If there isn't a binary release, determine if there is a source only distribution. Here is an example workflow how to create a workspace to test the availability:

```
:: activate the ROS environment
c:\opt\ros\melodic\x64\setup.bat

:: create a empty workspace
mkdir c:\catkin_ws\src
cd c:\catkin_ws

:: generate the released package sources list and its ROS dependencies
:: you can customize the command line to checkout the sources from different channels
:: see the tips section for more details
rosinstall_generator <package_name> --deps --exclude RPP --tar --flat > pkg.rosinstall

:: you can manually edit the pkg.rosinstall for more customizations.
:: see the tips section for more details

:: checkout the sources for real
vcs import --force src < pkg.rosinstall

:: attempt to acquire the external dependencies
rosdep update
rosdep install --from-paths src --ignore-src -r -y

:: now catkin make to build the workspace
catkin_make
```

If everything goes well, now you can activate the development space - by calling `devel/setup.bat` - and use the package.

```
:: activate the development space
devel\setup.bat

:: test the new package is discoverable
rospack find <package_name>
```

Tips for Source Installation

If something has not successfully been built, you can try to the following steps:

- If a Windows port has not yet been released, you can repeat [Source Installation](#) but ask `rosinstall_generator` to use the development branch this time.
- `:: use the development branch`
- `rosinstall_generator <package> --upstream-development --deps --exclude RPP > pkg.rosinstall`

- If using the upstream development branch doesn't help, you can edit `pkg.rosinstall` to switch the `version:` of the broken package to other branch, like `windows` or `init_windows`, if it exists.
- In addition to consuming the upstream repositories, you can check to see if [Microsoft's ms-iot Github organization](#) has a fork of that project and is working on a port. If it does, you can edit `pkg.rosinstall` to point `uri:` to the different fork.
- For the package not registered to the [ROS distributions](#), you can manually create a `.rosinstall` file to maintain a list of repositories to consume.
- [ROS Wiki](#) or [ROS Index](#) are also good resources to search for the package repository.

NOTES

Note: had to uninstall choco, choco uninstall all, then manually removed C:\ProgramData\chocolatey and all the environment variables related to chocolatey

Note: Problem <https://github.com/ms-iot/ROSONWindows/issues>