

# How to setup Utsusemi for Ubuntu on Windows 10 + WSL2

July 6, 2021

Ryoichi Kajimoto

Last modified November 18, 2022

<b>Installation of WSL2 (Windows Subsystem for Linux 2) and Ubuntu 20</b>	<b>1</b>
<b>Installation of X Server (VcXsrv)</b>	<b>1</b>
Installation of VcXsrv	1
Setting of Windows Defender Firewall	2
<b>Setting of the DISPLAY Environment Variable in WSL2</b>	<b>6</b>
<b>Installation of Utsusemi</b>	<b>6</b>
Installation from the Source Code	6
Installation from deb Packages	8
<b>Appendix 1: Autostart of VcXsrv</b>	<b>8</b>
<b>Appendix 2: Use of Windows Terminal</b>	<b>9</b>
<b>Appendix 3: Memory Management</b>	<b>9</b>

## Note for Windows 11 Users

If you have Windows 11 Build 22000 or later, you do not need to do procedures regarding the Xserver (VcXsrv) and the DISPLAY environment variable, because you can use WSL2 supporting Linux GUI applications.

See: [WSL で Linux GUI アプリを実行する | Microsoft Learn](#) (Japanese) or [Run Linux GUI apps with WSL | Microsoft Learn](#) (English).

*Note: Now WSL2's X server runs on display 0. This means that if you have already changed the DISPLAY environment variable, you should reset such a setting. See: [Diagnosing "cannot open display" type issues with WSLg · microsoft/wslg Wiki · GitHub](#).*

## Installation of WSL2 (Windows Subsystem for Linux 2) and Ubuntu 20

You can find a lot of information on the internet. For example, see:

「手動インストールの手順」 in [WSL のインストール | Microsoft Learn](#) (Japanese) or “Manual Installation Steps” in [Install WSL | Microsoft Learn](#) (English).

# Installation of X Server (VcXsrv)

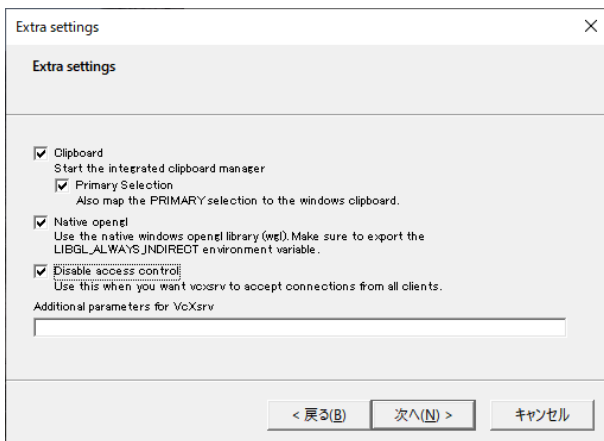
*Note: The following part regarding VcXsrv is unnecessary if you have Windows 11 and the latest WSL2.*

## Installation of VcXsrv

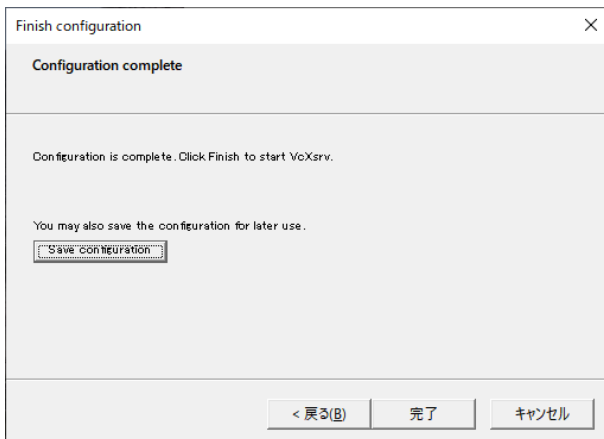
Download the latest installer from [VcXsrv Windows X Server download | SourceForge.net](https://sourceforge.net/projects/vcxsrv/), and install VcXsrv.

Launch Xlaunch.

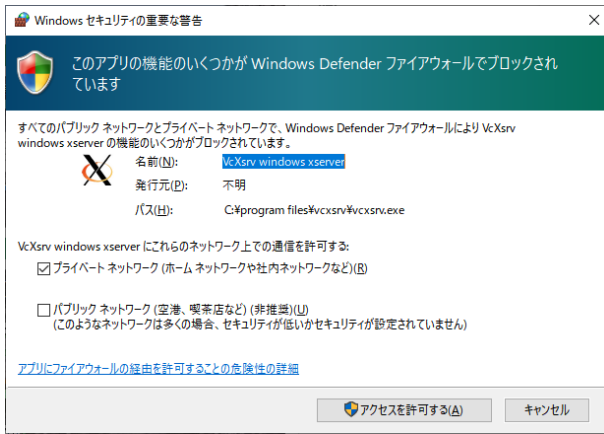
The following window will appear. Most of the default settings are fine, but you have to **Check “Disable access control”** in the third panel.



You may save the configuration to xxx.launch file. Then, click Complete.



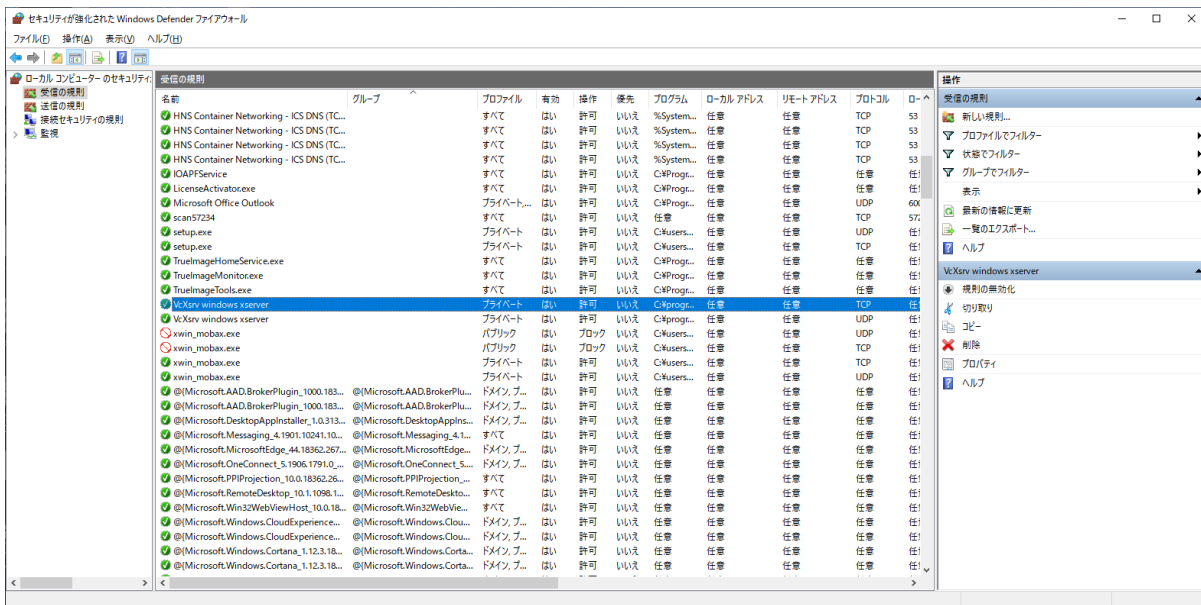
A warning message from Windows Defender will appear. **DO NOT check “public network”** for safety.



Press Ctrl+Shift+Esc to launch Task Manager, and confirm “VcXsrv windows xserver” is running.

## Setting of Windows Defender Firewall

### Launch Windows Defender Firewall

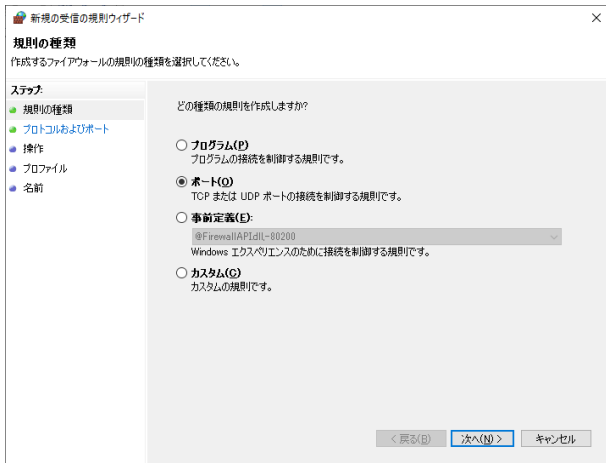


In Inbound Rules, you will find entries of VcXsrv windows xserver.

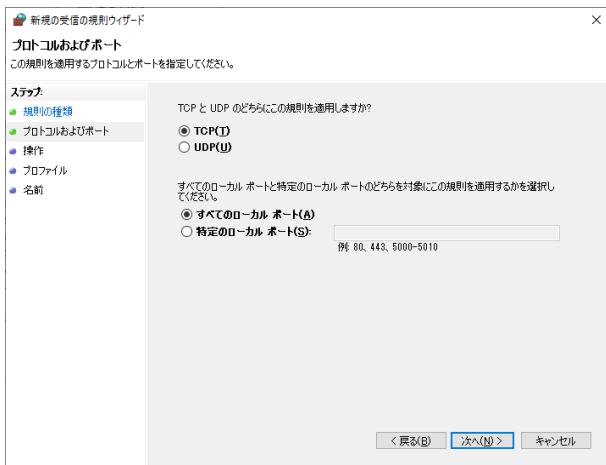
Find the entry of: **Profile = Public, Protocol = TCP.**

If you cannot find such an entry, create a new one by clicking “New Rule...”

Choose “Port”



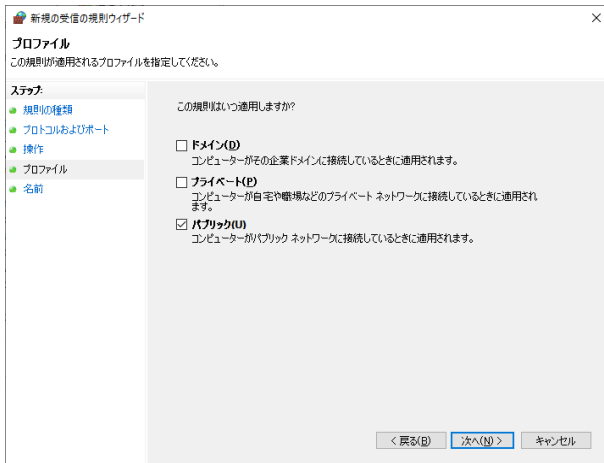
## Choose “TCP” and “All local ports.”



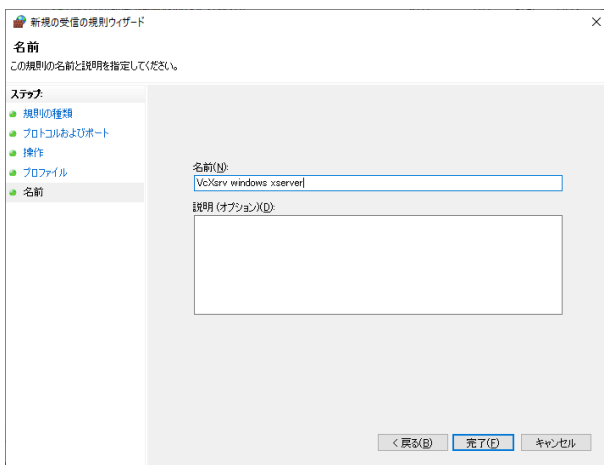
## Choose “Allow connection.”



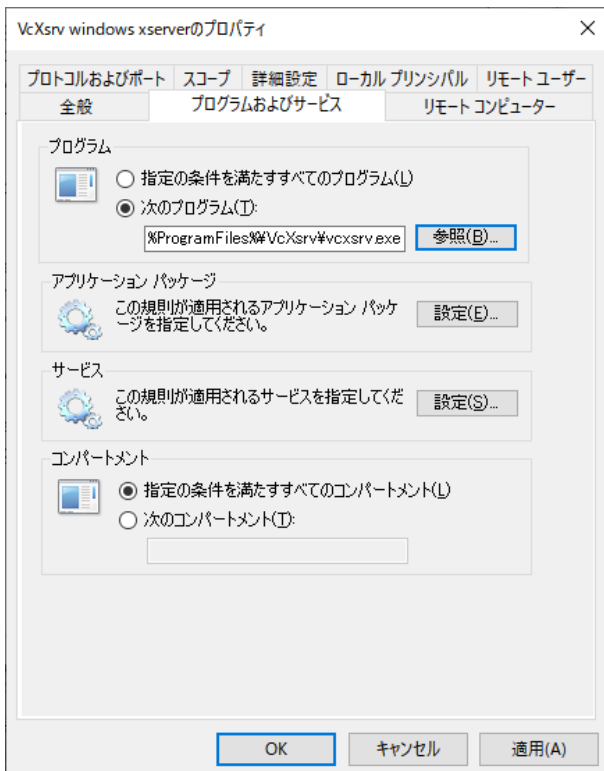
## Choose “Public”



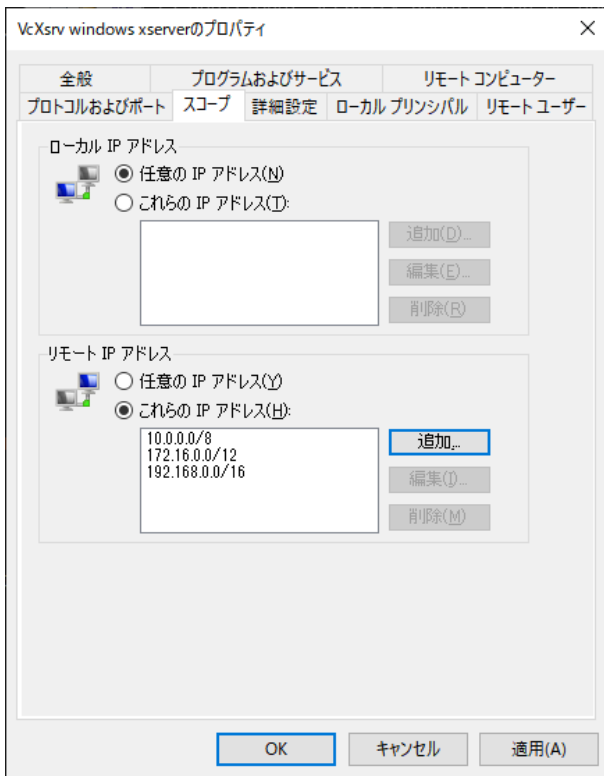
Name the rule.



Double click the newly created entry, and specify the path to the executable file of VcXsrv in the "Programs and Services" tab.



Set the range of private IP addresses in the Remote IP Address in the “Scope” tab. Typically it is 10.0.0.0/8 (class A), 172.16.0.0/12 (class B), or 192.168.0.0/16 (class C).



## Setting of the DISPLAY Environment Variable in WSL2

*Note: This part is unnecessary if you have Windows 11 and the latest WSL2, because X server runs on display 0 (default value).*

You need to set the DISPLAY environment variable to the address of nameserver which is stored in /etc/resolv.conf like this:

```
$ cat /etc/resolv.conf
# This file was automatically generated by WSL. To stop automatic
generation of this file, add the following entry to /etc/wsl.conf:
# [network]
# generateResolvConf = false
nameserver 172.19.240.1
```

To do this, execute the following command on the WSL2 terminal (“\$” at the beginning is the command prompt. The following line is one line.)

```
$ export DISPLAY=$(cat /etc/resolv.conf | grep nameserver | awk '{print $2}'):0.0
```

Check if the DISPLAY variable is correctly set

```
$ echo $DISPLAY
172.19.240.1:0.0
```

Check the settings were done correctly by launching any GUI program. For example,

```
$ xeyes
```

(xeyes program is included in the x11-apps package.)

To set the DISPLAY environment variable automatically, you may add the following line (one line) in one of the configuration files of bash, e.g., .profile.

```
export DISPLAY=$(cat /etc/resolv.conf | grep nameserver | awk '{print $2}' ):0.0
```

## Installation of Utsusemi

The procedure is basically the same as that for Utsusemi for Ubuntu.

See: <https://cdn.mlf.plus/ce/Utsusemi4/install/downloads.html> (Japanese) or [https://cdn.mlf.plus/ce/Utsusemi4/install/downloads\\_e.html](https://cdn.mlf.plus/ce/Utsusemi4/install/downloads_e.html) (English).

However, if you install it from the source code using the installation scripts, you should do a trick on the PATH environment variable as follows:

*(You do not need to do the following if you install deb packages.)*

By default, PATH of Windows is added to PATH of Ubuntu. You can see the current PATH by

```
$ echo $PATH
```

However, some of Windows' PATH causes an error in installing Utsusemi. Then, please set a minimal PATH **temporarily**.

```
$ export PATH=/usr/sbin:/usr/bin:/sbin:/bin
```

```
$ echo $PATH
```

```
/usr/sbin:/usr/bin:/sbin:/bin
```

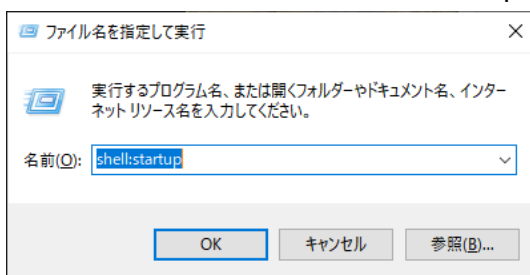
After the installation, the default PATH including Windows' PATH causes no problem.

## Appendix 1: Autostart of VcXsrv

*Note: You do not need VcXsrv to run Linux GUI applications including Utsusemi if you have Windows 11 and the latest WSL2.*

If you save the configuration of VcXsrv into a file xxx.xlaunch, you can double click the file to launch VcXsrv. On the other hand, it would be a good idea to register the xlaunch file in the startup menu, and launch VcXsrv automatically at the starting up of the PC.

Press Window+R to launch the "RUN" panel. Input "shell:startup" and click "OK."



It will open the C:\Users\username\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup folder.  
Create the shortcut of the xlaunch file in this folder.

## Appendix 2: Use of Windows Terminal

The default terminal of WSL2 is based on the legacy command prompt, and it is not functional. Here I recommend using Windows Terminal. This is a highly configurable program developed by Microsoft and can be installed through Microsoft Store.

Once Windows Terminal is installed, the entry of WSL2 is automatically added. If you configure Windows Terminal so that WSL2 is the default program, you can use it as if it is the terminal for Ubuntu.

## Appendix 3: Memory Management

You can change the size of memory and swap file used by WSL2 by editing the configuration file `.wslconfig`. This file should be located at C:\Users\<yourUserName>\.wslconfig . For example, the entry like below will set the memory size and swap file size to 12 GB and 16 GB, respectively.

```
[wsl2]
memory=12GB
swap=16GB
```

After preparing the `.wslconfig` file, shutdown WSL2 for the configuration to take effect by executing:

```
wsl --shutdown
```

See <https://docs.microsoft.com/en-us/windows/wsl/wsl-config> for details.