

# WSLg Initial Impressions

This whitepaper explores the newest feature addition to the WSL platform and reviews its setup, customization & usage facets.

The paper tries to focus on using WSLg as a part of daily driver for enterprise developers and covers some specific usage related to app development, RPA, productivity, etc.



# Table of Contents

1	Introduction .....	3
2	<b>What is WSL ?</b> .....	3
3	<b>Then why WSLg ?</b> .....	3
4	<b>What's under the HOOD!</b> .....	4
5	<b>Setup Experience</b> .....	4
6	<b>App Installation</b> .....	6
7	<b>Application Performance &amp; Usage Notes</b> .....	7
7.1	<b>Productivity Apps</b> .....	8
7.2	<b>Solution Development</b> .....	8
8	<b>Usage in RPA</b> .....	9
9	<b>The final word</b> .....	9



# 1 Introduction

Web development can be as simple as difficult you want it to be. Unfortunately, in most cases it's not the developer's vision who drives the development but it's the business cases, customer eco-system and existing solutions who drive the decisions not to mention cost.

For eons now (atleast it feels like it) Linux has been one of the standout platforms for developing and deploying enterprise grade solutions. If your customer is not an out-in-out Microsoft shop, chances are your applications will be deployed and maintained on Linux clusters. This has gained traction in last decade with the exponential adoption of open-source, security focused & mostly free platforms and solutions. Even Microsoft has embraced this 'openness' and had ported or completely redeveloped some of its platforms to support these Linux based solutions.

The one specific silo where Linux distributions struggled is the rate of adoptions in the workstations especially developer & admin workstations. Many if not most OME's ship their workstation hardware with Windows. Hence developers & system admins always had to resort to either virtualization or dual booting for running their development & support workloads. But this was a hacky-duct tape solution. One of the major factors which made these solutions feel that way is lack of inherent support for the underlying operating system Windows.

Microsoft recognized this use case and came up with WSL.

## 2 What is WSL ?

The Windows Subsystem for Linux (WSL) is a feature of Windows 10 that enables you to run native Linux command-line tools directly on Windows, alongside your traditional Windows desktop and apps. In a nutshell, WSL enables you to run Linux in a Bash shell with your choice of Linux distribution. But the interesting thing here is unlike VM's & dual booting you can also access your local machine's file system (Windows filesystem) from within the Linux Bash shell. This is an absolute boon for developers who can now run & test Linux artifacts without leaving other productivity tools based on Windows.

## 3 Then why WSLg ?

Now if you have followed me till here the next obvious question is **WHAT IS WSLg ?**

As Microsoft defines it:

*WSLg is short for Windows Subsystem for Linux GUI and the purpose of the project is to enable support for running Linux GUI applications (X11 and Wayland) on Windows in a fully integrated desktop experience.*

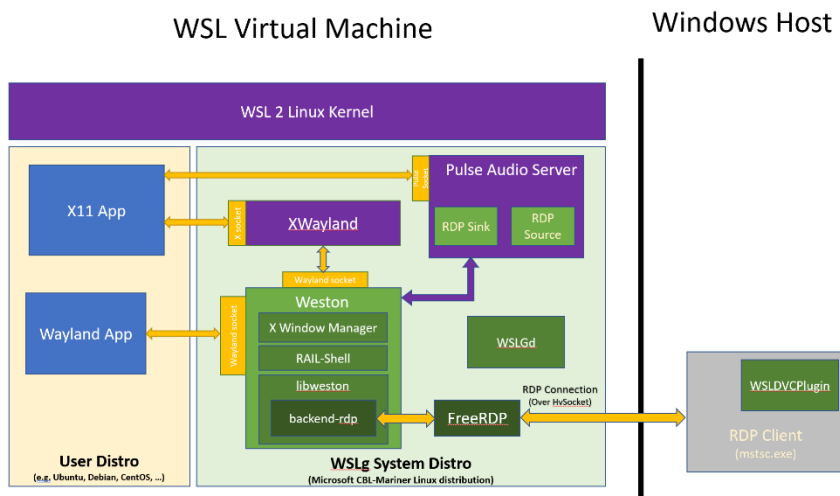
Basically, WSLg expands on the features provided by WSL and provides you an interactable GUI based applications windows build on Linux without you actually having to install a full-fledged Linux distro (*sort-of*) while running your other applications and productivity suites like MS Office on your existing Windows installation.

This comes in handy when you want to develop and run tools which are specifically build and execute in Linux environments.



# 4 What's under the HOOD!

Behind the scenes WSLg implements a very simple two system architecture. The architecture is built on a single WSL2 Linux kernel supporting 2 Linux distros.



The system distro is **Microsoft CBL-Mariner Linux** which contains all the audio & display servers along with all the piping's needed for the User distros GUI & Audio to pass through to an windows RDP client instance. You can learn more about the inner working [at this link](#).

Figure 1: WSLg Architecture ([original Link Here](#))

# 5 Setup Experience

WSLg is still not generally available; hence user needs to on the 'Windows 10 Insider Preview build' from the dev channel.

**TWEAK & TRICK:**

*You can even install WSL & WSLg on Windows 11. The setup steps shown bellow are same for both Windows 10 & 11 tho I noticed the process executed a bit faster on Windows 11; but that might be due to the other CPU & IO optimizations done on Windows 11 rather than any specific optimization for WSL.*

Microsoft has made the installation process of WSLg a breeze and just single command bellow in admin console gets you up and running

```
wsl --install -d Ubuntu
```

This command installs required WSL components and the server components required along with the base Ubuntu image.



```
Administrator: Command Prompt - wsl --install -d Ubuntu
Microsoft Windows [Version 10.0.21354.1]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>wsl --install -d Ubuntu
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
Downloading: WSL Kernel
[=====66.9%=====]
```

Figure 2: Installing WSL & Ubuntu Image

This process might take anywhere from 5 to 15 min. depending on your machine configuration and your internet bandwidth.

Larger components will be displayed with the progress bar to keep users updated on the progress

**NOTE :**

*During Kernel installation or GUI app support installation if you get any error then just rerun the command again.*

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22000.51]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>wsl --install -d Ubuntu
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
Downloading: WSL Kernel
Installing: WSL Kernel
WSL Kernel has been installed.
Downloading: GUI App Support
Installing: GUI App Support
GUI App Support has been installed.
Downloading: Ubuntu
The requested operation is successful. Changes will not be effective until the system is rebooted.
```

Figure 3: Install Completion

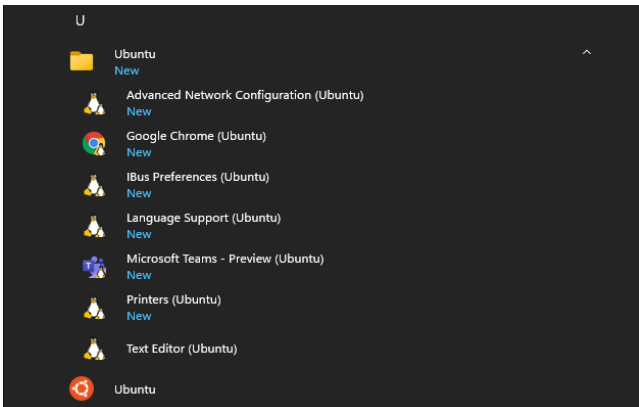


Figure 4: Ubuntu in Start Menu

Once installed you can get "Ubuntu" menu in you Start menu clicking on "Ubuntu" item opens up the Ubuntu CLI.

Any application you install in Ubuntu will be show in this menu for ease of access as well. Sometimes this menu is not visible immediately; in that case restart your host system and after reboot the menu would be visible



During first boot user creation occurs and user can create the username & password of their choice.

Once done user also receives basic statistics and information of the underlying system with some notifications about sudo users and system updates.

```
rohan@DESKTOP-RA1ESVP: ~  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows username.  
For more information visit: https://aka.ms/wslusers  
Enter new UNIX username: rohan  
New password:  
Retype new password:  
passwd: password updated successfully  
Installation successful!  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
System information as of Fri Jun 25 20:51:05 PDT 2021  
  
System load: 0.56 Processes: 8  
Usage of /: 0.4% of 250.98GB Users logged in: 0  
Memory usage: 5% IPv4 address for eth0: 172.27.159.5  
Swap usage: 0%  
  
0 updates can be installed immediately.  
0 of these updates are security updates.  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
rohan@DESKTOP-RA1ESVP:~$  
  
This message is shown once once a day. To disable it please create the  
/home/rohan/.hushlogin file.  
rohan@DESKTOP-RA1ESVP:~$
```

Figure 5: First Boot experience

## 6 App Installation

App installation is a breeze when it comes to adding new applications. Since we have a full-fledged Ubuntu instance, we can use the same toolset & package managers which are provided out of box by ubuntu like 'apt'.

One thing to note here is that during installation of the first GUI application large number of UI dependency packages get downloaded since the original ubuntu images does not come with any GUI or client-side packages. This package list includes packages associated with python, pulseaudio, xserver, etc.

To start off, we installed a simple text editor **GEDIT** using the following command

```
sudo apt install gedit -y
```

The above command instructs ubuntu to use the 'apt' package manager to install package called 'gedit'

Gedit's main package is hardly of few MB's; but Ubuntu downloads all its dependencies which install around 449 packages with 140MB's of download. But it's a onetime chunk of download. Once the base packages are installed other application reuse them and hence don't need re downloading with each new install.

```
Select rohan@DESKTOP-TN4T9V2: ~  
rohan@DESKTOP-TN4T9V2:~$ sudo apt install gedit -y  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
acl adwaita-icon-theme apg aptdaemon aptdaemon-data aspell aspell-en at-spi2-core avahi-daemon avahi-utils bluez  
bubblewrap cheese-common colord colord-data cpp cpp-9 cracklib-runtime crda cups-pk-helper dconf-cli  
desktop-file-utils dictionaries-common dns-root-data dnsmasq-base docbook-xml emacs-common enchant-2  
evolution-data-server evolution-data-server-common fontconfig fprintd gcc-9-base gcr gdm3 gedit-common geoclue-2.0  
gir1.2-accountsservice-1.0 gir1.2-atk-1.0 gir1.2-atspi-2.0 gir1.2-freedesktop gir1.2-gck-1 gir1.2-gcr-3  
gir1.2-gdesktopenums-3.0 gir1.2-gdkpixbuf-2.0 gir1.2-gdm-1.0 gir1.2-geoclue-2.0 gir1.2-gnomebluetooth-1.0
```

Figure 6: Install gEdit



```
x11-xserver-utils xdg-dbus-proxy xfonts-base xfonts-encodings xfonts-utils xml-core xserver-common xserver-xephyr
xserver-xorg xserver-xorg-core xserver-xorg-input-all xserver-xorg-input-libinput xserver-xorg-input-wacom
xserver-xorg-legacy xserver-xorg-video-all xserver-xorg-video-amdgpu xserver-xorg-video-ati xserver-xorg-video-fbdev
xserver-xorg-video-intel xserver-xorg-video-nouveau xserver-xorg-video-qxl xserver-xorg-video-radeon
xserver-xorg-video-vesa xserver-xorg-video-vmware xwayland yaru-theme-gnome-shell yelp yelp-xsl zenity zenity-common
0 upgraded, 449 newly installed, 0 to remove and 1 not upgraded.
Need to get 135 MB of archives.
After this operation, 533 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 libsane-common all 1.0.29-0ubuntu5 [277 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 libtalloc2 amd64 2.3.0-3ubuntu1 [29.5 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal/main amd64 libtevent0 amd64 0.10.1-4 [35.5 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal/main amd64 libwbclient0 amd64 2:4.11.6+dfsg-0ubuntu1 [221 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/main amd64 liborc-0.4-0 amd64 1:0.4.31-1 [188 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/main amd64 libgstreamer-plugins-base1.0-0 amd64 1.16.2-4 [735 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/main amd64 libspand-qlib1 amd64 1.57-0ubuntu3 [90.2 kB]
```

Figure 7: Plethora of dependencies during first GUI app installation

Once application is installed it is either available from the 'Start' menu or you can start the application from the Windows command prompt

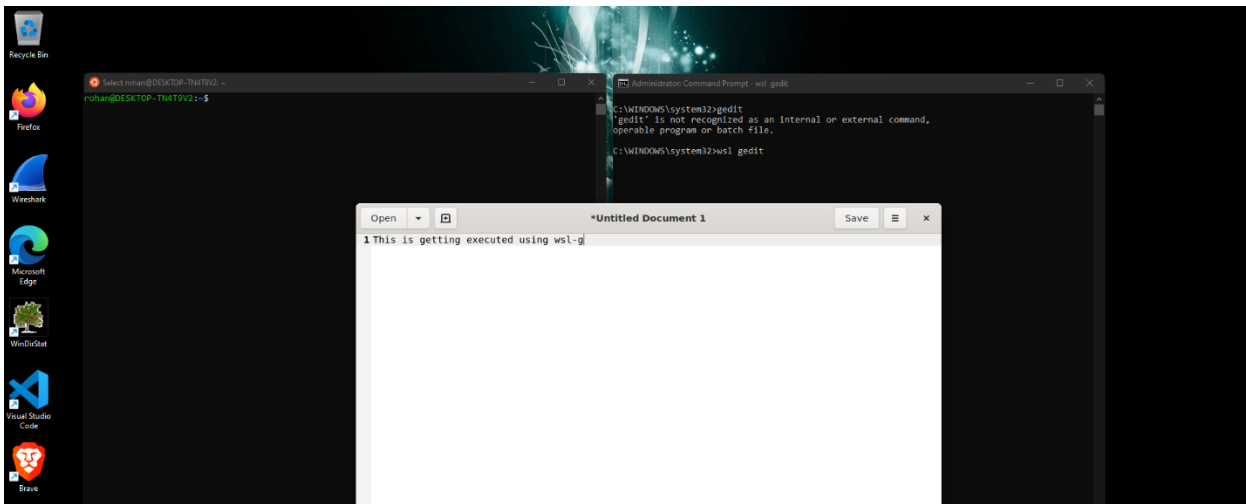


Figure 8: Execute WSLg app from Windows CLI

# 7 Application Performance & Usage Notes

We installed multiple productivity, development, and everyday applications and all seem to work quiet well.

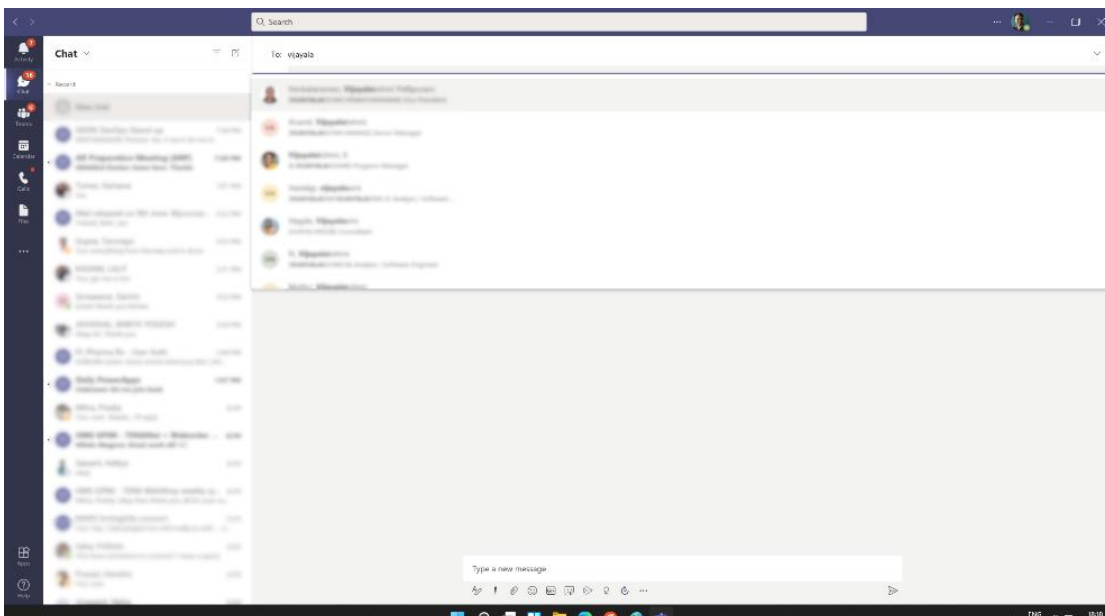


Figure 9: Teams running on WSLg



# 7.1 Productivity Apps

You can easily run **Microsoft Teams** with full voice features. Screen Sharing was quite usable. On WebApps side we installed **Microsoft Edge (Beta) for Linux** and it ran flawlessly.

**SharePoint** and other enterprise portal opened without any issues and we did not get any issues with either session management or SSO logging.

**Google Chrome for Linux** and native Linux tools work flawlessly with all the bling & chimes associated with them. Graphics intensive applications like **GIMP** works well and near-native experience can be achieved. (there were some issues in child-windows and their alignment; but noting too serious)

# 7.2 Solution Development

From development perspective Visual Studio Code & Eclipse ran flawlessly. It was a bit odd to get into the grips that these apps were not the part of the Windows environment and file system being used is the host Linux system (in our case Ubuntu). Apart from that the development environment was setup like any other Linux box. We did face a minor issue with a pop-up windows during installation; but a blind press of 'Enter' key got rid of it.

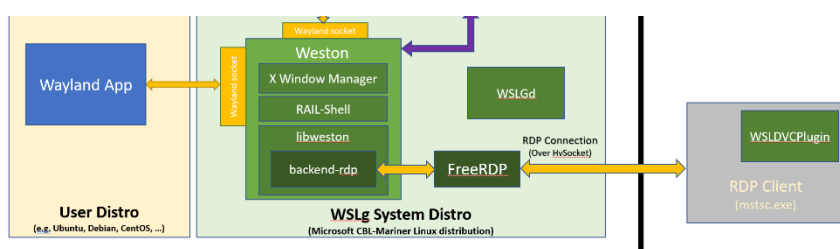
We created and executed some java apps using eclipsed and the debugging experience was completely compatible with any Linux environment. Debugging & overall performance of the IDE was quite compatible to any Linux installation.

We also installed Visual Studio Code and ran a small Node.js app. The installation was flawless and installing and setting up node environment was as fast as regular Linux installation. Both debugging and execution experience again was extrmly pleasant and we were running a node server in no time. The node server (ours was running on <http://127.0.0.1:3000>) was acceptable using the Chrome Browser installed in WSLg (Linux version) as well as was accessible from Microsoft Edge from the parent Windows 11.

**Important Note:**

*The Ctrl+C & Ctrl+V feature does not always work even when copying from a WSLg app to another WSLg app. We had some instance in which code copied from Edge Browser in Windows 11 environment did not get pasted in either Eclipse or VS Code installed in WSLg eco-system*

Unfortunately, having said that, it's still not exactly a native experience. If you look closely at the architecture design we had discussed earlier



you will see that the windows opened on the client system are actually RDP client windows (mstsc.exe windows) hence you will see some minor lag especially during video calling & playback or in graphics intensive application

which has lot of moving UI components.

Similarly, as all the WSL-g windows are basically instances of mstsc; we faced issues with docking and grouping windows on desktop especially for side-by-side docking. The windows tend to keep their dimensions and docking like left-edge docking or top-corner anchoring are not applied to these windows.





## 8 Usage in RPA

One of the most interesting use-case you might consider is to use Linux app on your Windows PC via. RPA to automate certain processes & business flows. This is a huge domain with multiple flavors and initial reports points that many RPA vendors are currently looking into this solution.

We did quick test using Power Automate Desktop installed on Window 11 and tried executing the Linux version of Google Chrome installed in WSLg. The RPA was able to execute the desktop icon and application opened and we were able to interact with some controls; but keystrokes were not being registered. This might be down to certain settings or an unbaked part of WSLg.

## 9 The final word

WSLg continued Microsoft recent strategy of adopting & implementing open standards and platforms. This marks the evolution of Microsoft ecosystem which seem to have a more in-dept grasp of & granular understanding of existing enterprise and their evolving IT ecosystem.

WSLg provides developers and even end-users to utilize tools they are comfortable with without worrying about the underlying ecosystem of Windows or Linux. This is especially helpful for users and processes who depend on user interfaces or need an easy to access sandbox for specific solutions.

WSLg is a boom to IT developers who are developing hybrid multi-platform solutions; they no longer need to depend on different PC's or VM's for development and testing and their whole end-to-end solution can be developed & tested using a single PC.

As with any new platform WSLg does have some rough edges to file-out (e.g. unreliable Copy/Paste feature) but looking at the rate of big fixes & features being implemented we can expect WSLg to be a mainstay of Power users & IT developer's workstations.



## About Capgemini

Capgemini is a global leader in partnering with companies to transform and manage their business by harnessing the power of technology. The Group is guided everyday by its purpose of unleashing human energy through technology for an inclusive and sustainable future. It is a responsible and diverse organization of 270,000 team members in nearly 50 countries. With its strong 50 year heritage and deep industry expertise, Capgemini is trusted by its clients to address the entire breadth of their business needs, from strategy and design to operations, fueled by the fast evolving and innovative world of cloud, data, AI, connectivity, software, digital engineering and platforms. The Group reported in 2020 global revenues of €16 billion.

Get the Future You Want | [www.capgemini.com](http://www.capgemini.com)



This document contains information that may be privileged or confidential and is the property of the Capgemini Group.

**Choose an item.** Copyright © 2021 Capgemini. All rights reserved.