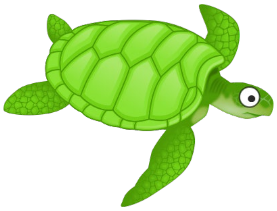
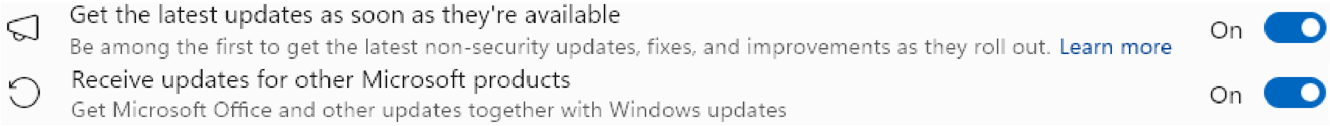
Congrats on installing ubDistBuild !



request: Configuration

Windows Update



Get the latest updates as soon as they're available
Be among the first to get the latest non-security updates, fixes, and improvements as they roll out. Learn more
On

Receive updates for other Microsoft products
Get Microsoft Office and other updates together with Windows updates
On

As a particular example, 'other' updates may include the MSW component WSL. Often necesssary to:
1) MSW Windows Update to update stock WSL Kernel .
2) Repeat   _install_wsl2   to cause successful 'wsl –update' .
3) Download rootfs -   _get_vmImg_ubDistBuild-rootfs internal   .
4) Install rootfs -   _install_vm-wsl2   .
5) Custom kernel for USB   _install_vm-wsl2-kernel   .

The installer 'ubDistBuild' is not yet able to manage the MSW Windows Update, nor is automatic download of tens of gigabytes from the internet a reasonable additional step to wait for from the installer. Moreover, some steps will become unnecessary as newer versions of WSL are shipped with MSW before update and mirror servers distribute verifiable VM/Live/ISO as a single file. A GUI for these manual post-installation steps and a website for distribution to end users are still high priorities.

endUserFunctions

For MSW users, most if not all of these are available as batch files. Especially, MSW users should download VM images, etc, with the '_get_vmImg_…' functions.

C:\core\infrastructure\ubDistBuild\

# Recommend 'internal' instead of 'latest' .

./_get_vmImg_ubDistBuild internal

./_get_vmImg_ubDistBuild-live internal

./_get_vmImg_ubDistBuild-rootfs internal

./_convert-vdi

./_convert-vmdk

./_convert-vhdx

# As Admin .
./_install_wsl2

# As Admin .
./_install_vm-wsl2
./_uninstall_vm-wsl2
./_install_vm-wsl2-kernel

# As Admin .
./_sshid-import-wsl2
./ sshid-export-wsl2

# Expand size of disk.

dd if=/dev/zero bs=1M count=384000 >> _local/vm.img

# Expand filesystem inside VM .

btrfs filesystem resize max /

Extra

As with ‘ubcp’, ‘ubdist’ when used with WSL includes YubiKey/USB compatibility for SSH out of the box. Please use the GUI to attach/detach USB devices.

USB mass storage (eg. Magneto Optical disc drives) and USB serial devices (eg. Arduino), is also usable out of the box.

Special

multi-WAN

# Discouraged , very strongly . May be less secure, may be less reliable, and may cause issues to other users of the relevant networks and servers. Normally otherwise not much different than curl . The functionality is intended possibly for extreme load-balanced multi-WAN (ie. more than 3 WANs) . The code, particularly the separate authentication step, was created more as a template towards possible use with any less trusted mirror or protocol .

# Degrades network traffic collision backoff algorithms (ie. this may be very not nice for other users of the ISP or servers affected) .

# Best done from recent Linux distribution with recent version of Network Manager supporting ‘bonding’ .

# Backend may be ‘aria2’ instead of ‘axel’ , and may download two files simultaneously at half as many connections each.

```
export FORCE_AXEL=12
```

```
./_get_vmImg_ubDistBuild internal #hash-whirlpool
```

```
./_get_vmImg_ubDistBuild-live internal #hash-whirlpool
```

```
./_get_vmImg_ubDistBuild-rootfs internal #hash-whirlpool
```

Download to Device

# Downloads using curl (FORCE_AXEL must be empty or unset), pipes to dd or wodim (ie. cdrecord), writing the ubdist dist/OS directly to a disk or disc .

```
./_get_vmImg_ubDistBuild internal "" /dev/sda
```

```
./_get_vmImg_ubDistBuild-live internal "" /dev/sr1
```

Join

Files downloaded individually (eg. ‘ package_image.tar.flx.part* ’ , ‘ package_rootfs.tar.flx.part* ’ , ‘ vm-live.iso.part* ’) may be joined . Datacenter quality internet connections do NOT require this, with extensive routine testing both with datacenter servers and GitHub Actions never having been unable to use normal scripted downloading, which is also much more memory efficient.

```
./_join.bat
```

Windows11 on ARM64 Compatibility

Proper configuration of WSL on ARM64 is provided by this installer. This functionality will be maintained due to commonality with the more often used x64 WSL installation scripting.

A native ARM64 build of ubdist for use with ARM64 WSL and/or Hyper-V may also be available.

./_get_vmImg_ubDistBuild-arm64 internal #hash-whirlpool

./_get_vmImg_ubDistBuild-live-arm64 internal #hash-whirlpool

./_get_vmImg_ubDistBuild-rootfs-arm64 internal #hash-whirlpool

./_join.bat

Computers comparable to these tested systems are specifically recommended:

*) Microsoft Surface Pro 2-in-1 Laptop/Tablet (2024), Windows 11 Copilot+ PC, 13" Touchscreen OLED Display, Snapdragon X Elite (12 Core), 16GB RAM, 256GB Storage

Only limited functionality is definitely expected to work under ARM64 . Proper configuration of WSL on ARM64 is still a particularly a high-priority feature.

Development of more native applications, hardware implementations of abstraction layers, etc, is a substantial goal of the ubDistBuild project, and developers usually benefit already from access to x64 VR capable PC workstations.

Performance

Change “Power mode” to “Best Performance”. The “Recommend” setting is usable, although slower and less consistent in some situations. ARM64 tablets (ie. Microsoft Surface tablets are remarkably instantaneous to Suspend/Resume), so there is often less of a need for absolute power efficiency.

Exceptions

*) USB IPD

*) WSL USB (GUI)

*) QEMU

*) dotNet

When bundled installers for these programs prompt to install under a Windows11 on ARM64 system (or comparable), simply click through cancelling their installers. Continue the ubDistBuild installation.

Further

Native ARM64 build of ubdist could be developed using a WSL Ubuntu or similar Linux distribution under ARM64. Many, of the instructions used to build Debian packages, install bootloader, etc, would remain applicable as-is, though an ARM64 kernel, both for normal full kernel and for WSL kernel building, would definitely be necessary.

Such an approach should also work well with the ecosystem of VSCode, GitHub CoPilot, triple 4k monitors, etc, through recent Microsoft Surface tablets.

Reference

https://learn.microsoft.com/en-us/windows/wsl/connect-usb

‘Arm64’ ‘not supported’ ‘usbipd-win’

Derivatives

Specialized derivative dist/OS is possible and may be available for small footprint (ie. no GUI for reduced data size) or very unusual features (eg. proprietary software).

Specialization incurs technology debt however, so use of the more standard ‘ubdist’ is strongly encouraged. Especially, ‘ubdist’ itself is ‘self-hosting’, built entirely within CI, thus is kept well maintained.

Compression (fast) already keeps the footprint very small for a dist/OS, and feature requests are very welcome.

Please consider reporting possible improvements to the standard ‘ubdist’ dist/OS, before considering a more specialized dist/OS.