Building Mutter Using GN for Testing Wayland

This document tracks all the work needed to build mutter as a third_party project within chromium tree, in order to be able to run the interactive UI tests on it using chromium CI infrastructure.

PoC CLs:

- https://chromium-review.googlesource.com/c/chromium/src/+/6104176
- https://chromium-review.googlesource.com/c/chromium/src/+/6180004

Generic requirements

- Generating configs.h using meson from a script: generate_configs.py
  - Generated once and checked in. Needs to be done again when rolling newer mutter.
  - Note the `DEFAULT_BUILD_ARGS` was used to disable a lot of unwanted features and simplify the build as much as possible.
- See `common_config` in BUILD.gn
- Glib version: Recent mutter uses newer glib than the one we use in chromium sysroot and so a lot of the code uses newer APIs:
  - Versions:
    - Ubuntu 22.04 version: 2.72.4
    - sysroot (debian bullseye) version: 2.66.8
    - Mutter req: 2.81.1
  - If sysroot glib package is to be used, some references can be fixed by updating the max version to 2.64: build/config/linux/BUILD.gn
  - If using sysroot glib package or Ubuntu 22 host package, backports can be added for some new APIs that are not available in sysroot glib. See glib_backports.h.
  - Some other issues can be resolved by making changes in mutter, e.g. missing `g_timeout_add_once` API was worked around by reverting the change that uses that API.
  - Ideally we may want to use the latest glib to be able to maintain it more easily in future and avoid running into potential side-effects, either by using a newer sysroot when building mutter, or by creating glib as a third_party repo.

Module-specific requirements

Mtk

- See `mtk` in BUILD.gn

Cogl

- Enums need to be generated using `glib-mkenums` tool: generate_cogl_enum_types.py

- See `cogl` in BUILD.gn

Clutter

- Enums need to be generated using `glib-mkenums` tool: generate_clutter_enum_types.py
    - Generated once and checked in. Needs to be done again when rolling newer mutter.
- Marshallers need to be generated using `glib-genmarshal` tool: generate_clutter_marshallers.py
    - Generated once and checked in. Needs to be done again when rolling newer mutter.
- Packages not in sysroot
    - gsettings-desktop-schemas package: Need to just include this package in our sysroot script as it is available in debian bullseye. Currently worked around by checking in the gdesktop-enums.h which is the only thing the package contains.
- See `clutter` in BUILD.gn

Libmutter

- Enums need to be generated using `glib-mkenums` tool: generate_meta_enum_types.py
    - Generated once and checked in. Needs to be done again when rolling newer mutter.
- Wayland protocol boilerplate needs to be generated using wayland-scanner: We could use our existing `wayland_protocol` GN template.
- DBUS boilerplate needs to be generated using `gdbus-codegen` tool.
    - Generated once and checked in. Needs to be done again when rolling newer mutter.
- Packages not in sysroot:
    - libdisplay-info package: This is needed for EDIDs.
        - Currently not available for debian bullseye or Ubuntu 22.04. So we may need to build this as a third_party project or use newer sysroot for building mutter. If we need to build it then we also need hwdata.
            - hwdata package: Available in Ubuntu 22.04 and in debian bullseye, but not in sysroot
    - liblcms2: Available in bullseye but not in sysroot.
    - libcolord-dev: Available in bullseye but not in sysroot (although libcolord2 exists).
- Packages in sysroot that are too old:
    - Gbm
        - Versions:
            - mutter needs: >= 21.3
                - Note: Version 21.3 is only needed because of `gbm_bo_create_with_modifiers2()`. See the version bump change for reference. So it may be okay

to use gbm version >= 17.3 and < 21.3 with a backport of that function.
        - Note with chromium's minigbm it builds successfully but there are odd issues seen at run time.
    - sysroot version: 20.3.5
    - Ubuntu 22.04 version: 23.2.1
- Libgudev
    - versions:
        - Mutter needs: >= 238
        - Sysroot version: 234
        - Ubuntu 22.04 version: 237
    - Only thing missing in the Ubuntu 22 version is `g_udev_device_get_current_tags`. Could either use a backport or maybe simple enough to build as a third_party repo
- libinput-dev
    - Versions:
        - Mutter needs >= 1.26.0
        - Sysroot version: 1.16.4
        - Ubuntu 22.04 version: 1.20.0
    - On version 1.20.0 or higher it can be by commenting out usage of libinput_tablet_tool_config_pressure_range_set
    - Depends on mtdev, which needs to be added to sysroot
- Gvdb subproject: This is needed by xdg-session-management implementation in mutter.
    - Added in DEPS and built using GN.
- Need to specify a folder path for plugins, locales, libexec and pkgdata: It's specified using macros in config.h which is checked in currently, but we may need a script to update it with those macros specified correctly at build time to point to the absolute path of the out folder on each machine.
- See `libmutter` in BUILD.gn

Default plugin

- TBD

Mutter

- See `mutter` in BUILD.gn

Build Options

Option 1: (In progress) Sysroot Build

https://chromium-review.googlesource.com/c/chromium/src/+/6180004

This is the standard option to build third_party repos in chromium, with the default `use_sysroot = true` GN arg. It is the ideal solution, but will take the longest as it will include building all mutter dependencies, including transitive dependencies, as third party

repos essentially as most of the packages in the debian oldstable sysroot are too old, and requires converting all of them to GN based builds. See productization tasks section below for this option.

Third-party repos with GN configs

- mutter
- gvdb
- libdisplay-info
- glib
- libgudev
- libinput

Option 2: (Not recommended) Non-sysroot build against host libs

https://chromium-review.googlesource.com/c/chromium/src/+/6104176

The idea with this option is to primarily support Ubuntu 22.04, which is what the chromium test bots use, using host libs by setting `use_sysroot = false`, with possible limited support on newer distributions but not on older distributions.
One benefit with this option over using sysroot is that more package version requirements are met as Ubuntu 22.04 is newer than the debian old stable based sysroot, hence requiring creating fewer third_party repos.
Since the host library versions are older than what latest mutter uses, this option requires some workarounds and backports, primarily for glib.

Build Setup

Host: Ubuntu 22.04 Docker Container
GN args: `use_sysroot = false use_siso = false`

Installed host packages: `sudo apt install libgles2 libgraphene-1.0-0 libcolord2 libxkbcommon0 libinput10 libinput-dev qtbase5-dev libcolord-dev liblcms2-dev libgdk-pixbuf-2.0-dev xcvt`

Workarounds and backports

- chromium_workarounds.h
- atk_autocleanups.h
- glib_backports.h

Third-party repos with GN configs

- mutter
- gvdb
- libdisplay-info

Results

Succeeded in building most of libmutter, with ~10 compile errors left which just need the following glib backports or modifying mutter, some of which are non-trivial however:

- glib_autoptr_cleanup_GUnixFDList

- GUnixFDList_autoptr
- g_unix_fd_list_new
- g_unix_fd_list_append
- g_udev_device_get_current_tags
- g_timeout_add_seconds_once
- g_ptr_array_new_null_terminated
- GPtrArray_autoptr

Based on these results, this could be an initial option, albeit it's hacky.

Option 3: (Possible future alternative) Using a container

If chromium build infrastructure supported containers, we could for instance use a recent fedora, debian testing or arch linux container to build mutter using meson instead of GN and run mutter within the container as well. Thus we wouldn't need to maintain our third party repos or GN build configurations.

Test Results

The following environment was tested for building forked mutter with test protocol and interactive_ui_tests was run within it:

Fedora container: registry.fedoraproject.org/fedora-toolbox:41
Installed build dependencies: `sudo dnf builddep mutter`
Installed runtime dependencies: `sudo dnf install dbus-daemon mutter-common`

Productization Tasks

- ☑ Build mutter with dependencies
  - Option 1: sysroot build
    - ☑ Create third_party repos and GN configs for mutter dependencies (because sysroot packages for them are too old):
      - ☑ gvdb
      - ☑ libdisplay-info
      - ☑ glib
        - ☑ Create new `use_system_glib` GN arg and make `//build/config/linux:glib` conditional based on that.
        - ☑ glib-2.0
        - ☑ gobject-2.0
        - ☑ gmodule-2.0
        - ☑ gio-2.0
      - ☑ libgudev
      - ☑ libinput
      - ☑ ~~gbm~~ (use sysroot package with workarounds)

        - [x] ~~Additionally, if needed, create the following as third_party repos and GN configs (because they depend on glib, even though sysroot packages are not too old):~~
            - [x] ~~graphene~~
            - [x] ~~atk~~
            - [x] ~~cairo~~
            - [x] ~~pangocairo~~
            - [x] ~~colord~~
            - [x] ~~gdk-pixbuf~~
        - [x] Request the following packages to be added to sysroot (available in debian bullseye):
            - [x] ~~liblcms2-dev~~
            - [x] ~~libcolord-dev~~
            - [x] ~~libmtdev-dev~~
        - [x] Use upstream mutter instead of downstream
        - [x] Remove chromium workarounds
        - [x] Undo changes in build/config/linux/BUILD.gn for glib max allowed version
        - [x] Build mutter using GN
            - [x] ~~mtk~~
            - [x] ~~cogl~~
            - [x] ~~clutter~~
            - [x] ~~libmutter~~
            - [x] ~~Mutter~~
    - Option 2: Non-sysroot build against host libs on Ubuntu 22.04
        - [x] Create third_party repos and GN configs for mutter dependencies:
            - [x] ~~gvdb~~
            - [x] ~~libdisplay-info~~
        - [ ] Complete and optimize chromium workarounds and glib backports
        - [ ] Build mutter using GN
            - [x] ~~mtk~~
            - [x] ~~cogl~~
            - [x] ~~clutter~~
            - [ ] libmutter (mostly done)
            - [ ] mutter
- [x] Debug run-time issues by testing on the following
    - Ubuntu 22.04
        - [x] Needs these workarounds: https://gitlab.gnome.org/orko/mutter/-/commit/cd209db9e5d861759fd605ce304d668d465be974
    - Ubuntu 24.04
    - Arch
- [x] Request mirrors
- [x] Optimize scripts

    - Instead of running every script one by one we could run them from a single python or bash script
    - We could use meson compile and copy things over
    - ~~gen dbus sources script (if not using meson compile)~~
- ☑ Ensure all targets are proper type (source_set, static_library etc.)
- ☑ plugin and locale dir
- ☑ Finish other TODOs
- ☑ ~~Upload commits for the minimal mutter changes needed for it to run properly in Ubuntu 22.04 to workaround gsettings schema issues~~ (We can use Ubuntu 24 which doesn't need any mutter modifications)
- ☑ Create split CLs
    - Finish README.chromium, Add OWNERS and LICENSE as applicable
- ☑ Determine build time and runtime package deps on Ubuntu 24.04 if any other than what's available from chromium install deps script.
    - `sudo apt-get install file lsb-release`
    - `sudo build/install-build-deps.sh --no-arm`
    - `sudo apt install libgraphene-1.0-0 mutter-common`
- ☐ Nice-to-haves (Possibly create separate issues for them)
    - Investigate possibility to use gbm in third party and/or eglInitialize issue with latest minigbm
    - Mutter CL improvements
        - ☐ Put sources in separate .gni files
        - ☐ Create crbugs for TODOs
    - Copy missing gsettings schemas in Ubuntu 22.04 and pick those up from the config instead of modifying mutter
    - Create a doc file in //docs

Future Considerations

- Using cipd to host prebuilt binaries of mutter and/or other dependencies: The packages could be built in a controlled environment where it is easier to build using meson and the binaries could be uploaded to gcloud. This could come in handy if building and maintaining all the dependencies becomes difficult in the long run.
- Updating the bots for mutter interactive UI tests to use newer distros like Ubuntu 24 or other.

Tricks

- Introspecting meson targets after running `meson setup <build dir>`:
  `meson introspect --targets <build dir> | jq | less`
- Debugging a package in chromium sysroot:
  `./build/config/linux/pkg-config.py -s ./build/linux/debian_bullseye_amd64-sysroot/ -a amd64 -d <package> | jq`

- Querying a package version in chromium sysroot:
  `./build/config/linux/pkg-config.py -s ./build/linux/debian_bullseye_amd64-sysroot/ -a amd64 --version-as-components <package>`