Intel Graphics Driver Version

Author(s): Yang Gu (yang.gu@intel.com)

Public Document

- Overview
- Windows
  - Windows D3D
  - Windows Vulkan
- Linux and ChromeOS
  - Linux OpenGL
  - Linux Vulkan
- Browser Implementation
- Related Resources
- Document History

Overview

This document is to describe Intel Graphics Driver version schema and how to get them.

Windows

On Windows, D3D driver and Vulkan driver release together, so they share the same version number.

Windows D3D

Some details can be found at https://www.intel.com/content/www/us/en/support/articles/000005654/graphics-drivers.html.

[Microsoft Schema]
AA.BB.CCCCC.DDDDD (mandatory for inf/sys/dll)
AA: up to WDDM version (AA >= WDDM)
BB: (WDDM >= 1.2) = D3D feature level ; (WDDM <= 1.1) = DDI version
CCCCC: Number up to 5 digits from 0 to 65535
DDDDD: Number up to 5 digits from 0 to 65535

[Intel Schema]

Series	**Intel Release Version**	**Driver Version (inf/sys/dll)**	**Comment**
new	**AA.BB.CCC.DDDD (current)**	**AA.BB.CCC.DDDD**	CCC is 100 now (20191206), but

	YYMM.CCC.DDDD (future) AA = up to WDDM version BB = D3D feature level YY = Year MM = Month CCC.DDDD = Build number	AA = up to WDDM version BB = D3D feature level CCC.DDDD = Build number BB is set to 0 with the driver naming convention introduced by Microsoft with Windows® 10 May 2021 Update (21H2) and later.	it may increase if DDDD overflows. For example, the next version (CCC.DDDD) of 100.9999 is 101.0000.
Legacy	**15.XX.YY.DDDD** 15.XX = Baseline YY = Release revision DDDD = Build number	**AA.BB.CC.DDDD** AA = up to WDDM version BB = D3D feature level CC = Internal value to meet Microsoft's version schema requirements (See table below for the detailed mapping between YY and CC) DDDD = Build number Actually different release version has different codebase, we should take YYDDDD as build number.	In the foreseeable future (like 20 years), DDDD will never overflow. intel graphics driver version: 15.45.29.5077 (Windows Driver Store Version 21.20.16.5077)

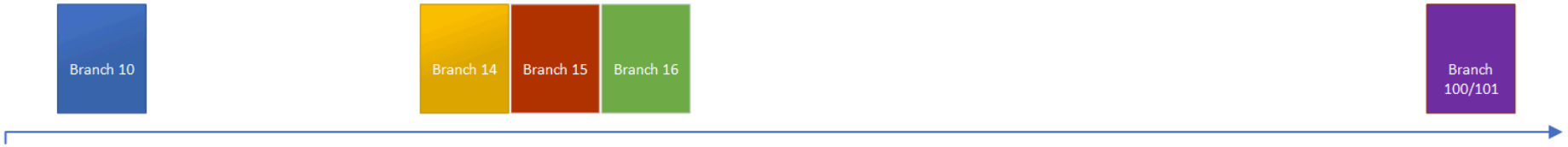
Legacy Intel Release Version	Driver Version
15.**22**.xx.xxxx 15.**28**.xx.xxxx 15.**33**.xx.xxxx	xx.xx.**10**.xxxx
15.**36**.xx.xxxx 15.**38**.xx.xxxx	xx.xx.**14**.xxxx
15.**40**.xx.xxxx	xx.xx.**15**.xxxx
15.**45**.xx.xxxx 15.**46**.xx.xxxx 15.**47**.xx.xxxx 15.**49**.xx.xxxx 15.**60**.xx.xxxx 15.**65**.xx.xxxx	xx.xx.**16**.xxxx

[Tell among Driver branches]

If the third component is CC, which means it's smaller than 100, it's the legacy driver. If the third component is CCC, which means it's equal to or larger than 100, it's the new driver.The first known new driver version is 24.20.100.6025.

We just have one branch for the new driver (100 and 101 are the same branch). But for legacy driver, actually we have many different branches, differentiated by Intel release versions. According to above table, we have at least 12 branches for legacy drivers. Different branches

should be considered as totally different code base. That's to say, one fix in a branch may not arrive in another branch. Different branches for legacy driver share the 4th field, which will increase whenever a branch has a new release.
We should always take the combination of last 2 components as build number, that is CCCDDDD for the new driver and CCDDDD for the legacy driver. As different branches may map to same 3rd field (e.g. both 15.45 and 15.46 map to xx.xx.16.xxxx), there is no way to differentiate them just based on driver version. This can be problematic when implementing some code logic (like workaround mechanism in application), but there is no better way given all the info we can have is the driver version.



[Get Driver Version (Microsoft Schema)]
IDXGIAdapter::CheckInterfaceSupport
HRESULT CheckInterfaceSupport(
  REFGUID   InterfaceName,
  LARGE_INTEGER *pUMDVersion
);
AA.BB.CCCCC.DDDDD = "%d.%d.%d.%d" % (HIWORD(umd_version.HighPart), LOWORD(umd_version.HighPart), HIWORD(umd_version.LowPart), LOWORD(umd_version.LowPart))

[Example]
For CCC >= 100, it's possible to have AA.BB.100.6025 and AA.BB.101.6025 in future.
Within each branch, the build number will increase for each release. For example, it will never happen that we have both 21.20.16.4839 and 22.20.16.4839, even if they really have same code base.

We can just compare the build number of 2 releases, and know if the driver was released earlier or later than the other.

Windows Vulkan

Vulkan spec says the driver version is implementation-dependent. Windows Vulkan driver releases together with D3D driver, so they share the same version. However, due to Vulkan spec definition (as below), only CCC.DDDD is encoded in 32-bit driverVersion.

```
typedef struct VkPhysicalDeviceProperties {
    uint32_t                            apiVersion;
    uint32_t                            driverVersion;
    uint32_t                            vendorID;
    uint32_t                            deviceID;
    VkPhysicalDeviceType                deviceType;
    char                                deviceName[VK_MAX_PHYSICAL_DEVICE_NAME_SIZE];
    uint8_t                             pipelineCacheUUID[VK_UUID_SIZE];
    VkPhysicalDeviceLimits              limits;
    VkPhysicalDeviceSparseProperties    sparseProperties;
} VkPhysicalDeviceProperties;
```

[Get Driver Version (Vulkan Schema)]
CCC.DDDD = driverVersion >> 14 + “.” + driverVersion & 0x3FFF

Linux and ChromeOS

On Linux and ChromeOS, Mesa includes both OpenGL (ES) driver and Vulkan driver, so they share same version. The schema of driver version is “MAJOR.MINOR[.PATCH]”. Note that from year 2017, Mesa began to use year for major version, so MAJOR jumped from 13 to 17.

Linux OpenGL

[Get Driver Version]
VERSION_STRING = glGetString(GL_VERSION)
Then you need to pass the VERSION_STRING to get relevant string.

Linux Vulkan

[Get Driver Version (Vulkan Schema)]
MAJOR.MINOR.PATCH = VK_VERSION_MAJOR(driverVersion) + "." + VK_VERSION_MINOR(driverVersion) + "." + VK_VERSION_PATCH(driverVersion)

Browser Implementation

Chromium: gpu/config/gpu_control_list.cc

Dawn: src/common/GPUInfo.cpp
ANGLE: src/libANGLE/renderer/driver_utils.cpp

Related Resources

Understanding the Intel® Graphics Driver Version Number
Intel Graphics Driver History
Intel GPU Codename

Document History

Date	**Version**
20210630	Intel driver is going to support AA.BB.101.DDDD. Add browser implementation.
20191206	First version