GStreamer on Windows: Everything New

Nirbheek Chauhan

(*nirbheek* on Twitter/IRC)

https://nirbheek.in

gstreamer

Centricular

Overview

History Lesson

1.14 release

1.16 release

1.18 plans

History Lesson

History Lesson

Everything sucked

History Lesson

Everything sucked

Release-driven testing on Windows

Development was stagnant

1.14 release

- WASAPI (Windows Audio Session API) plugin was rewritten

  http://blog.nirbheek.in/2018/03/low-latency-audio-on-windows-with.html

- `GstDeviceProvider` support in WASAPI and DirectSound plugins

1.16 release

Too much to fit on one slide.

1.16 release: GStreamer

- Meson support is feature-complete on all platforms
- WASAPI plugin gained support for loopback recording
- WASAPI plugin rank increased, now the default
- NVIDIA hardware accelerated codec plugins build on Windows
- Intel Media SDK hardware accelerated codec plugin builds on Windows (need MSVC)
- Misc bugfixes and perf improvements to `wasapisrc`, `wasapisink`, `ksvideosrc`, `dshowsrc`, `GstGL`, etc
- New project `gst-build` created, uses Meson subprojects and replaces `gst-uninstalled.sh`

  https://gitlab.freedesktop.org/gstreamer/gst-build/

1.16 release: Cerbero

- Cerbero ported to Python 3
- Meson support merged, and Cerbero requires it for all targets
- `visualstudio` variant, to build with Microsoft's Visual C++ Compiler (2015 and 2017)
- `gettext.recipe` (+ deps) replaced with `proxy-libintl.recipe`
- `intelmsdk` and `nvcodec` variants to build NVIDIA and Intel plugins
- `qt5` variant to build the Qt5 QML/GL video sink plugin
- New `-v` command-line argument that takes a comma-separated list of variants
- `webrtcdsp` plugin re-enabled, upstream library ported to Meson

1.16 release: Cerbero #2

- Checksum support in Cerbero, fixes security hole on Windows since we can't do certificate checking
- Use Python 3's `urllib` module to download instead of MSYS's ancient `wget`
- Use Python 3's `tarfile` module to extract tarballs instead of MSYS's ancient `tar`
- `fetch-bootstrap` command + global source cache makes (re)bootstrap faster everywhere
- Improved installation and usage instructions
  - Convert to Markdown
  - Add screenshots for installers (Python, Git, MSYS, etc)

1.16 release: Cerbero #3

- New installers for binaries built with MSVC!
- New download page written in markdown!

  https://gstreamer.freedesktop.org/download/#windows
- Toolchain compatibility notes!

  https://gstreamer.freedesktop.org/download/#toolchain-compatibility-notes

GStreamer version	MinGW	MSVC
1.14.x	msvcrt.dll	N/A
1.16.x	msvcrt.dll	ucrtbase.dll
1.17.x (development)	ucrtbase.dll	ucrtbase.dll

1.16: Toolchain Notes

App Toolchain	1.14 MinGW	1.16 MinGW	1.16 MSVC
Visual Studio 2015 and newer	PARTIAL	PARTIAL	FULL
Visual Studio 2013 and older	PARTIAL	PARTIAL	PARTIAL
MinGW	FULL	FULL	PARTIAL
MinGW-w64	FULL	FULL	PARTIAL
MSYS2 MinGW-w64	FULL	FULL	PARTIAL
Cygwin	NONE	NONE	NONE

1.16.1 release: bugfixes

- `d3dvideosink` crash on `WinProc` handler
- `wasapisrc` bug that was causing samples to be dropped due to a typo
- Workarounds for two cross-CRT `free()` crashes
- Visual Studio 2019 and Visual Studio 2019 Build Tools support
- Improved support non-English locales; tell MSVC that all our sources are UTF-8
- Bugfixes to x264 and FFmpeg `gst-build` subprojects
- More coming in 1.16.2!

1.18: merged, GStreamer

- `gst-build` now has native Windows CI for MinGW and Visual Studio
- `nvdec` and `nvenc` plugins merged into one, and dynamically load NVIDIA DLLs; can probably ship by default now that they have no build-time dependencies on proprietary headers.
- New `d3d11` video sink plugin, old sink used Direct3D 9, which is deprecated
- New WinRT/ANGLE `GstGLWindow` implementation for use with Universal Windows Platform apps
- More that I have missed!

1.18: merged, Cerbero

- New MinGW/GCC toolchain, upgraded from 4.7.3 to 6.0.0
- MinGW now uses `ucrtbase.dll` instead of `msvcrt.dll`, eliminating Cross-CRT issues
- Parallel fetch of recipes during `fetch-package` and `fetch-bootstrap`
- Parallel running of the configure and compile stages of all recipes, should speed-up building of Autotools recipes a *lot* on Windows since we build with `make -j1`

1.18: Toolchain Notes

App Toolchain	1.14 MinGW	1.16 MinGW	1.16 MSVC	1.18 ALL
Visual Studio 2015+	PARTIAL	PARTIAL	FULL	FULL
Visual Studio 2013 and older	PARTIAL	PARTIAL	PARTIAL	PARTIAL
MinGW	FULL	FULL	PARTIAL	PARTIAL
MinGW-w64	FULL	FULL	PARTIAL	FULL
MSYS2 MinGW-w64	FULL	FULL	PARTIAL	UNKNOWN
Cygwin	NONE	NONE	NONE	NONE

1.18: planned, Cerbero

- Universal Windows Platform support, come see my talk tomorrow!
- New default prefixes for toolchain variants (`visualstudio`, `uwp`, `debug`) inside `C:/gstreamer/1.0`
- Option for an Autotools-free build, first step for the CI
- Revamping of the MSI installers
    - Allow in-place upgrades
    - Bundle sources for better debugging
    - New env vars for the new prefixes
    - Ship more optional plugins, `nvcodec`, `qt5`, etc
    - Does anyone use merge-modules?
- Fridge functionality for build-tools, then all external dependencies
- Native Windows CI, lots of work needed!

Credits

Aaron Boxer
Andoni Morales Alastruey
Christoph Reiter
Jordan Petridis
Joshua M. Doe
Mathieu Duponchelle
Matthew Waters
Nicolas Dufresne
Pablo Marcos Oltra
Sebastian Dröge
Seungha Yang
Thibault Saunier
Tim-Philipp Müller
Xavier Claessens

... and more, including testers that report issues!