

Understanding WebKit/GTK+

Looking at the internals
COSCUP 2010

Shameless Plug

- I am working on WebKit since December 2006
- I have worked on WebKit/GTK+ and QtWebKit
- I provide teaching/consultancy

The Goal Of This Talk

- Take the fear from working on a big project
- Show ways to approach the codebase

Our Agenda

- What is the WebKit project?
- What is JavaScriptCore?
- What is WebCore?
- What is the WebKit API?
- Diving into the WebKitWebView
- How to use gdb to explore a large codebase?

What Is The WebKit Project?

- ~77 Reviewers
- ~100 Committers
- 8 ports in the tree
- Apple and Google are main contributors
- Many commits a day

What Is The WebKit Project?

- A platform/port makes WebKit run on a specific system
- A platform has a lot of freedom on how to do things.
- This allows the best possible integration.
- E.g. WebKit/GTK+ uses GTK+, Cairo, ATK, Soup, Empathy, ICU
- We use IRC, Mailinglists, Bugzilla for the project

What Is The WebKit Project?

- JavaScriptCore/ for JavaScript, some template magic
- WebCore/* for DOM,HTML,SVG...
- WebKit/* for the stable APIs
- LayoutTests/* for regression tests
- WebKitTools/* scripts, tools...

What Is JavaScriptCore?

- JavaScriptCore/wtf - a Web Template Framework
- JavaScriptCore/interpreter - The interpreter
- JavaScriptCore/jit - The jit
- JavaScriptCore is at the bottom of the stack.

What Is WebCore?

- C++ code that handles everything...
- HTML parsing, DOM, SVG, Editing, Layout...
- The code makes use of the porting layer
- WebCore/platform WebCore/platform/graphics

What Is WebKit/*?

- Directory for the port/platform API
- One directory for API, one for WebCore integration
- WebCoreSupport/* contains clients to interact with WebCore

Diving Into The WebKitWebView

- WebKit/gtk/webkit/webkitwebview.cpp
- A WebView is a GtkWidget
- It holds one WebKitWebFrame, settings, sends signals
- It wraps the WebCore::Page

Diving Into The WebKitWebView

- Expose handled by painting through WebCore::FrameView
- Input Events send to the EventHandler of WebCore::Page
- Let us see this with gdb

Diving Into The WebKitWebFrame

- WebKit/gtk/webkit/webkitwebframe.cpp
- A WebFrame is a GObject
- Wrap the WebCore::Frame
- Load content into a Frame
- Provide signals for events

Understanding The *Client Concept

- WebCore should not know types from WebKit/*
- But WebCore needs to interact with WebKit/*
- E.g. to open a new frame, ask for a file
- *Client is a pure virtual class (interface)
- It is used in WebCore code
- The implementation is in WebKit*/WebCoreSupport/*

Understanding The Clients

- `$ find WebCore -name *Client.h`
- ChromeClient to provide info to JS, show dialogs
- FrameLoaderClient to control loading
- DragClient to handle Drag and Drop
- EditorClient to handle IM
- ...

Looking Into FrameLoaderClient

- WebKit/gtk/WebCoreSupport/FrameLoaderClientGtk.cpp
- Sends signals about the load status
- Policy if a page should be opened

Looking Into ChromeClient

- WebKit/gtk/WebCoreSupport/ChromeClientGtk.cpp
- Provide information to JavaScript
- Run the prompt for the JavaScript

We Will Now Look Into The Porting Layer

- JavaScriptCore/wtf, WebCore/platform
- In general we have one common header file
- Implementation in port specific subdirectory.

What Is Handled By The Porting Layer?

- Unicode is handled by ICU most of the time
- Graphics can be done with Qt,SKIA,CoreGraphics,Cairo
- Networking abstraction
- We use notImplemented() to print debug messages

Looking At The Network Integration

- ResourceHandle and ResourceRequest
- ResourceRequest includes the URL, HTTP Headers and more
- ResourceHandle will handle the request
- ResourceHandle will call the ResourceHandleClient with the data
- Scheduling, Caching of requests is mostly done by WebCore

Networking On WebKit/GTK+

- WebKit/GTK+ is using soup for HTTP, GIO for others
- WebCore/platform/networking/soup/*
- ResourceHandle::start decides which backend to use

Graphics Abstraction

- GraphicsContext provides immediate painting interface
- Need to provide Bitmap/Pixmap/Image abstraction
- For WebKit/GTK+ it is realized with Cairo

Font Abstraction

- WebCore/platform/graphics/Font.h
- Is responsible for determining width of a string
- Has a fast path requiring the platform to have access to glyphs.
- For WebKit/GTK+ pango is used for complex text.

- Questions?
- 謝謝