Goodbye NPAPI (Chrome)

NPAPI (Netscape Plugin Application Programming Interface is a cross-platform plugin architecture used by many web browsers. It was first developed for Netscape browsers (starting with Netscape Navigator 2.0 in 1995) but was later adopted and implemented by many other browsers (e.g. Internet Explorer 3 in 1996).

For stability and security reasons, Google decided, in 2013, to start phasing out NPAPI support in Chrome. As of version 42, NPAPI is disabled by default and must be re-enabled explicitly. Google intends to remove support entirely in version 45, later in 2015. So Google Chrome will not support NPAPI plugins anymore, and will block plugins which use this technology. This includes Oracle's Java and Microsoft's Silverlight plugins (although these will be white listed for 5 months) which is a total buzz kill for me, because (even though I don't like it) I need Java and Silverlight. My banking software depends on Java and Outlook Web Access forks just so much better when Silverlight is present.

How does a plugin work?

A plugin declares that it handles certain content types. When the browser encounters that content type, it loads the associated plugin (that will runs inside the page), sets aside space within the browser context for the plugin to render and then streams data to it. The plugin is from now on responsible for rendering that data. Older browsers required the user to download the data and open an external application to handle unknown content types.

The API requires each plugin to implement and expose approximately 15 functions for initializing, creating, destroying and positioning plugin content. The NPAPI also supports scripting, printing, full screen plugins, windowless plugins and content streaming.

On August 12, 2009, a new project was introduced on Google Code (btw. goodbye to Google Code as well). Pepper, with the associated Pepper Plugin API (PPAPI). "A set of modifications to NPAPI to make plugins more portable and more secure". This extension is designed specifically to ease the implementation of out-of-process plugin execution. The goals of the project are to provide a framework for making plugins fully cross-platform. Topics considered include:

- Uniform semantics for NPAPI across browsers.
- Execution in a separate process from the renderer/browser.
- Standardize rendering using the browser's compositing process.
- Defining standardized events and 2D rasterization functions.
- Initial attempt at providing 3D graphics access.
- Plugin registry.

The following web browsers support PPAPI plugins:

- Google Chrome
- Chromium – open source browser was the development web browser that first utilized the new browser plug-in model.[15]
- Opera (since version 24)

Mozilla announced that it was "not interested in or working on Pepper at this time."

In February 2012, Adobe Systems announced that future GNU/Linux versions of Flash Player would only be provided via PPAPI, although the previous release, Flash Player 11.2, with NPAPI support, would receive security updates for five years.

So, I have no choice but to switch back to Firefox after using Chrome for years. I know that there is still a way to enable NPAPI in Chrome, and it will be completely removed with the version 45, but I doubt that Oracle will write a PPAPI Java plugin by that time, and I don’t want to be surprised twice. Yes, I forgot all about this and spent an hour reinstalling Java in Chrome and trying to figure out why don’t I see the plugin on the plugins list.

From:
https://wiki.plecko.hr/ - **Eureka Moment**

Permanent link:
https://wiki.plecko.hr/doku.php?id=general:npapi

Last update: **2019/10/31 08:55**