

XUL/XBL Replacement Roadmap

XUL is often mentioned as a [source of problems](#) for developers. Mozilla has been talking about “XUL Removal” in one form or another since at least 2007¹²³⁴⁵. Originally, XUL filled feature gaps in the Web platform needed for the Firefox UI, but the web has evolved enough that these gaps are largely closed today.

Defining Success

XUL slows down Gecko and Firefox development. Building Firefox with web standard technologies will let us move faster with the team we have and make it easier to recruit and onboard new contributors. It'll also allow us to better focus on performance and tooling for the web, since we'll only be supporting one platform for chrome and content.

Success won't be determined by removing any individual piece of code, although there will be technical milestones along the way as individual XUL features get removed. The project will be completed once:

1. Firefox will primarily use the Web platform for rendering web content in the browser chrome.
2. Mozilla is no longer thinking about “XUL Removal”.

Defining XUL

“XUL” is tricky to define (see [Attempting to Define XUL Removal](#)), but it can be broken down into 6 main categories. This is not an exhaustive list of XUL features, but the most important ones to address for this project. The costs and benefits of each of these technologies can be considered separately, and engineering work for each can proceed mostly independently.

1. XBL and the built-in binding implementations ([metabug](#), [tracking dashboard](#))
2. The XUL flex model ([metabug](#))
3. The XUL overlay system ([metabug](#), [design review](#))
4. L10n via DTDs ([metabug design doc](#), [prefs design review](#))
5. The XUL document ([metabug](#))

¹ Mozilla Developer Day in Paris in June 2007, where bsmedberg caused a shock in the room when referring to “XUL Dark Matter”: [bsmedberg's post](#) and a [reaction](#).

² Mitchell's posts from 2007 which reiterate support for XUL, but make it clear Firefox as an application is more important than XUL as a platform: [1](#), [2](#), [3](#)

³ 2010: [bug 546857](#): Drop support for XUL on web sites (remote XUL)

⁴ 2013: [@joewalker](#): From a Mozilla Summit talk on Servo - “There is no XUL in the product at all”. Applause from audience.

⁵ 2017: Mossop's [Deprecating XUL in new UI](#) mailing list post

6. The built-in XUL elements, including those with C++ implementations (`<tree>`, `<panel>`, `<menu>`, etc)

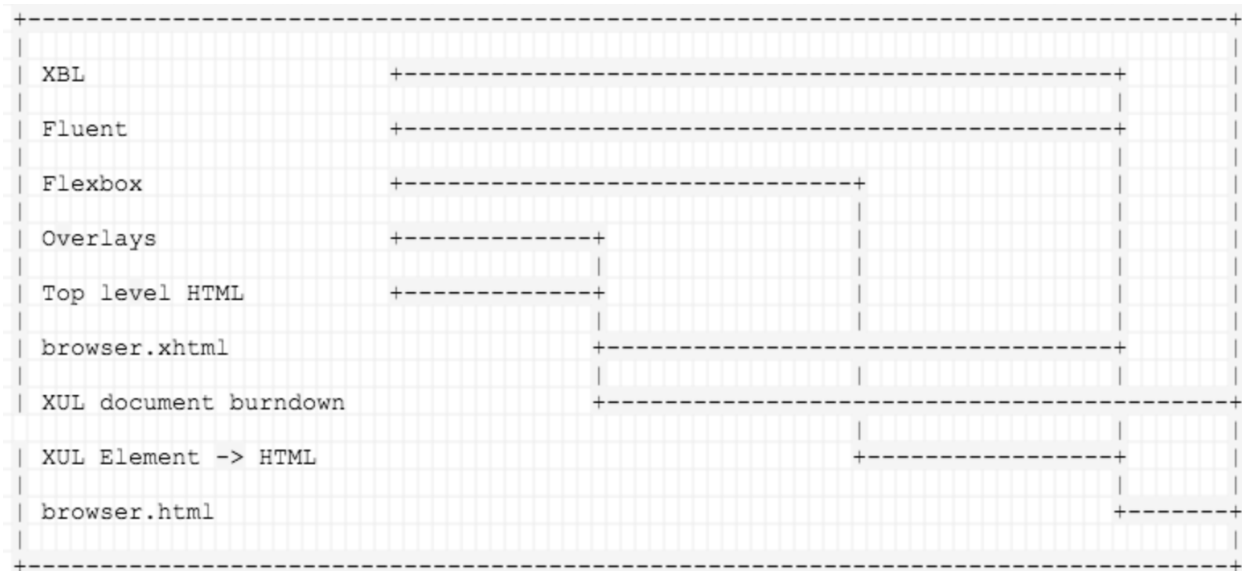
Strategy

There are separate engineering projects currently underway or completed for each for the 6 categories above. Work generally goes in one of two “directions”:

- “inside-out” refers to simplifying and migrating existing code to web technologies. The majority of the engineering work fits into this category, since we don’t think it’s feasible to rewrite Firefox from scratch. The incremental and relatively low-risk nature of this work tends to appeal to owners of the affected modules.
- “outside-in” refers to fully supporting windows that are already implemented in HTML. This allows us to identify issues that will come up as we get closer to supporting existing UI in HTML documents, and to get to the finish line with secondary UI more quickly.

We plan to “meet in the middle”, resulting in existing chrome documents like browser.xul being converted to HTML in the process. We expect to have an intermediate period as XHTML to support XUL markup and DTD files while we make progress on removing those features.

This work spans multiple projects and it can be hard to figure out what depends on what. Here’s a dependency graph (which is identifying dependencies and not timelines for individual projects):



Completed Milestones

Oct 2017 - XBL design review [is completed](#) with sign off from stakeholder engineers and management. [XBL In Firefox](#) post and [arewexblstill](#) dashboard are live.

Jan 2018 - Initial strings landed with Fluent. See [bug](#).

Jan 2018 - XUL (RDF) Templates [removed](#). Large amount of platform code removed (40K LOC), small impact on frontend since it was only lightly used. Took around 1 week of work.

March 2018 - Last overlay in Firefox frontend [removed](#). Took around 1 quarter of work to achieve, includes quite a number of simplifications with how resources get loaded in chrome.

April 2018 - ⅓ of XBL bindings removed - 200/300 XBL bindings remaining.

May 2018 - First Custom Element shipping in the browser chrome.

May 2018 - Size and location of top-level HTML windows are persisted.

July 2018 - Overlay platform implementation removed. ⅓ categories down . See [post](#).

July 2018 - HTML title tooltips work in chrome HTML.

July 2018 - Context menus can be opened from chrome HTML.

July 2018 - Browser Console no longer requires a XUL wrapper document. See [post](#).

Aug 2018 - Initial DevTools integration for Web Components. See [post](#).

Aug 2018 - Initial version of the browser window as XHTML landed as a build option. See [post](#).

Aug 2018 - `<xul:broadcaster>` and `<xul:observer>` removed from browser.xul. See [bug](#).

Aug 2018 - Custom Elements get loaded in HTML documents. See [bug](#).

Sept 2018 - Browser Toolbox no longer requires a XUL document. See [bug](#).

Sept 2018 - Fluent exposed by default for chrome privileged documents. See [bug](#).

Sept 2018 - “UA Widget” implementation landed as an alternative to in-content XBL. See [bug](#).

Oct 2018 - XUL commands exposed to top-level HTML docs. See [bug](#). Removing / modifying the XUL command implementation will be dealt with separately.

Nov 2018 - XUL persistence is exposed to XUL elements in HTML documents. See [bug](#). We will not remove the existing persistence feature from XUL docs.

Nov 2018 - No more consumers of XBL `<stylesheet>` ([metabug](#)).

Nov 2018 - Last in-content XBL binding removed ([metabug](#)).

Jan 2019 - XULDocument no longer exposes anything not on HTMLDocument ([metabug](#)).

Jan 2019 - In-content XBL platform implementation removed ([metabug](#), [mailing list](#)).

Jan 2019 - XBL `<stylesheet>` platform support removed ([metabug](#), [mailing list](#)).

Mar 2019 - Build a dashboard tracking the number of XUL documents in the tree ([dashboard](#)).

Mar 2019 - Chrome XHTML documents use prototype cache ([bug](#)).

May 2019 - “browser.xul” is shipped with XHTML ([metabug](#), [bug](#)).

May 2019 - “hiddenWindow.xul” (on OSX) is using XHTML ([bug](#)).

July 2019 - Old about:addons UI removed. External work, removes a bunch of XBL bindings.

July 2019 - XULDocument removed. All existing XUL documents are loaded as XHTML ([metabug](#)).

Q3 2019 - All XBL bindings removed ([metabug](#)).

Q3 2019 - GeckoView is no longer shipping XBL ([bug](#)).

Q4 2019 - XBL removed from Gecko ([metabug](#)).

Q4 2019 - “browser.xhtml” is using an `<html>` root element ([bug](#)).

Projected Milestones

Note: this section hasn't been updated since late 2019. A number of these have already been completed but the document hasn't been updated.

Q4 2019 - GeckoView is no longer using XUL ([bug](#))

Q4 2019 (was Q3 2019) - All .xul files are removed from mozilla-central.

Q4 2019 (was: Q1 2019) - Last DTD removed from browser chrome

Sometime 2020 (was: Q4 2019) - XUL elements that have web equivalents are replaced by HTML elements ([metabug](#))

Sometime 2020 - XUL layouts that have web equivalents are removed ([metabug](#))

- **Q? 2019 (was: Q1 2019)** - Flexbox emulation reaches perf parity. Layout team has prioritized this work in H2, but it's too soon to make a confident estimation as of Sept 2018.
- **Q4 2019 (was: Q2 2019)** - Flexbox emulation enabled by default (necessary frontend callers for attrs like [width] removed).
- **Q4 2019 (was: Q2 2019)** - XUL flexbox implementation removed ([metabug](#)).
- **Q4 2019** - XUL stack implementation removed ([bug](#)).
- **Q4 2019** - XUL groupbox implementation removed ([bug](#)).
- **Q4 2019** - XUL deck implementation removed ([bug](#)).
- **Q4 2019** - XUL grid implementation removed ([bug](#)).

No Longer Active Milestones

These are milestones that we were originally projecting but are now not likely to be worked on until other needs arise around them.

Sometime 2020 (Probably not) - <xul:tree> removed ([metabug](#)). Depends on projects like about:library removing our most complex trees from the frontend, or an API-compatible frontend rewrite ([doc](#)).

Sometime 2019 (Maybe) - browser.xhtml -> browser.html