

MICHIGAN STATE
UNIVERSITY

Project Plan

Improvements to <select> Dropdown for Firefox

The Capstone Experience

Team Mozilla

Jared Beach

Mark Golbeck

Fred Luo

Tyler Maklebust

Michael Wright

Department of Computer Science and Engineering
Michigan State University

Fall 2016



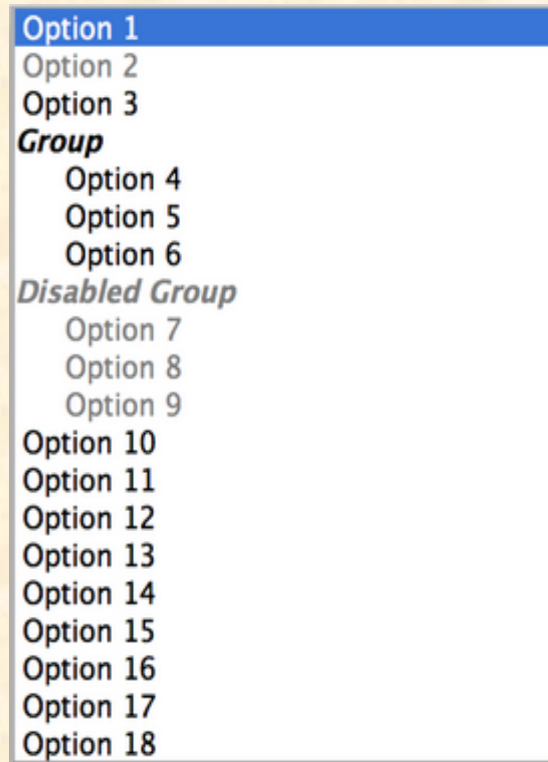
*From Students...
...to Professionals*

Functional Specifications

- Select dropdown improvements
 - Make disabled options more clear
 - Add search functionality for long lists
- Update single-process dropdown implementation to follow the multi-process (e10s) architecture



Legacy Firefox Select Dropdown



Design Specifications

- Change fonts in order to make disabled states within the dropdown more clear
- Make dropdowns easier to read with improved spacing and font properties
- Vertically center the dropdown's initially selected item on the select widget
- Increase the size of the items in dropdowns when responding to touch events



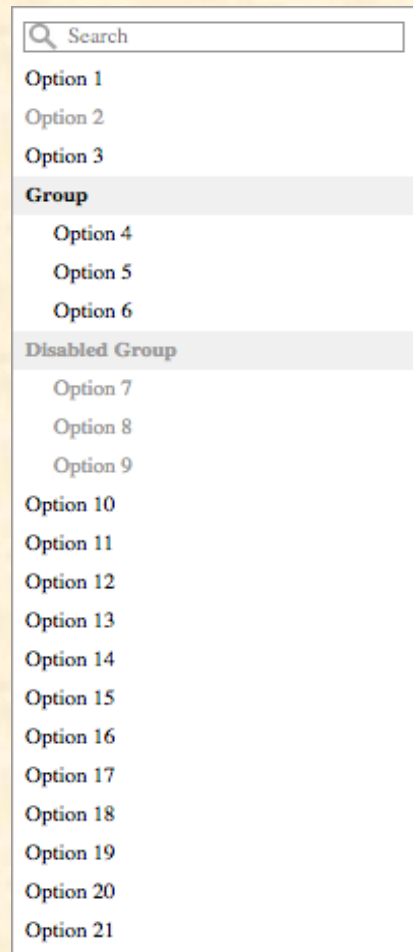
Screen Mockup: Disabled Options

Styling of disabled options and groups to make them more clear

Option 1
Option 2
Option 3
Group
Option 4
Option 5
Option 6
Disabled Group
Option 7
Option 8
Option 9
Option 10
Option 11
Option 12



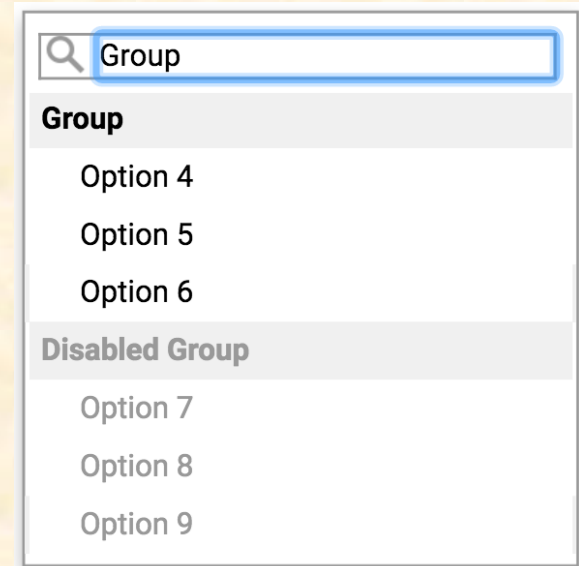
Screen Mockup: Dropdown Search



A mockup of a dropdown menu. At the top is a search bar with a magnifying glass icon and the text "Search". Below the search bar is a list of options. The first three options are "Option 1", "Option 2", and "Option 3". These are followed by a group header "Group" in bold. Under "Group" are "Option 4", "Option 5", and "Option 6". This is followed by a disabled group header "Disabled Group" in bold. Under "Disabled Group" are "Option 7", "Option 8", and "Option 9". Finally, there are "Option 10" through "Option 21" listed sequentially.

- Option 1
- Option 2
- Option 3
- Group**
 - Option 4
 - Option 5
 - Option 6
- Disabled Group**
 - Option 7
 - Option 8
 - Option 9
- Option 10
- Option 11
- Option 12
- Option 13
- Option 14
- Option 15
- Option 16
- Option 17
- Option 18
- Option 19
- Option 20
- Option 21

Addition of a search option
in dropdowns with many
options



A mockup of a dropdown menu, similar to the one on the left, but with the search bar containing the text "Group". The search bar has a magnifying glass icon. Below the search bar is a list of options. The first three options are "Option 4", "Option 5", and "Option 6". These are followed by a group header "Group" in bold. Under "Group" are "Option 4", "Option 5", and "Option 6". This is followed by a disabled group header "Disabled Group" in bold. Under "Disabled Group" are "Option 7", "Option 8", and "Option 9".

-
- Group**
 - Option 4
 - Option 5
 - Option 6
- Disabled Group**
 - Option 7
 - Option 8
 - Option 9

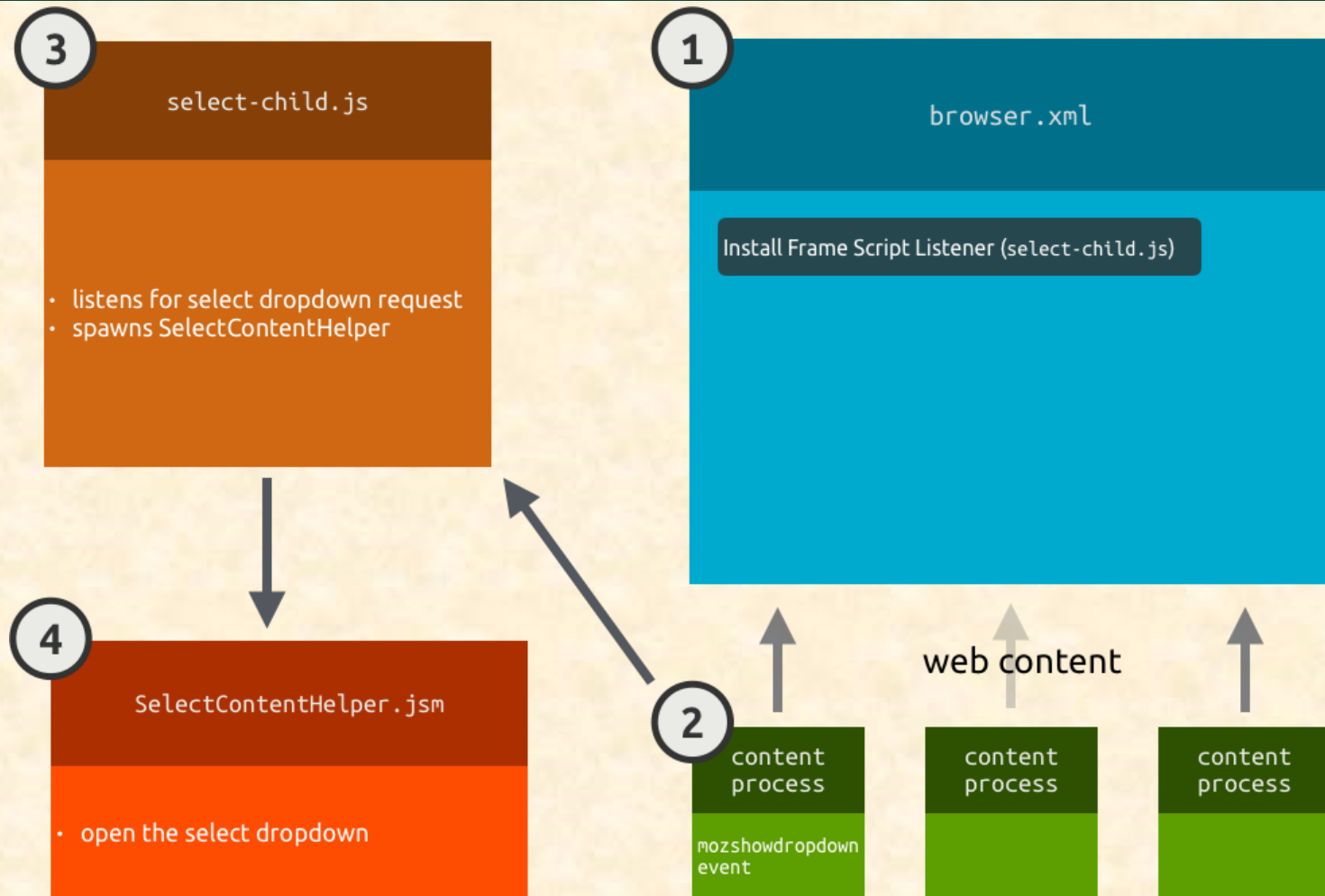


Technical Specifications

- Backend Architecture - C++ and JavaScript
- Frontend Architecture - CSS, XUL, and JS
- Development Environments:
 - Mercurial - Version Control
 - Bugzilla - Track progress of the bug
 - Firefox native debugger and 'Browser Toolbox'
 - Xcode - OSX
 - Visual Studio Community 2015 - Windows
 - TryChooser - Firefox's in-house Mercurial Extension



System Architecture



System Components

- Software Platforms / Technologies
 - CSS, JavaScript
 - C++, XUL
 - Mercurial Version Control System (VCS)
 - TryChooser - Amazon instances used to test builds across different platforms



Testing

- Firefox native debugger and 'Browser Toolbox' used to view changes to browser before testing
- Tests run against Mozilla's testing services using Firefox's in-house Mercurial Extension called TryChooser
- Tests will be developed simultaneously with features
- Code Review through MozReview

Risks

- Understanding code base
 - Firefox is a huge product, finding and understanding relevant portions will be critical
 - Mitigation: use client contacts as a resource in development
- Implementing search in dropdowns
 - Imported search algorithm has a different use case and will need to be heavily adapted to our functionality
 - Mitigation: Consulting with subject matter experts and researching similar use cases
- Code quality control
 - Mozilla has very strict coding standards that must be followed
 - Mitigation: study standards and get feedback early

