Introduction to Firefox Extensions
Section 1: Basic Setup
CS493 – University of Virginia

The full guide will lead to the creation of an extension that lets a user e-mail text that has been highlighted on a webpage. In this first section, we go through the steps of creating the files and folders for a basic extension. The end result of the section will be the phrase "Example Application" added to the status bar. Before you start, you may want to make a new Firefox developer profile so that you don't accidentally muck up your normal browser experience (instructions can be found at http://lifehacker.com/software/firefox/geek-to-live--manage-multiple-firefox-profiles-231646.php).

Begin with a folder, e.g. `C:\extensions\example\` or `~/extensions/example/`. Change to this directory. Make two new empty text files, `chrome.manifest` and `install.rdf`, and a folder named `chrome`. To demonstrate:

```
cd ~/extensions/
mkdir -p example/chrome/
cd example
touch chrome.manifest install.rdf
```

Open the `install.rdf` file to edit it. This file provides basic application and installation information for the extension. My version of the file is below to be used as a template. The bold & blue fields need to be customized by you.

```
1   <?xml version="1.0"?>
2   <RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3        xmlns:em="http://www.mozilla.org/2004/em-rdf#">
4
5     <Description about="urn:mozilla:install-manifest">
6
7       <!-- Extension information -->
8       <em:id>{64263247-1932-46b7-9b7c-58a7c7be4344}</em:id>
9       <em:version>1.0</em:version>
10      <em:type>2</em:type>
11      <em:name>Example Application</em:name>
12      <em:description>An example application for CS493</em:description>
13      <em:creator>Adrienne Felt</em:creator>
14      <em:homepageURL>http://www.cs.virginia.edu/</em:homepageURL>
15
16      <!-- Target Application, with min & max versions -->
17      <em:targetApplication>
18        <Description>
19          <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
20          <em:minVersion>1.5</em:minVersion>
21          <em:maxVersion>2.0.*</em:maxVersion>
22        </Description>
23      </em:targetApplication>
24
25    </Description>
26  </RDF>
```

The mysterious-looking hex string used for the id field (line 8) is a GUID (Globally Unique Identifier). You can make your own at http://extensions.roachfiend.com/cgi-bin/guid.pl; note that the dashes and brackets must be included. The `type` field on line 10 must be set to 2; this lets Firefox know that the application is an extension. On line 19, this GUID identifies Firefox as the target of the extension's installation.

Next, open `chrome.manifest` to edit it. It will only have two lines:

```
content       example       chrome/content/
overlay       chrome://browser/content/browser.xul chrome://example/content/example.xul
```

The first line specifies the type of material within the chrome package, the name of the package, and the location of the extension's files. Make sure the pathname has a trailing slash, and the name of the chrome package must be all lowercase. The second line adds the extension's overlay (`example.xul`) to the browser window (`browser.xul`) whenever the browser window loads. We'll make this overlay file in a minute.

Note the path for your extension in that second line: it is *not* the actual path on your hard drive that you're using to develop (e.g., `~/extensions/example/`). Instead, you need to create a pointer to the files by registering the extension with Firefox. Go to your Firefox profile folder (instructions can be found at http://kb.mozillazine.org/Profile_folder_-_Firefox). Find the `extensions` folder inside the profile folder, or create it if it doesn't exist. Inside the `extensions` folder, create a new text file with the GUID of your extension as the name of the file. Inside that file, put only the path to your extension folder (e.g., `~/extensions/example`).

Now, go back to your development directory, `~/extensions/example`. Create a subfolder, `chrome`, with a subfolder `content`. (Running `mkdir -p chrome/content` will make both at once.) All of the extension files we'll be editing will be in this `content` folder. Create the file `example.xul` in `content`, and open it to edit it. Skeleton code to add "Example Application" to Firefox's status bar follows:

```
1   <?xml version="1.0"?>
2
3   <overlay id="example"
4            xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
5
6       <!-- Adds text to the status bar -->
7       <statusbar id="status-bar">
8           <statusbarpanel id="my-panel" label="Example Application" />
9       </statusbar>
10
11  </overlay>
```

The overlay specifies what should be added to the main browser window. To see the result, re-start Firefox; the extension will be automatically installed as Firefox loads. (You always need to re-start Firefox to see changes.) In the next section of the guide, additional elements will be added to the overlay to insert hooks into the Tools and right-click menus.