Chapter 7. Using Files

Chapter 4 focused on common operations for working with files such as reading, writing, and removing files. It did not discuss in detail the ways in which an application can utilize a file residing in its filesystem. However, in most scenarios, you do not need to read the contents of a file in order to use it.

Filesystem URLs

Perhaps the easiest way to use a file is to reference it by URL. The HTML5 Filesystem API exposes a new type of URL scheme, `filesystem:`. The structure of a filesystem URL is as follows:

```
filesystem:<ORIGIN>/<STORAGE_TYPE>/<FILENAME>
```

The format is the scheme, `filesystem:`, followed by the application's origin, the storage type the filesystem was requested with, and finally, the full path of the file or folder as it resides on the filesystem. For example, if your application lived at `http://www.example.com/myapp` and used `PERSISTENT` storage, the filesystem URL to the root folder would be: `filesystem:http://www.example/temporary/`.

So why are filesystem URLs handy? They're useful because they can be used anywhere a normal URL can be used. For example, you could cache a *.js* file and later use that file's URL to fill a `script.src` on demand. You could do the same with a *.html* file, but instead populate an `iframe.src`. Lastly, you could display an image by setting its `src` to a filesystem URL.

Currently in Chrome, filesystem URLs are guessable—one can be constructed manually by knowing the proper format. This might not always be the case and other browser ...