HTML5 Offline Web Aplications

Jfokus 2011
Stockholm
Peter Lubbers,
Kaazing

about:peterlubbers

peter lubbers

Search

peter lubbers
peter lubbers **html5**
peter lubbers **kaazing**
twitter peter lubbers

I'm Feeling Lucky »

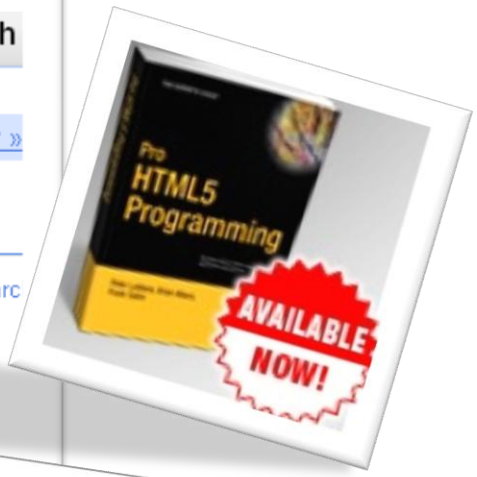
About 196,000 results (0.37 seconds)

Advanced searc

Run Lake Tahoe

Feb 11, 2010 ... **Peter Lubbers**' blog about running and ultrarunning. Peter is the 2007 and 2009 ultrarunner.net series champion and three-time winner of the ...

runlaketahoe.blogspot.com/ - Cached - Similar

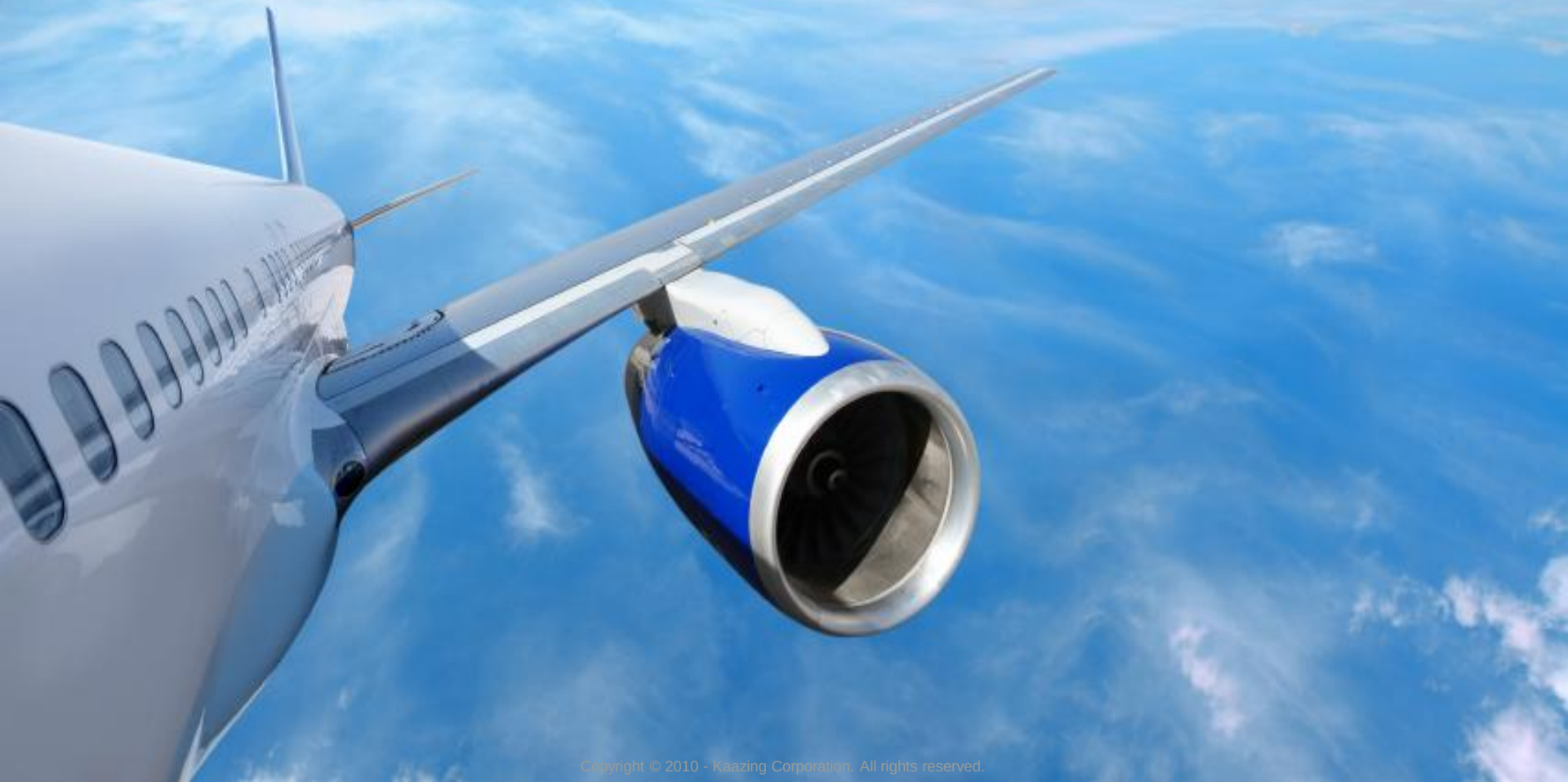


Agenda

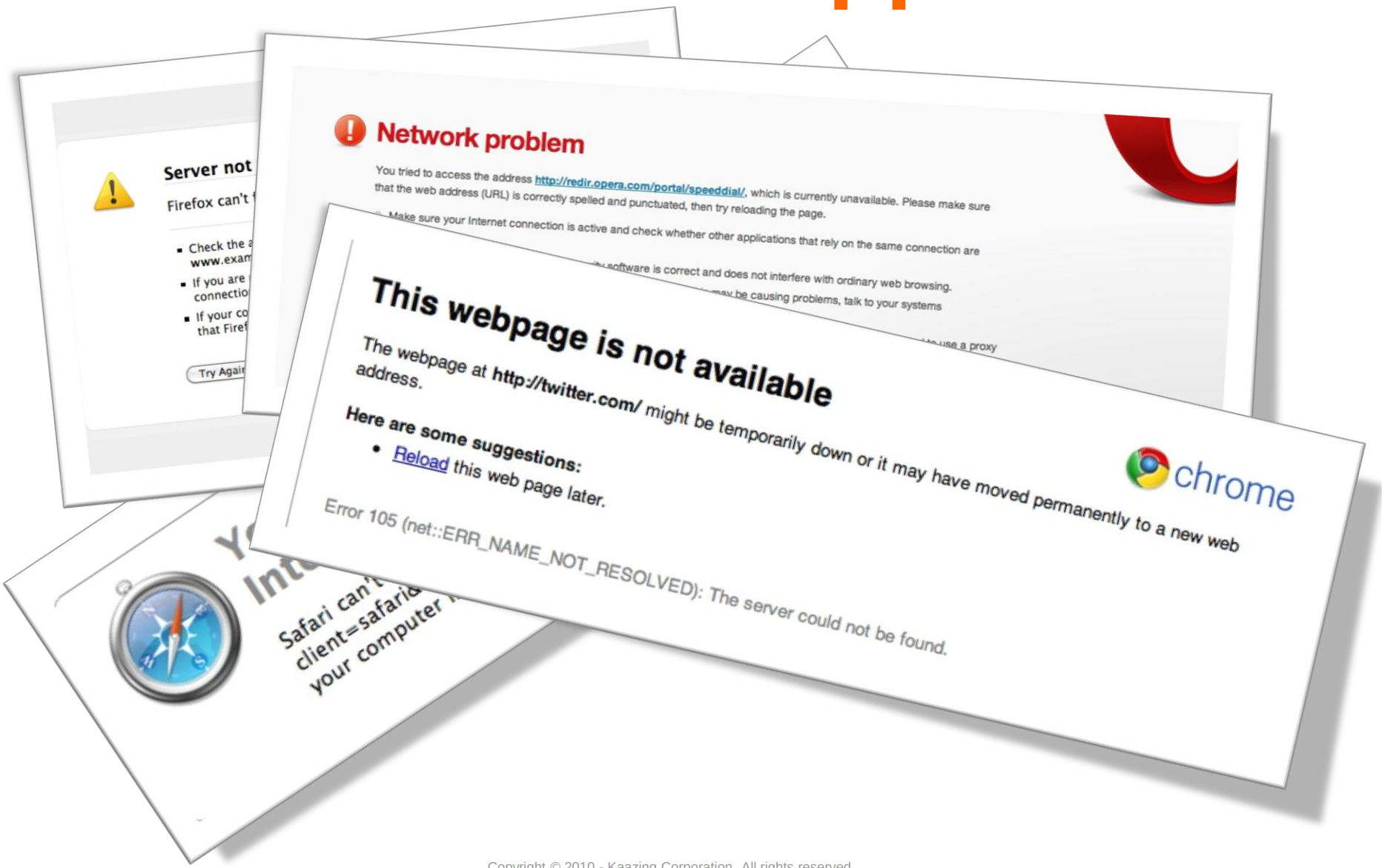
- About Offline Web Apps
- Creating Offline Web Apps
- Server Configuration
- Tips, Tricks, and Q&A
- Pay attention, win prizes!
- Tweet it: #jfokus @peterlubbers #html5 #appcache

HTML5 Offline Web Apps

Allow you to keep using web apps and sites without a network connection (for example, on an airplane or in rural areas and subways)

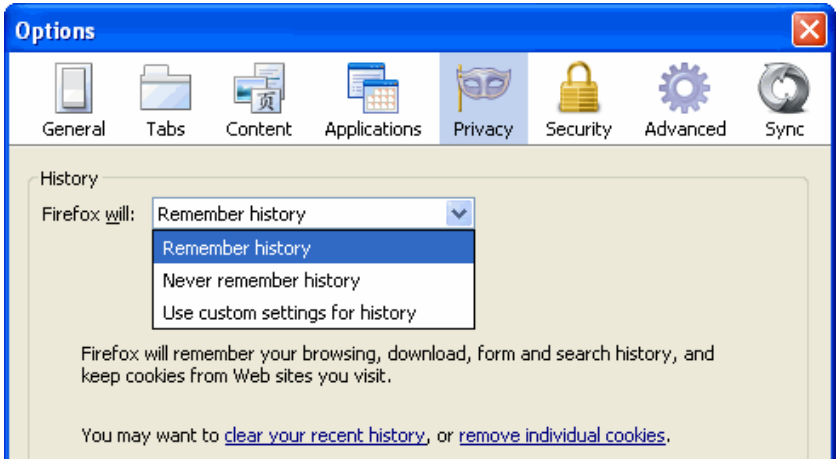


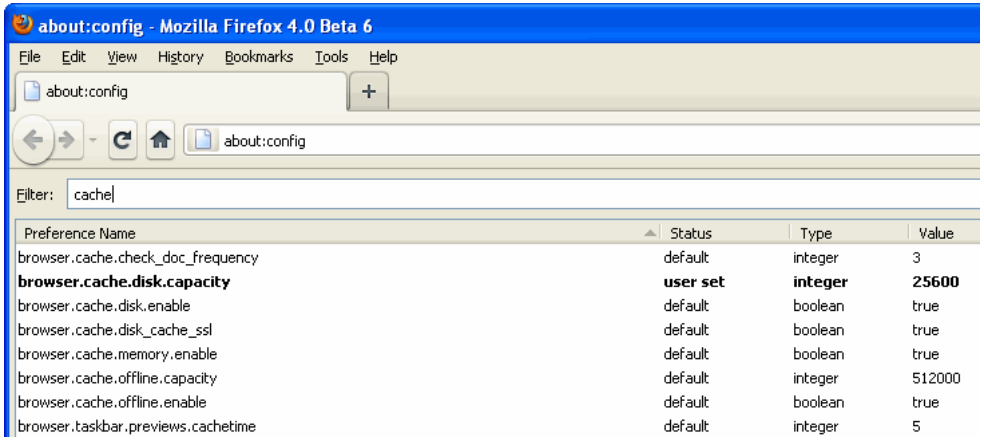
Offline without AppCache



Traditional Caching

- Cache duration can be controlled through Web Server's Cache HTTP headers
  - For example, in Apache's `.htaccess` file (see example on next slide)
- Not reliable for offline use
- Browser caching settings also apply





Web Server Caching Settings

.htaccess File

```
<IfModule mod_expires.c>
  Header set cache-control: public
  ExpiresActive on
# your document html
  ExpiresByType text/html                    "access"
# rss feed
  ExpiresByType application/rss+xml        "access plus 1 hour"
# favicon (cannot be renamed)
  ExpiresByType image/vnd.microsoft.icon   "access plus 1 week"
# media: images, video, audio
  ExpiresByType image/png                  "access plus 1 month"
  ExpiresByType image/jpg                  "access plus 1 month"
  ExpiresByType image/jpeg                 "access plus 1 month"
  ExpiresByType video/ogg                  "access plus 1 month"
  ExpiresByType audio/ogg                  "access plus 1 month"
  ExpiresByType video/mp4                  "access plus 1 month"
# webfonts
  ExpiresByType font/ttf                   "access plus 1 month"
  ExpiresByType font/woff                  "access plus 1 month"
  ExpiresByType image/svg+xml              "access plus 1 month"
# css and javascript
  ExpiresByType text/css                   "access plus 1 month"
  ExpiresByType application/javascript     "access plus 1 month"
  ExpiresByType text/javascript            "access plus 1 month"
</IfModule>
```

Source: Paul Irish & Divya Manian http://html5boilerplate.com/

Offline Web Applications

- New HTML5 Specification
- Also known as Application Cache (AppCache)
- WHATWG: http://www.whatwg.org/specs/web-apps/current-work/multipage/offline.html#offline
- W3C: http://dev.w3.org/html5/spec/offline.html#offline
- Spec is primarily aimed at browser developers to ensure interoperability

Offline Web Applications

- Allow you to cache pages that have *not* been visited
- Browsers cache data in an *Application Cache*
- Once resources are cached, you can access them very quickly (without a network request)
- HTML5 also allows online and offline detection
- Using offline mechanism allows you to easily *prefetch* site resources (speeds up pages, but uses bandwidth)
- Pages served using TLS (SSL) can also be included to work offline

Browser Support for Offline Web Applications

Source: http://caniuse.com/

(the best site for checking browser support for HTML5 features)

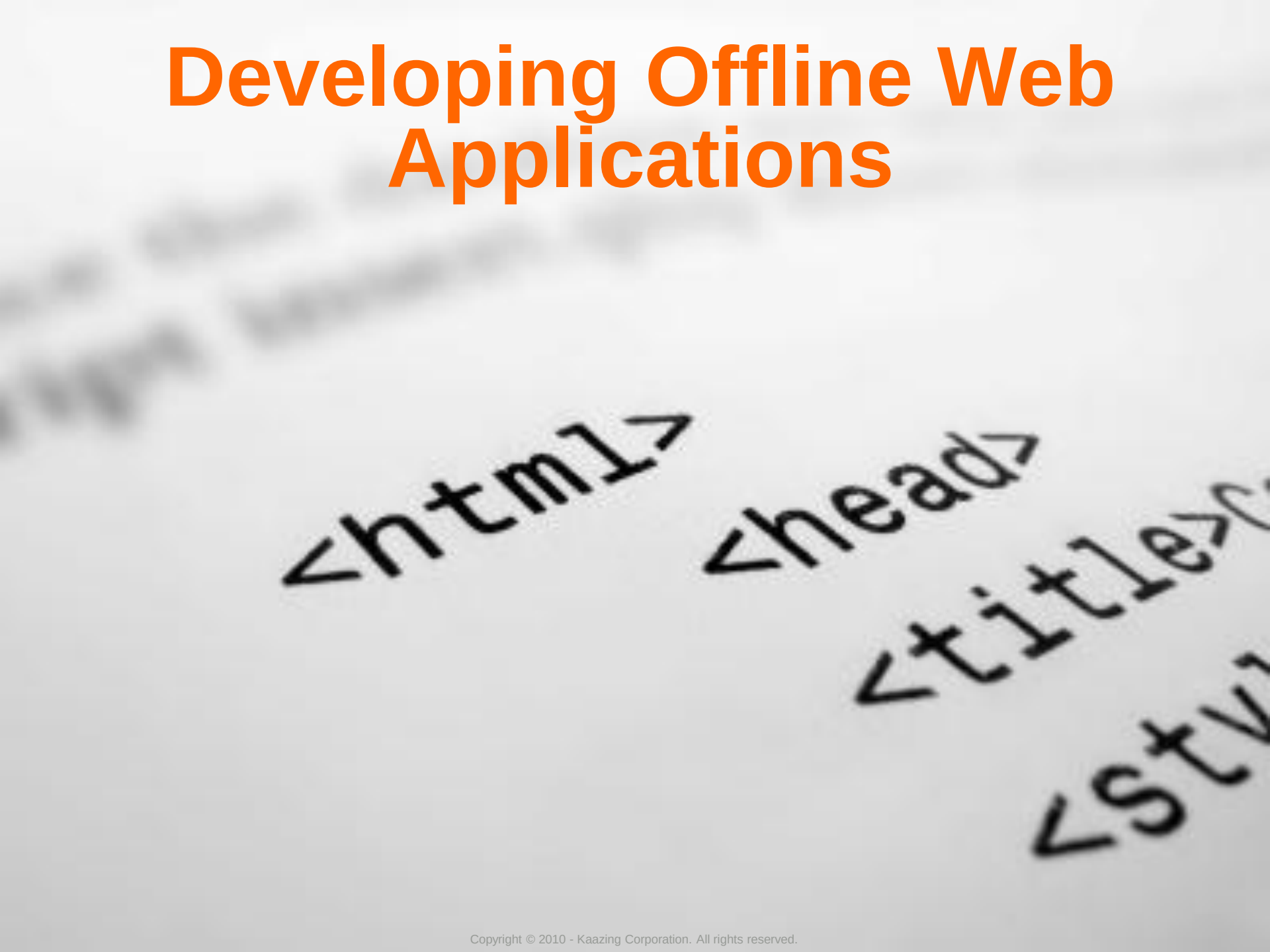
Hi-res browser logos: http://paulirish.com/2010/high-res-browser-icons/

The idea is simple…

Add a manifest file. reference it. Done.

But in reality…

Developing Offline Web Applications



Example Manifest File

Manifest File

```
CACHE MANIFEST
# manifest version 1.0.1
# Files to cache
index.html
cache.html
html5.css
image1.jpg
img/foo.gif
http://www.example.com/styles.css

# Use from network if available
NETWORK:
network.html

# Fallback content
FALLBACK:
/ fallback.html
```

Using a Manifest File in HTML

- Reference the manifest file:
  - **<name>.manifest** – it must match the name of the manifest file
  - Add as attribute to HTML element

HTML

```
<!DOCTYPE html>
<html manifest="offline.manifest">
  <head>
    <title>HTML5 Application Cache</title>
```

Files: http://bit.ly/9pJ1Zq or:
ch.kaazing.com/training/offline/peter-lubbers-html5-offline-web-apps-presentation-code.zip

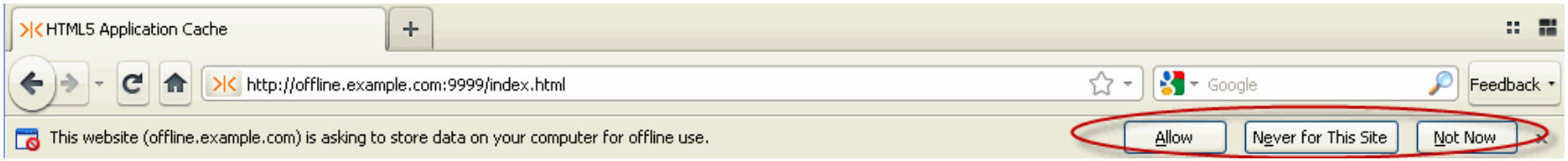
Demo

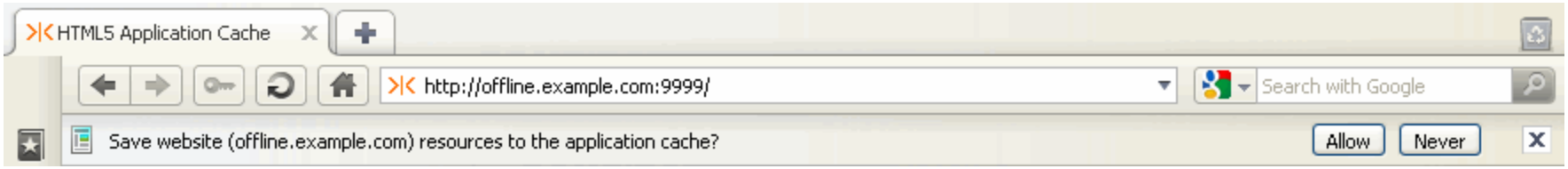
Initial Cache Sequence

1. Access the page (with the manifest attribute)
2. Page is loaded and page's resources are loaded (from the server)
3. Manifest is encountered and parsed, all files flagged for caching are loaded in the background
4. Go offline (regular caching is also in effect, so watch for false positives)
5. Access a `CACHE:` resource (loads from cache)
6. Access a `NETWORK` resource (`FALLBACK` content is served, files will be available if you go back online)

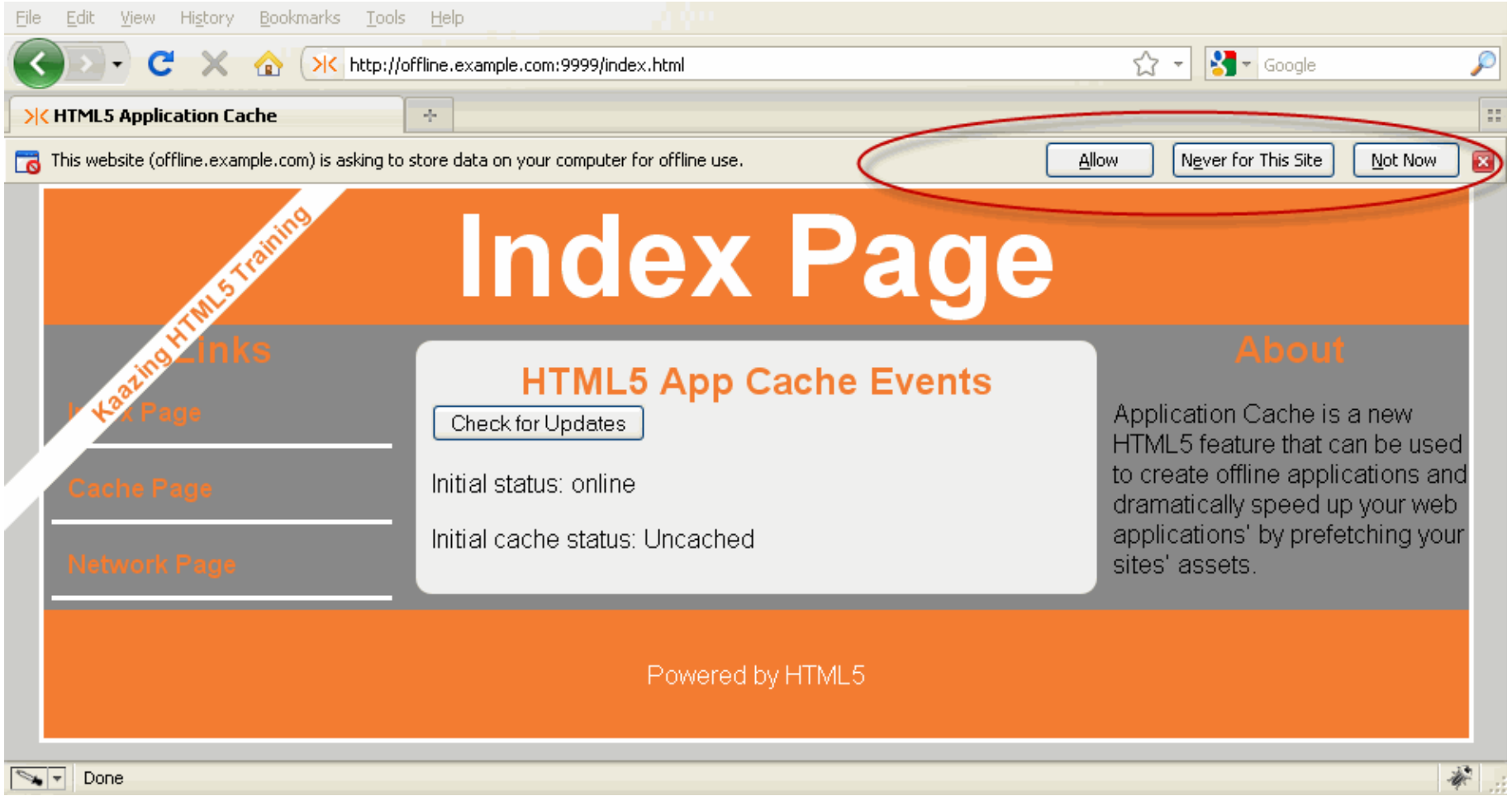
Browser Notification

- Users may have to opt in (similar to Geolocation)
- Configurable
- Not default in all browsers

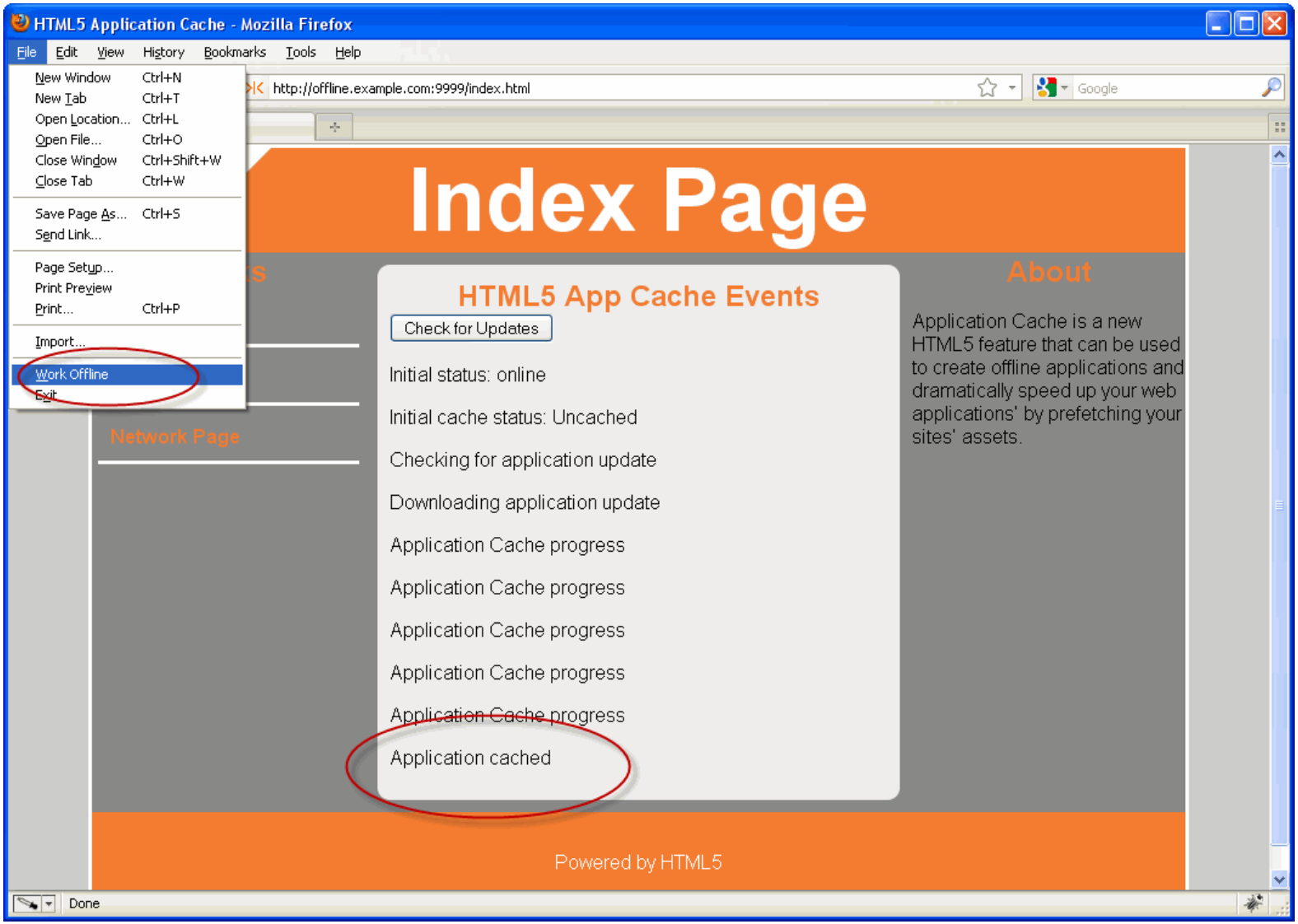




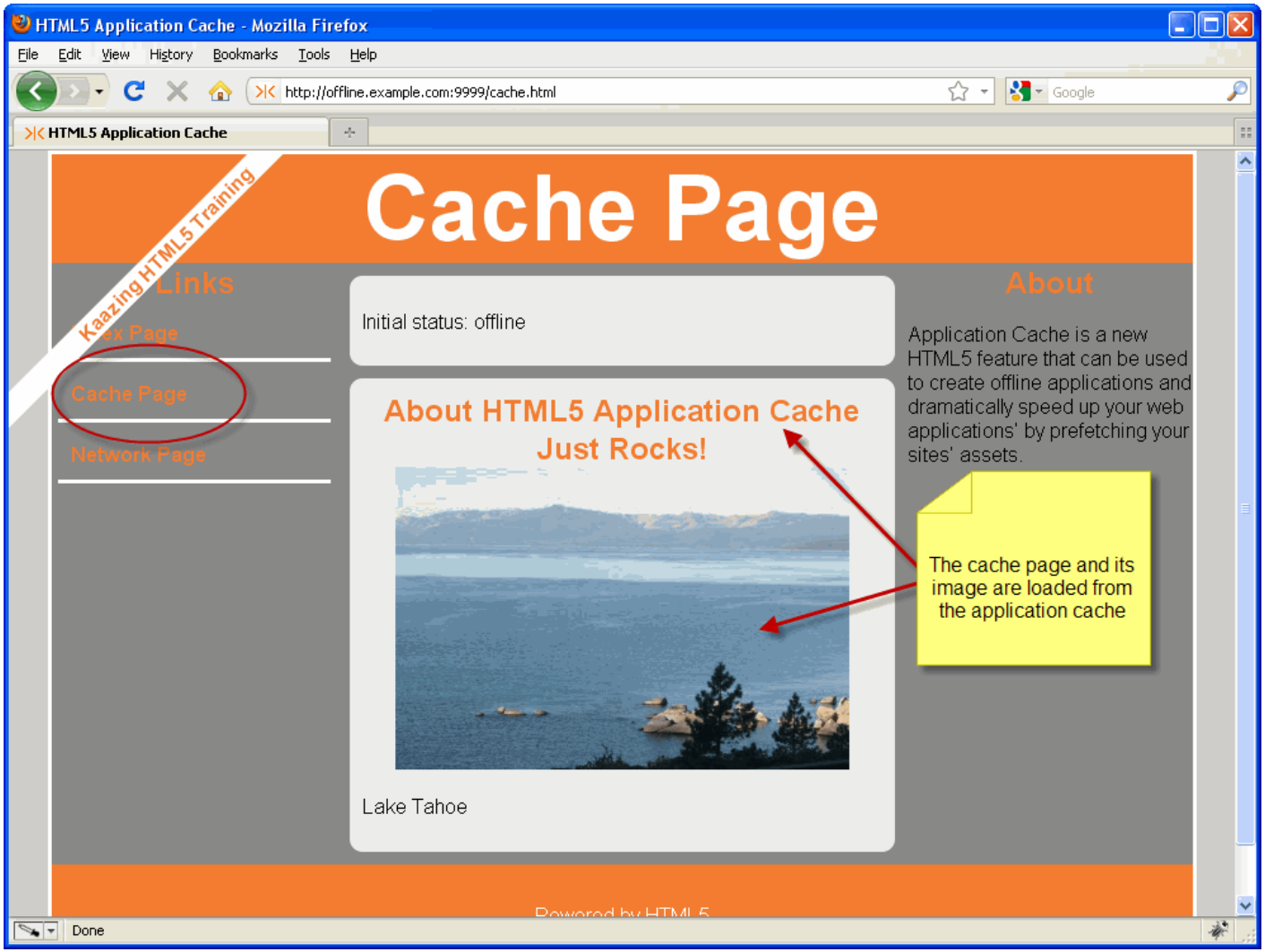
Step 1: First Page Load



Step 2: Going Offline



Step 3: Offline Web Page



Step 4: Network Page

Consecutive Load Sequence 1

1. Go back to online mode
2. Change the `cache.html` page on the server
3. Reload the `cache.html` page in the browser
4. The (old) page loads from the application cache (The changes do not appear!)
5. The browser checks to see if the referenced `manifest` file has been updated and does nothing since it has not been modified

Consecutive Load Sequence 2

6. Update the manifest file (make a trivial change, like a version number comment update)
7. Reload the cache page in the browser
8. The (old) page loads from the application cache (This is always the first action from the browser, so the changes *still* don't appear!)
9. The browser checks to see if the referenced manifest has been updated and since it has, it downloads all the files flagged to be cached
10. The new files are now in the application cache, reload the page once more to see the latest changes

</Demo>

The Manifest File

- Manifest file has three sections:
  - `CACHE:`
  - `NETWORK:`
  - `FALLBACK:`
- Multiple sections (of the same kind) are allowed
- First line *must* be `CACHE MANIFEST`
- Comments start with # (don't use inline comments)
- An application cache is created using the manifest's complete URL (you can have multiple manifest files in a site)

CACHE: Section

- To cache files in the AppCache, include them in the `CACHE:` section or list files directly under `CACHE MANIFEST` (default is to cache files)
- Add one file per line (full name required)
- Files can contain path information or even be an absolute URL
- Application caches can’t include fragment identifiers (#) or wildcards
- Case sensitive
- Files that reference the manifest file will automatically be cached

NETWORK: Section

- Also called the online whitelist
- Files listed in this section listed will not be loaded from the cache, but retrieved over the network (from the server) if online
- You can specify “*”
  - Sets the online whitelist wildcard flag to “open”
  - Access to resources on other origins will not be blocked

FALLBACK: Section

- Provides a way to fall back if resources cannot be found
- Specify a fallback namespace and a fallback page for that namespace:

```
/ fallback.html
```

- You can only list one fallback namespace

Checking for Browser Support

JavaScript

```
// Simple method
if(window.applicationCache) {
  // this browser supports offline web apps
}

//Or just use Modernizr (source: http://www.modernizr.com/)
if (Modernizr.applicationcache){
  // We have offline web app support! Continue
     operation,
  // indicating to the user that the app will sync up
     once they get back online
} else {
  // No offline support, show errors if the user goes
     offline
}
```

Checking for Online and Offline Events

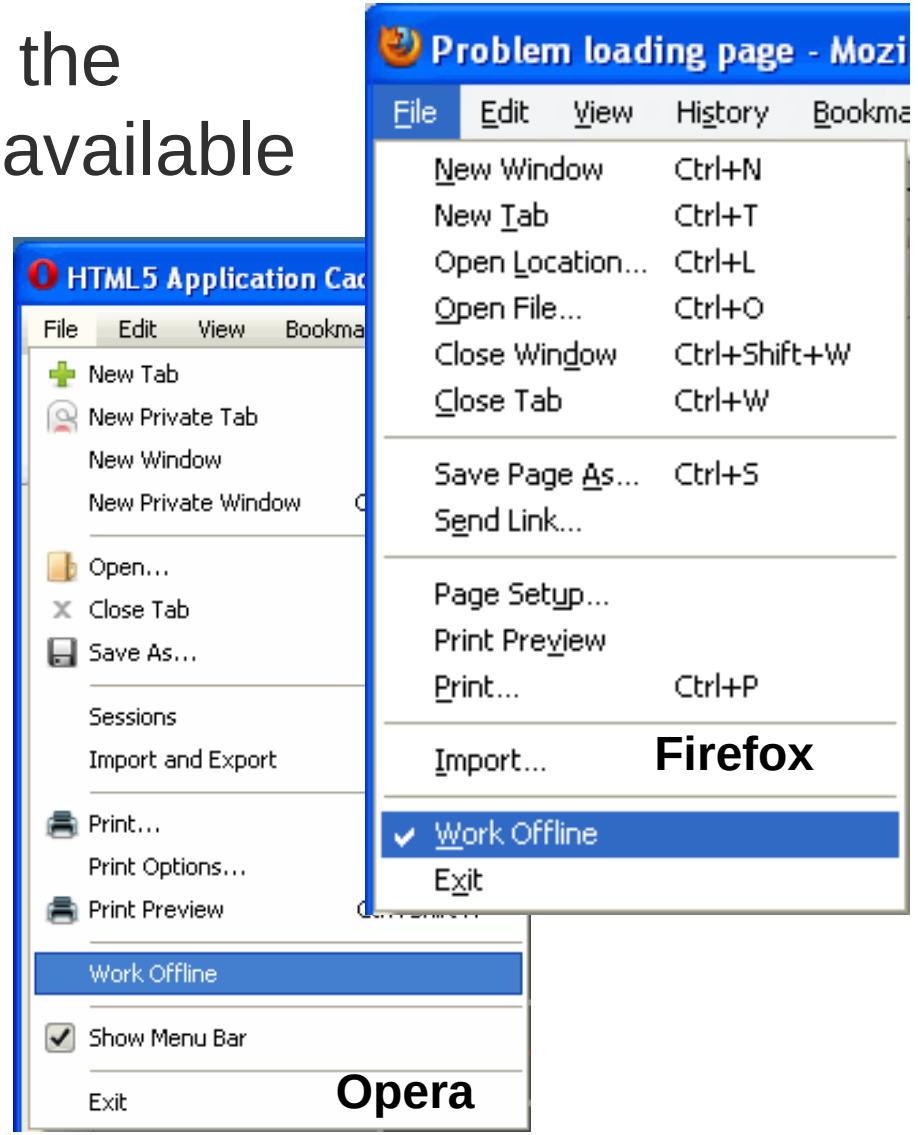
JavaScript

```
window.addEventListener("online", function(e) {
  log("Application is now online");
}, true);

window.addEventListener("offline", function(e) {
  log("Application is now offline");
}, true);
```

Using Web Pages Offline

- Test offline pages using the “Work Offline” feature if available (Not in Chrome and Safari)
- Disconnect your computer (does not work for localhost)
- Watch out for false positives (regular browser caching)



Working Offline

Browser	Work Offline
Chrome	Disconnect from the network
Firefox	File > Work Offline
Safari	Disconnect from the network
Opera	File > Work Offline
Internet Explorer	N/A (Does not support App Cache)

ApplicationCache Events

- The window.applicationCache object fires several events related to the state of the cache
- `window.applicationCache.status` is a numerical property indicating the state of the cache
  - 0 UNCACHED
  - 1 IDLE
  - 2 CHECKING
  - 3 DOWNLOADING
  - 4 UPDATEREADY
  - 5 OBSOLETE

Event Callback Attributes

Callback Attribute	Event
onchecking	CHECKING
ondownloading	DOWNLOADING
onupdateready	UPDATEREADY
onobsolete	OBSOLETE
oncached	CACHED
onerror	ERROR
onnoupdateready	NOUPDATE
onprogress	PROGRESS

Checking App Cache Status

JavaScript

```
window.applicationCache.onchecking = function(e) {
    log("Checking for application update");
}
window.applicationCache.onnoupdate = function(e) {
    log("No application update found");
}
window.applicationCache.onupdateready = function(e) {
    log("Application update ready");
    window.applicationCache.swapCache();
}
window.applicationCache.onobsolete = function(e) {
    log("Application obsolete");
}
```

Checking App Cache Status

JavaScript

```
window.applicationCache.ondownloading = function(e) {
    log("Downloading application update");
}
window.applicationCache.oncached = function(e) {
    log("Application cached");
}
window.applicationCache.onerror = function(e) {
    log("Application cache error");
}
window.applicationCache.onprogress = function(e) {
    log("Application Cache progress");
}
```

Serving the Manifest File

- Manifest files have the MIME type `text/cache-manifest`
- Most web servers need to be configured to serve the manifest files correctly
  - Served correctly by default by Apache on Ubuntu Linux and Kaazing WebSocket Gateway
  - For Python on Windows/Mac OS X or Apache, update the configuration files with the MIME type

Apache Configuration

mime.types File

```
# Apache mimetype configuration
# APACHE_HOME/conf/mime.types
text/cache-manifest manifest
```

Or:

.htaccess File

```
# Apache mimetype configuration
AddType text/cache-manifest .manifest
```

Prevent Manifest Caching

.htaccess File

```
# Cache settings for the manifest file
<IfModule mod_expires.c>
  Header set cache-control: public
  ExpiresActive on
.
.
.
# Prevent receiving a cached manifest
  ExpiresByType text/cache-manifest       "access plus 0 seconds"
.
.
.
</IfModule>
```

Source: *Introducing HTML5*, Offline Chapter, Bruce Lawson and Remy Sharp

Python Configuration

mimetypes.py File

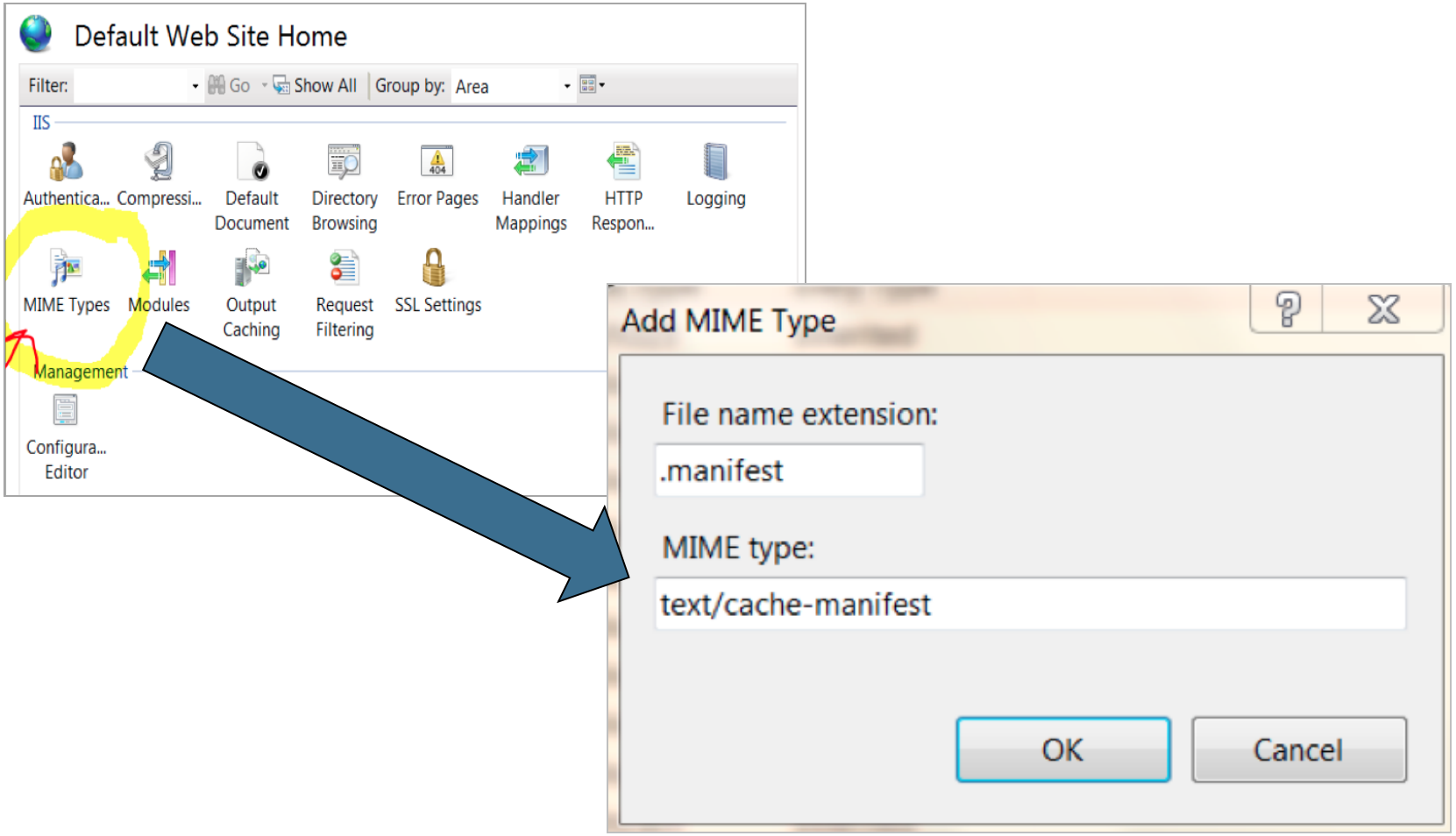
```
# Python SimpleHTTPServer mimetype Configuration
# python -m SimpleHTTPServer 9999)
'.manifest'     : 'text/cache-manifest',
```

Windows:
PYTHON_HOME/Lib/mimetypes.py, for example:
C:\Python26\Lib\mimetypes.py

Mac:
PYTHON_HOME/Lib/mimetypes.py, for example:
/System/Library/Frameworks/Python.framework/Versions/2.6/lib/python2.6/mimetypes.py

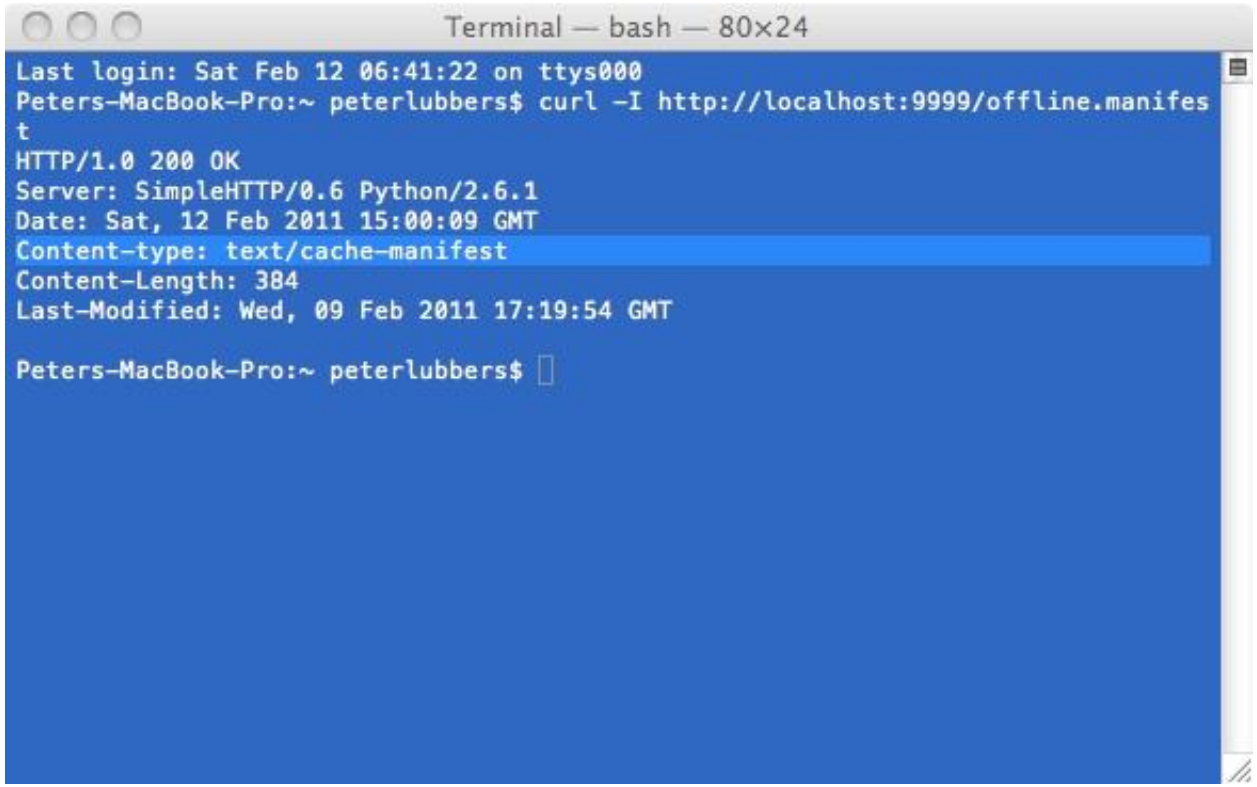
Important: If you do not have a mimetypes.py file, you can use `mimetypes.py` from the `offline/ mac-config-file` example folder. If you already have a compiled `mimetypes.pyc` file in the same directory, ensure that the permissions on this file are changed to read/write. When you start Python with the new file, Python compiles it and generates or overwrites the `mimetypes.pyc`.

Microsoft IIS Configuration



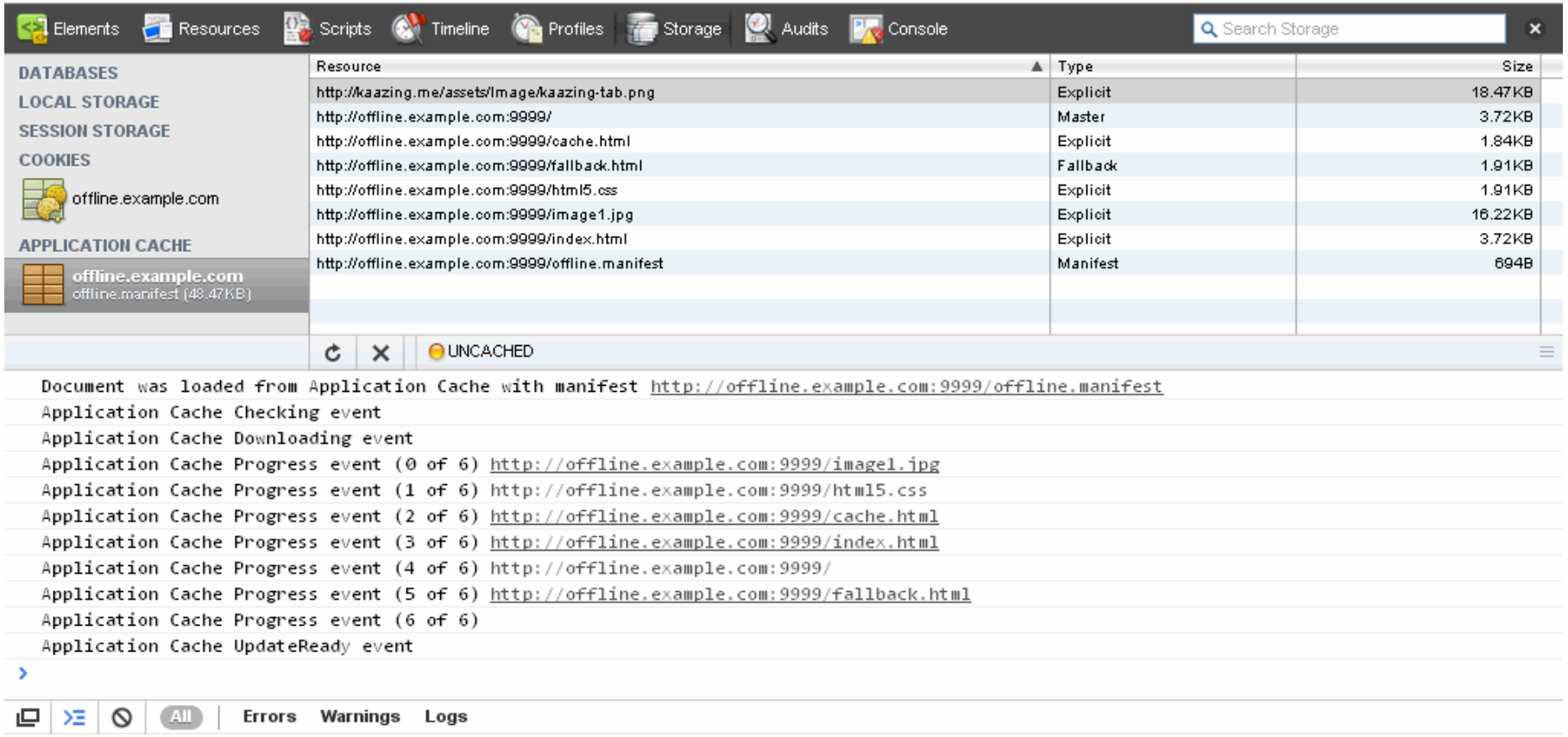
Verify Mime Type

- Verify that the manifest file is served correctly:
  - `curl -I http://offline.example.com/offline.manifest|` **
  - Check Chrome Dev Tools console



****Source**: *Introducing HTML5*, Offline Chapter, Bruce Lawson and Remy Sharp

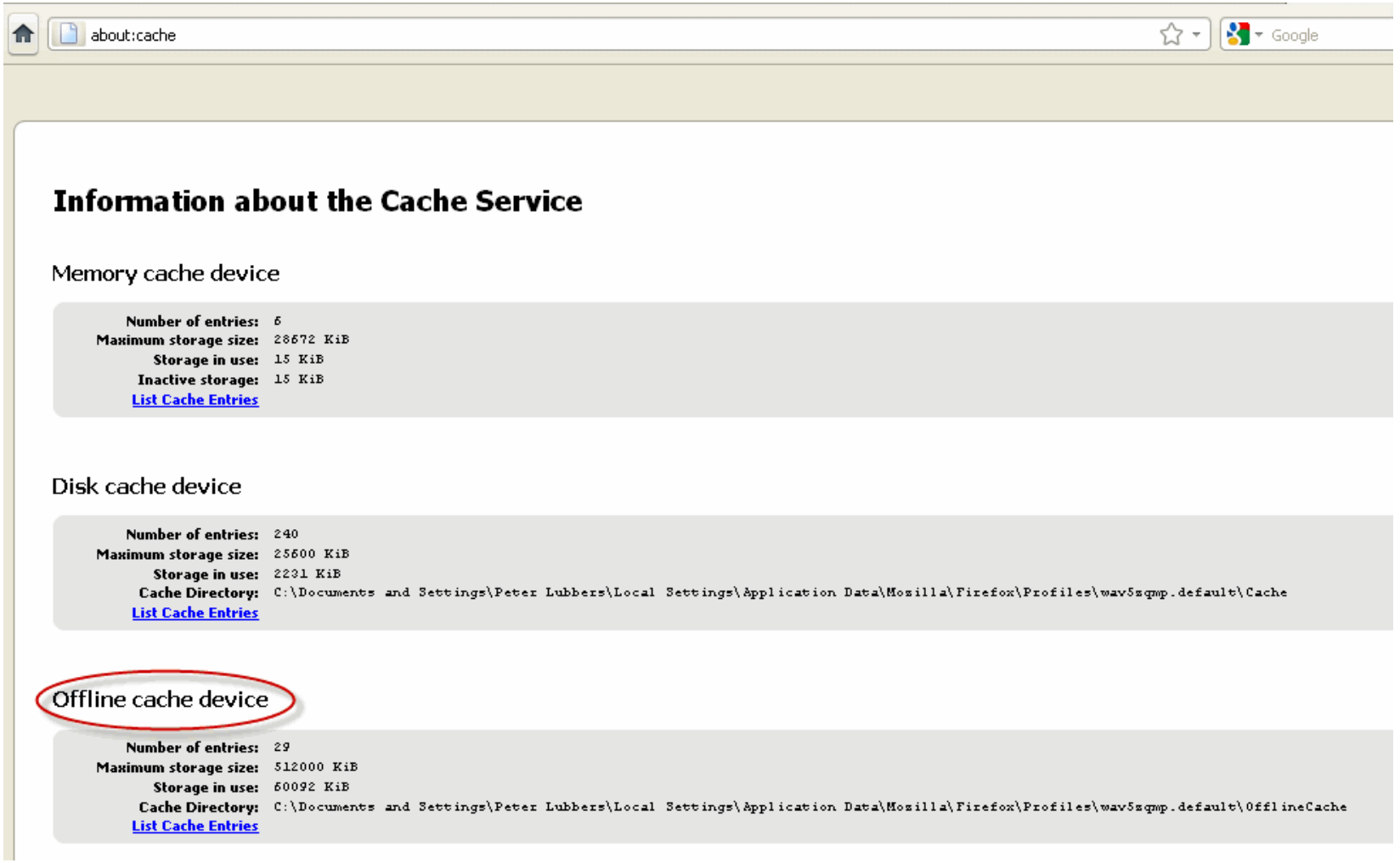
Chrome Developer Tools



KAAZING

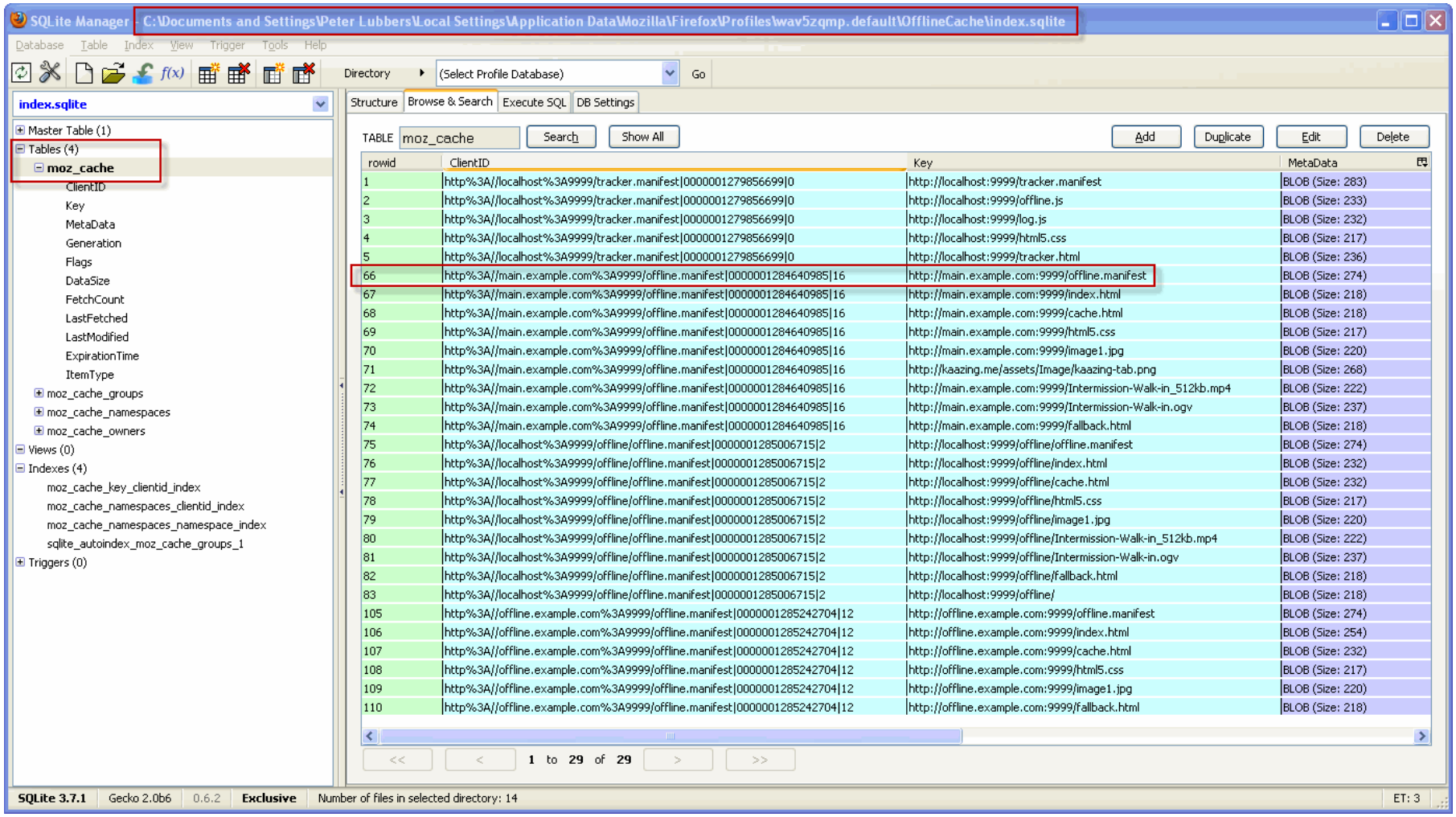
Offline Web Applications Tips and Tricks

Accessing the Offline Cache



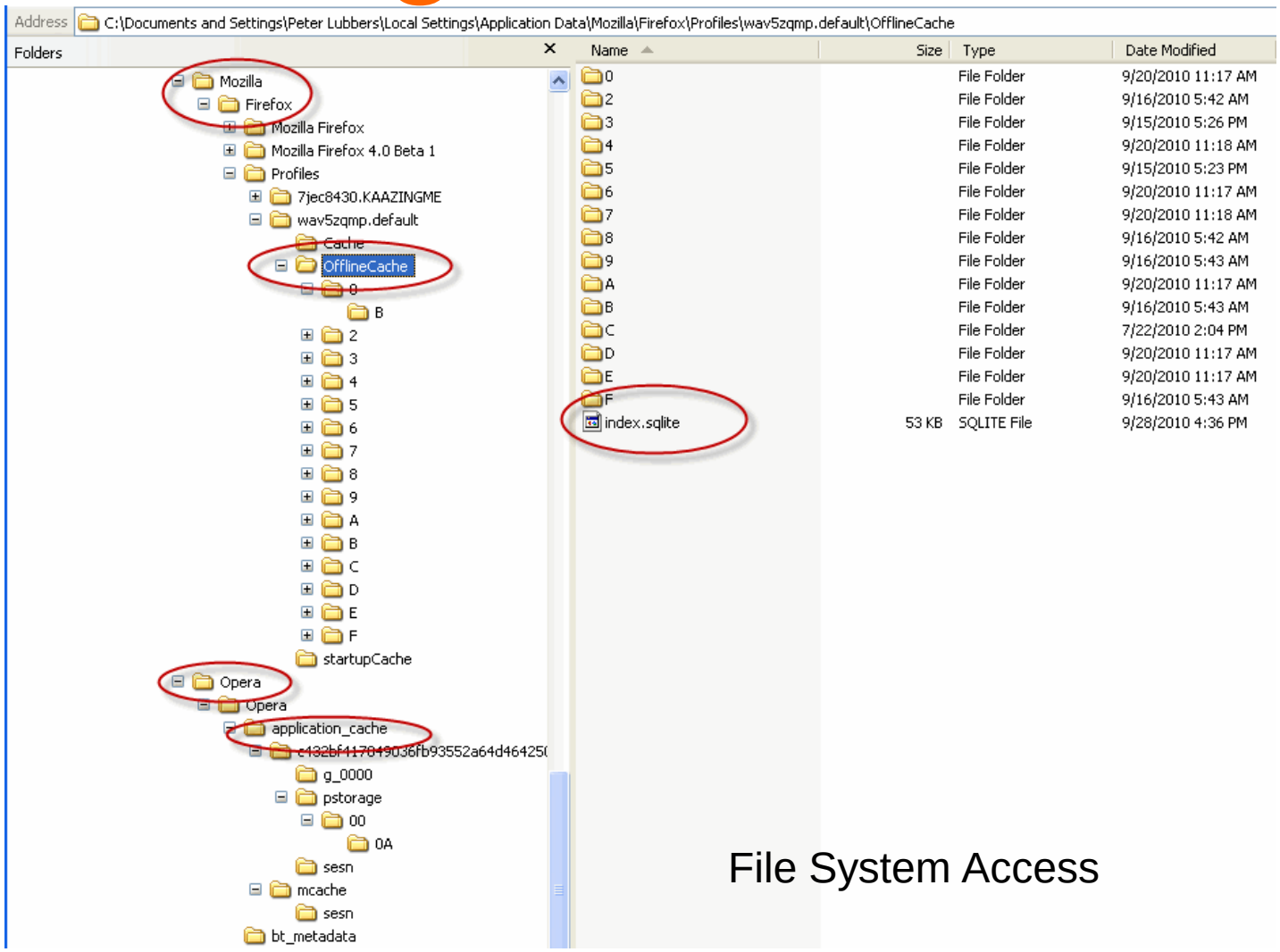
Firefox: `about:cache`

Accessing the Offline Cache



Firefox SQLite Manager Add-on: https://addons.mozilla.org/en-US/firefox/addon/5817/

Accessing the Offline Cache



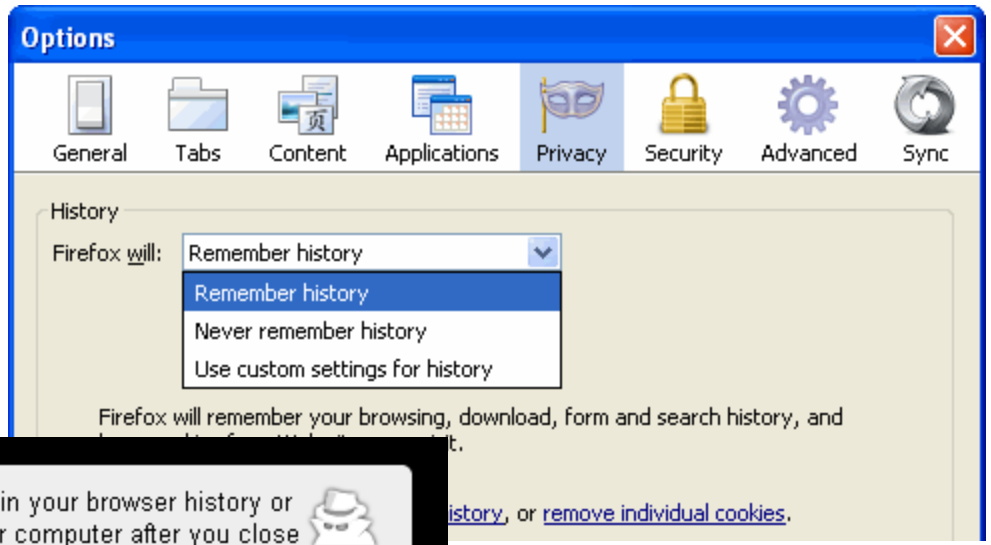
File System Access

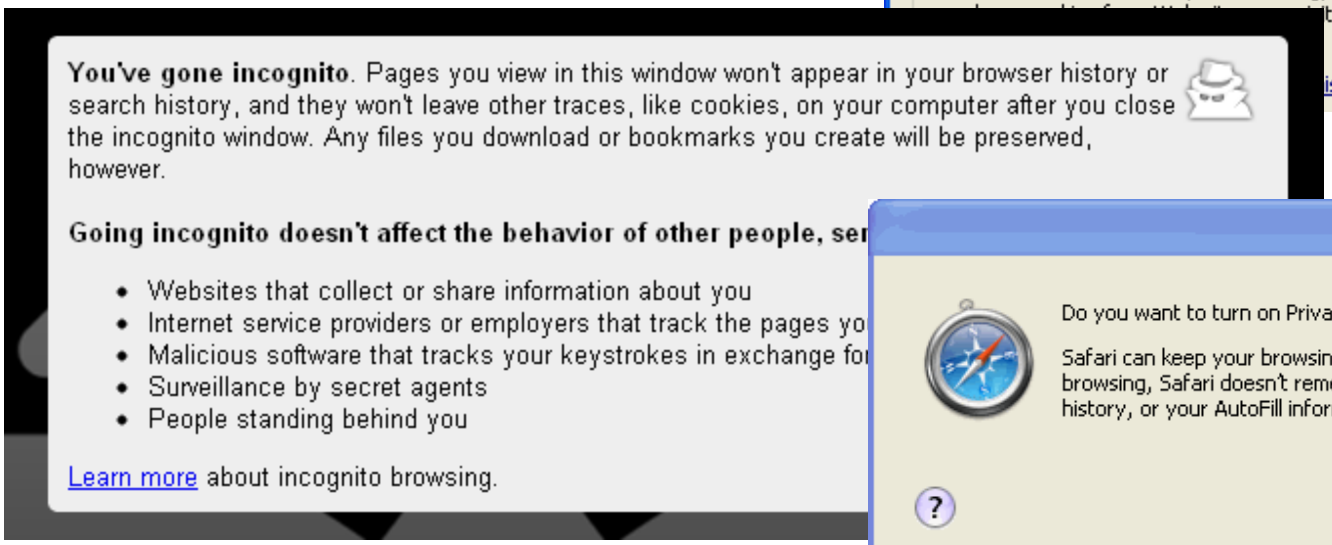
Security Considerations

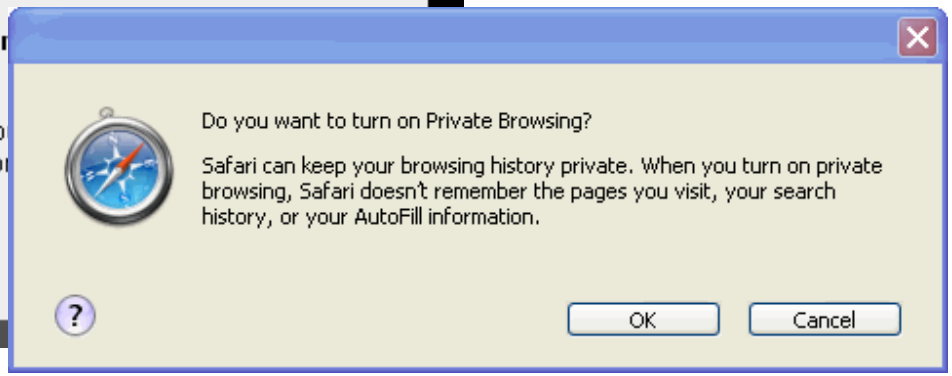
- Others browsing the same site (on the same machine in the same browser) can potentially access your cached data (data is cached based on the manifest file URL)
- Do not store sensitive, personal data in the application cache

Private Browsing

- Most private browsing modes prevent writing to application cache
  - For example, Safari's Private Browsing mode, and Chrome's Incognito Mode



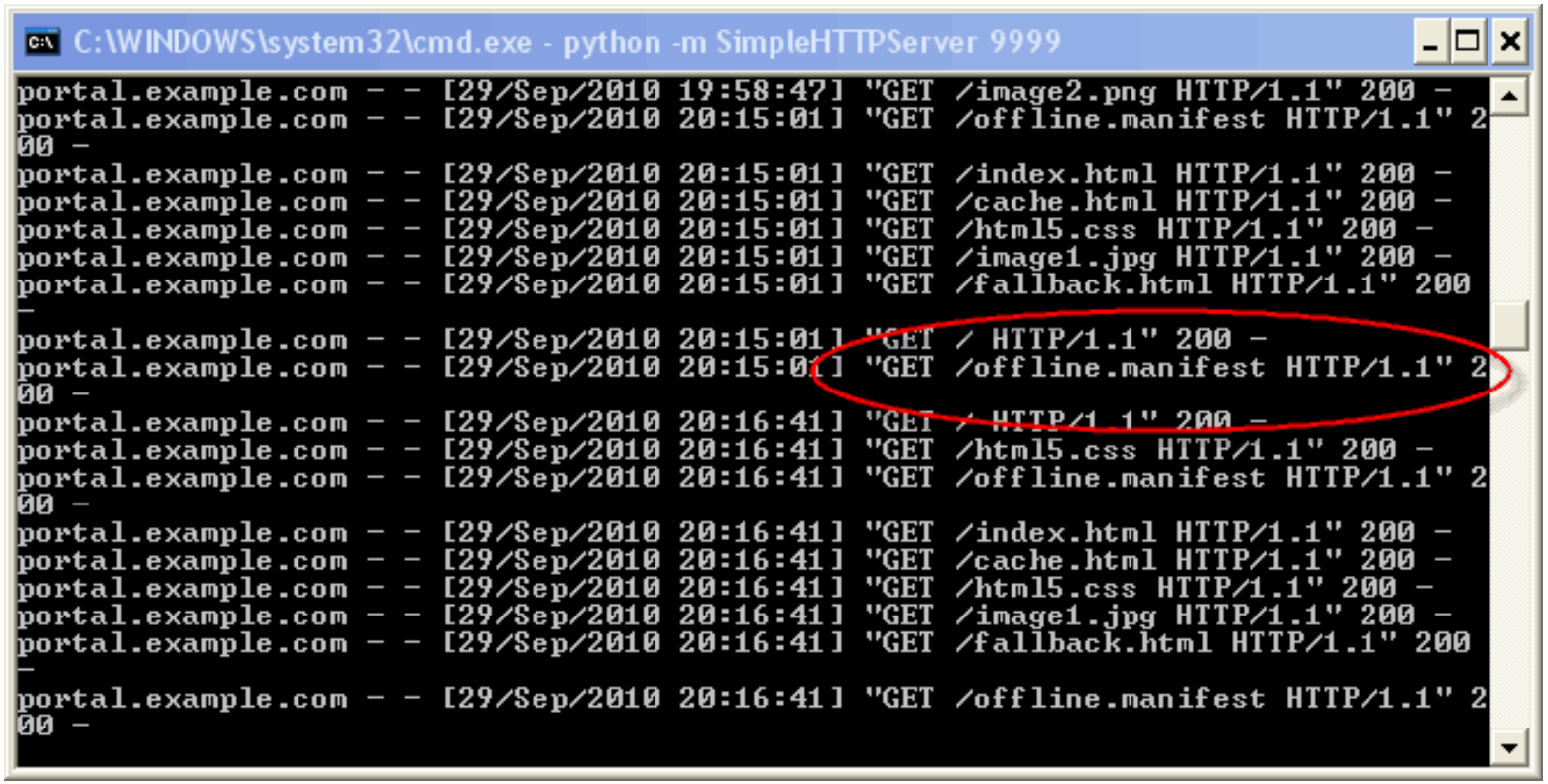




Best Practices

- Manifest errors are fatal (case sensitive entries)
- If you are adding and removing (lots of files) files, remember to update the manifest file
  - Use a predeployment script
  - Use a version Comment in the manifest file
- Host your site on different domain names
  - You can do this on your local machine by hacking the hosts file (see example on the next slide)
  - Windows: `\WINDOWS\system32\drivers\etc\hosts`
  - UNIX: `/etc/hosts`
- To see if files are requested, watch the server log

Checking Server Access



hosts File Hack for Local Testing

hosts file

```
# For example
127.0.0.1 localhost
127.0.0.1 offline0.example.com
127.0.0.1 offline1.example.com
```

Disk Quota

- Don’t assume success and check for errors
  - Example in Chrome:
    `Application Cache Error event: Failed to commit new cache to storage, would exceed quota`
- In the future, browsers will hopefully have graceful quota upgrade mechanisms like Opera’s for Local Storage

Works Well With HTML5 Web Storage…

JavaScript

```
if (navigator.onLine) {
  //Send updates to server
} else {
  window.localStorage.myLocalKey = 'Some Data';  }
}
```

Cache-As-You-Go

- If you add the manifest attribute, files will be added to the cache implicitly
- Subresources and dependencies may not be loaded properly (CSS, JS, etc.)

Manifest File

```
CACHE MANIFEST

FALLBACK:
/ /offline.html

NETWORK:
*
```

Offline Emulation

- HTML5 Gears Project by Brad Neuberg, Google: http://code.google.com/p/html5-gears/


QUESTIONS
ANSWERS

THANKS!

E-mail: peter.lubbers@kaazing.com

Twitter: @peterlubbers

LinkedIN: Peter Lubbers

Apress HTML5 Book: http://prohtml5.com

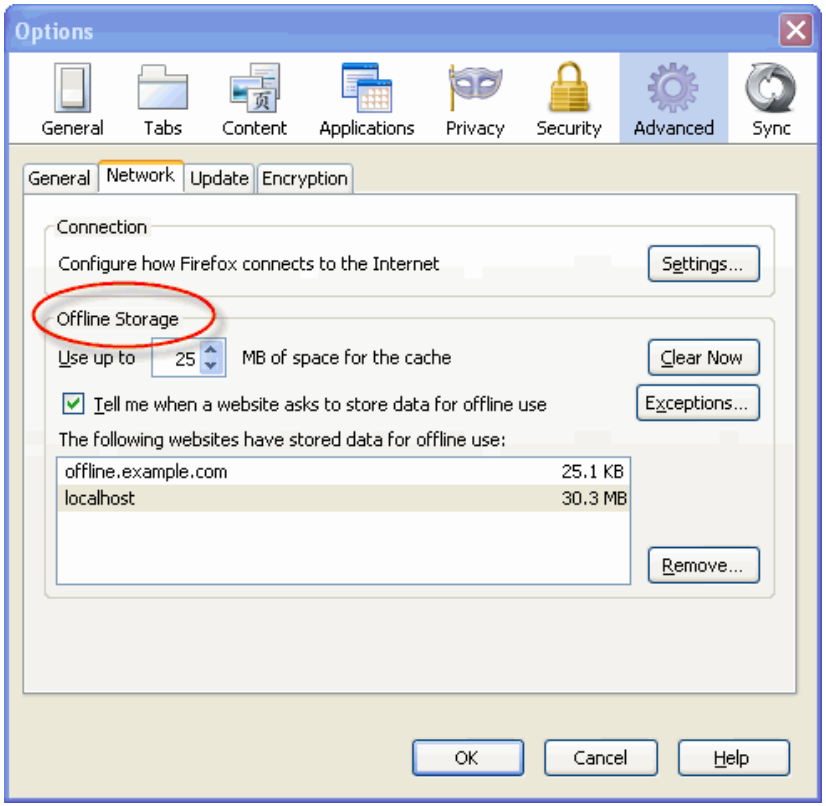
Book Coupon:
50% off Pro HTML5 e-book: **JFOKUSVTRU**

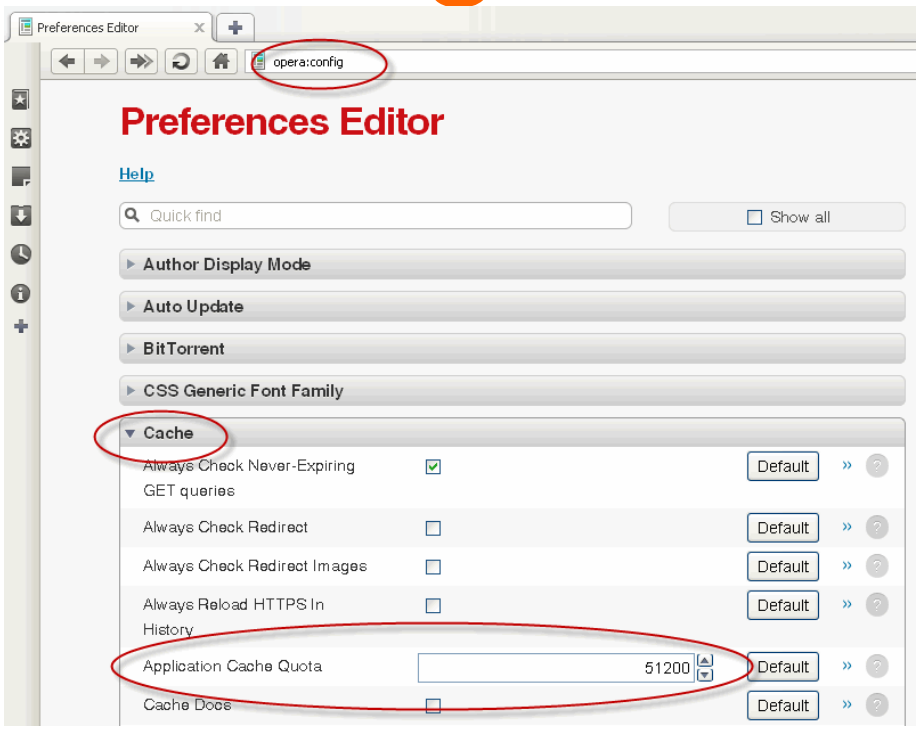
Resources

- ***WHATWG Offline Web Apps spec***: http://www.whatwg.org/specs/web-apps/current-work/multipage/offline.html#offline
- ***W3C Offline Web Apps spec***: http://dev.w3.org/html5/spec/offline.html#offline
- ***Offline example source files***: http://bit.ly/9pJ1Zq
- ***Pro HTML5 Programming***, Offline chapter, Peter Lubbers, Brian Albers, and Frank Salim
- ***Introducing HTML5***, Offline chapter, Bruce Lawson and Remy Sharp

Appendix: Browser Offline Cache Settings and Clearing the Cache

Offline Cache Settings

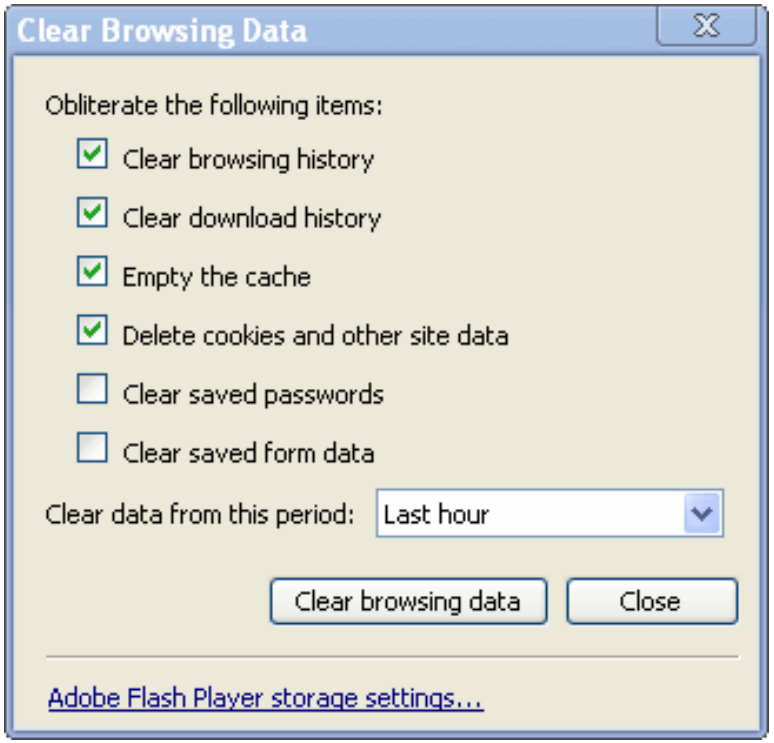




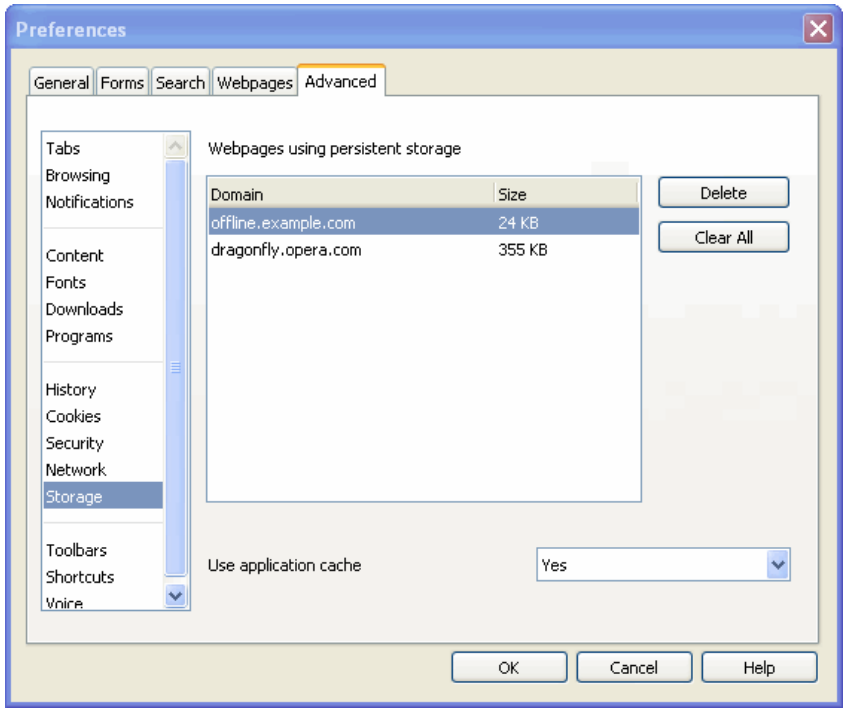
Firefox: Tools > Options (Preferences on Mac OS X) > Advanced > Network

Opera: `opera:config`

Clearing the Cache

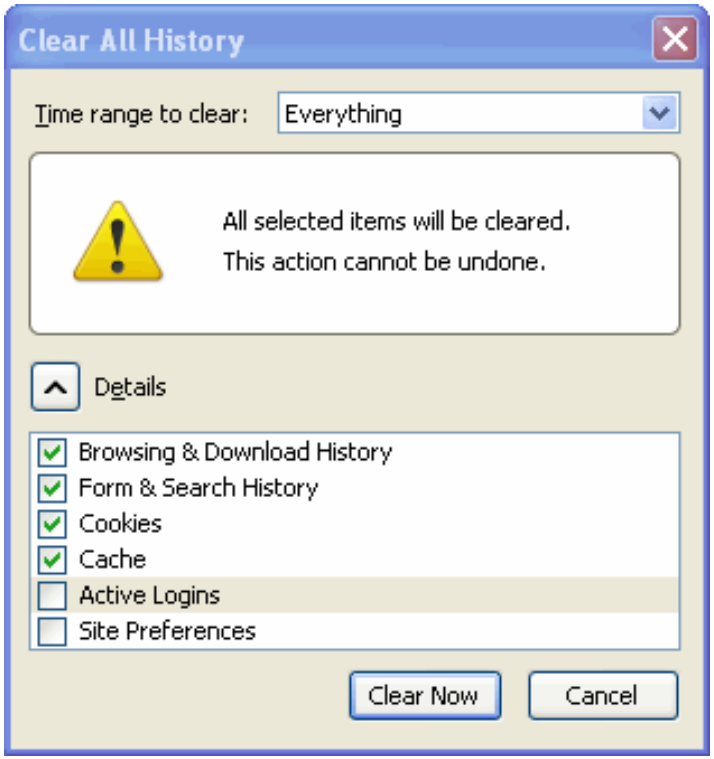


Chrome:
Settings Menu > Tools > Clear Browsing Data

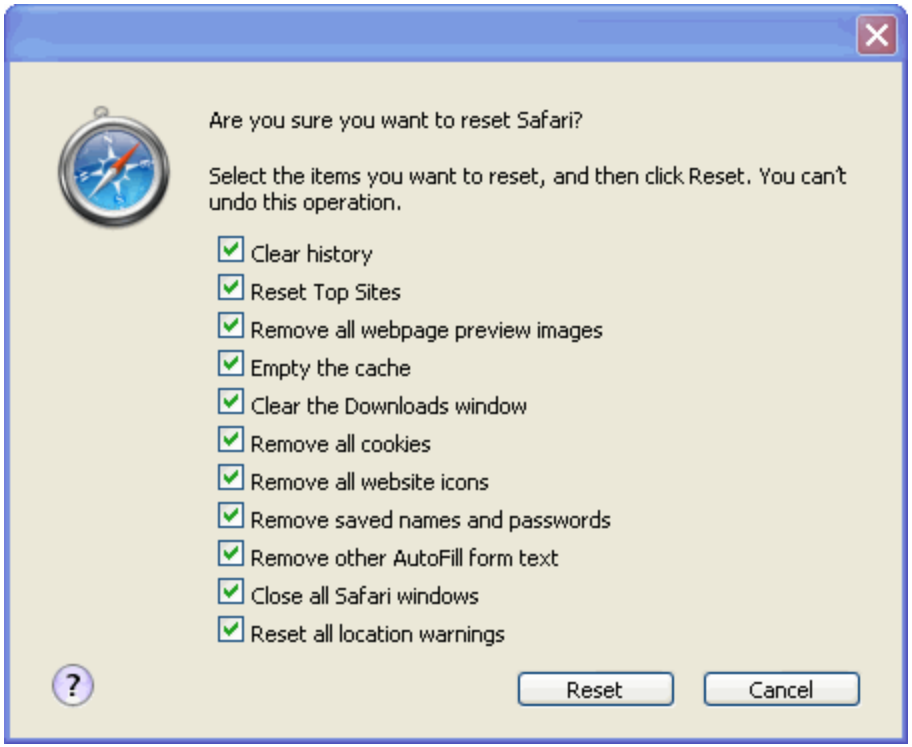


Opera:
Tools > Preferences > Storage

Clearing the Cache



Firefox:
Tools > Clear Recent History



Safari:
Settings Menu > Reset Safari

Clearing the Cache

Browser	Steps to Clear the Cache
Chrome	**Settings Menu > Tools > Clear Browsing Data**
Firefox	**Tools > Clear Recent History (and Tools > Options (Preferences on Mac OS X) > Advanced > Network > Remove for app cache)**
Safari	**Settings Menu > Reset Safari**
Opera	**Tools > Preferences > Storage (+ Tools > Clear Private Data)**
Internet Explorer	**Coming soon…**

Note: Close any offline pages before you do this to avoid problems

KAAZING