HTML5 Application Cache

* 1. “Offline Apps” concept enables the browser to load the web page from the server for the first time and store it in the browser memory; but second time onwards, it loads the page from the browser memory directly, instead of loading it from the server.
  2. • **Advantage:** The user can view the web page, even without network connection.

**Steps for development of Offline Apps**

**------------------------------------------------------------**

* • Create a “manifest” file:

**filename.appcache**

**--------------------------------------------------------------**

**CACHE MAIFEST**

**#101**

**Javascript.js**

**StyleSheet1.css**

**Img1.jpg**

**------------------------------------------------------------------**

* • Set the “manifest” file to the html file:
  1. <html manifest=”filename.appcache”>

**Example on Offline Apps**

* **c:\html\StyleSheet1.css**

**body{**

**font-family:Tahoma;**

**font-size:16px;**

**}**

**c:\html\JavaScript.js**

**function fun1(){**

**alert(“hi”);**

**}**

**c:\html\sample.appcache**

**CACHE MAIFEST**

**#101**

**Javascript.js**

**StyleSheet1.css**

**Img1.jpg**

**AppCache** is specifically designed to allow web apps (and web sites) to be made available offline, though the same speed benefits which the normal browser cache provides, when the user is online, are also provided by AppCache.

The key difference with the **Browser cache** is that you can specify all the assets the browser should cache in a manifest file (conceivably your entire site) whereas the browser cache will only store the pages (and associated assets) you have actually visited.

## **Application Cache**

[Offline / Storage](https://www.chromestatus.com/features/6192449487634432)

AppCache was standardized in the Offline Web applications section of the HTML specification. The standard is formally referred to as application caches. New Web applications should be built around Service Workers. Existing applications that use AppCache should migrate to Service Workers. AppCache access was removed from insecure origins in M70. This intent addresses AppCache usage in secure origins.

### **Motivation**

First and foremost, AppCache is a deprecated standard with serious architectural concerns. Second, Chrome's AppCache implementation is a security and stability liability. AppCache is documented as deprecated and under removal in MDN and in the WHATWG standard, and marked as obsolete in W3C’s HTML 5.1. It is incompatible with CORS, making it unfriendly for usage with CDNs. Overall, AppCache was changed in over 400 Chromium CLs in 2018-2019. It has imposed a tax on all of Chrome’s significant architectural efforts: Mojofication, Onion Souping, and the Network Service. The security benefits of the removal are covered under Security Risks.