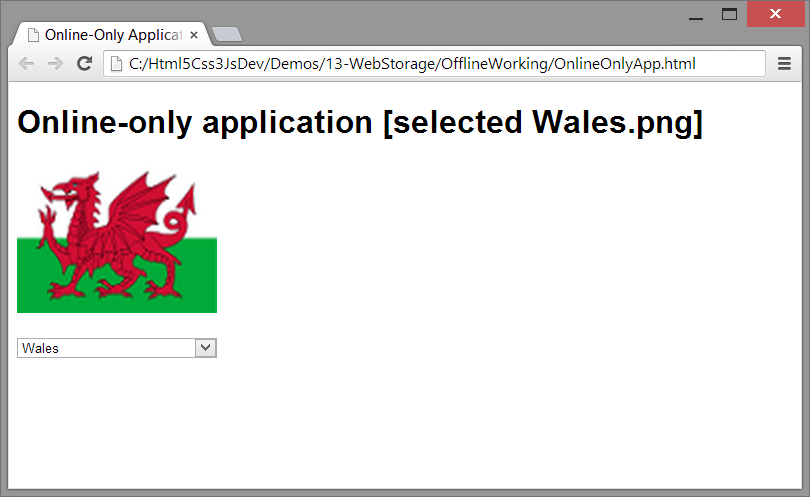
Demonstration instructions

**Demo\_OfflineWorking.docx**

This demonstration shows how to use application caching in HTML5. There are two HTML documents in this folder, which are fairly similar to each other at first glance:

* **OnlineOnlyApp.html**This page allows you to choose images from a drop-down list box. You can run this file from the file system (i.e. just open the file in the browser) and it all works ok – you can choose an item from the drop-down list box and it displays the country's flag at the top of the web page:  
    
    
  Now take a look at the source code. Notice that the images are not actually part of the DOM tree – i.e. they're not defined as **<img>** elements in the web page. This means the browser doesn’t know about the image file names until the user actually selects an item from the drop-down list box. At that moment, the browser will send an HTTP request back to the server in real time, to download the image JIT. This works fine if the browser still has a connection to the server, but it will fail if there's no connection available (e.g. if you're sitting in a train tunnel or on a plane sipping a cool beer).
* **OnOfflineApp.html**This page is almost identical to **OnlineOnlyApp.html**, except that the **<html>** tag has a **manifest** attribute as follows:  
   **<html manifest="countries.appcache">**This specifies an HTML5 application cache. Take a look at the **countries.appcache** file – it tells the browser to cache various image files (but not all of them – for demonstration purposes, we've excluded Spain.png, USA.png, and Wales.png from the application cache).

We'll describe how to test these pages in this demo. We'll use IIS as the server, and we'll forcibly switch stop the server at strategic moments to see what happens.

**Setting things up**

There are several set-up steps you need to perform to prepare the environment for the demo. Follow these steps:

1. Copy all the demo files to IIS, as follows:
   1. Open File Explorer and go to **C:\inetpub\wwwroot**.
   2. Make a new sub-folder named **OfflineWorking**.
   3. Copy all the demo files into this folder.
2. Convert the **C:\inetpub\wwwroot\OfflineWorking** folder into an IIS application as follows:
   1. In the Windows Start menu, type **inetmgr** to bring up the IIS Manager window.
   2. In the left hand side of the IIS Manager window, expand your machine name, then expand **Sites**, then expand **Default Web Site**. Right-click **OfflineWorking** and click **Convert to Application**. Accept all the default options and click **OK**.
3. Enable IIS to be able to return static content (if you don't do this, you might just get empty web pages when you request HTML files). Follow these steps:
   1. Open Control Panel, and click **Programs and Features**.
   2. In the Programs and Features window, click **Turn Windows Features on or off**.
   3. In the Windows Features dialog box, expand **Internet Information Services**, then expand **World Wide Web Services**, and then expand **Common HTTP Features**. Ensure that the **Static Content** check box has a "check" mark inside it, and then click **OK**.
4. You must tell IIS what to do when it sees the **<html manifest="countries.appcache">** element in the HTML page. Specifically, you must associate the **.appcache** file extension with a MIME type, so IIS knows how to interpret the file extension. Follow these steps:
   1. Open a Command Prompt window as an Administrator.
   2. Change directory to the following directory:  
       **C:\Windows\System32\inetsrv**
   3. Run the following command (all on one line). This command associates the **.appcache** file extension with the **text/cache-manifest** MIME type: **appcmd set config /section:staticContent   
       /+"[fileExtension='.appcache',mimeType='text/cache-manifest']"**
5. Restart IIS to ensure all these changes have taken effect, as follows:
   1. In IIS Manager, in the left hand side of the window, select your machine name.
   2. In IIS Manager, in the **Actions** section on the right-hand side of the window, click the **Restart** link.

**Running the OnlineOnlyApp.html web page**

In this part of the demo, you'll try out the **OnlineOnlyApp.html** web page. You'll find that it works fine when there's a connection available to the server, but fails when there's no connection.

Follow these steps:

1. Open a browser window and enter the following URL:  
    <http://localhost/OfflineWorking/OnlineOnlyApp.html>
2. In the web page, in the drop-down list box, select Brazil. It should display the Brazilian flag.
3. Now select China. It should display the Chinese flag.
4. Stop the IIS server (i.e. in IIS Manager, click the **Stop** link in the **Actions** section).
5. Back in the web page, in the drop-down list box, select Greece. This should fail now, because there's no connection to the server – i.e. the browser sent a real-time request for Greece.png to the server, but the server wasn't available. This identifies the need for HTML5 application caching!

**Running the OnOfflineApp.html web page**

In this part of the demo, you'll try out the **OnOfflineApp.html** web page. You'll find that it works fine even when there's no connection available to the server, thanks to application caching.

Follow these steps:

1. Start IIS server (i.e. in IIS Manager, click **Start**).
2. In the browser window, enter the following URL:  
    <http://localhost/OfflineWorking/OnOfflineApp.html>
3. In the web page, in the drop-down list box, select Greece. It should display the Greek flag (no surprises here, this would have worked anyway because the server is still "up").
4. Stop IIS server (i.e. in IIS Manager window, click **Stop**).
5. Back in the web page, in the drop-down list box, select Norway. It should display the Norwegian flag. This is application caching in action – this .png file was preloaded into the application cache when the HTML page was first loaded, so that the .png file is available even when the server is unavailable.
6. In the drop-down list box, select Spain, USA, or Wales. It should show the "fallback" image (i.e. an orange diamond with a lightning bolt). This is a deliberate feature in this demo, to demonstrate the purpose of the fallback setting in the application cache manifest file. We deliberately didn't include these .png files in the application cache, but instead specified the fallback image for any file requests that weren't preloaded in the cache.

That concludes our demonstration. We've seen how to define and apply an application cache file, and we've seen the effect of application caching when the server (IIS) is unavailable.

**One last thing: start IIS again (i.e. in IIS Manager, click the Start link).**