

Xap StartHere

Note: download coming soon...

1. Downloading the XAP zip file

Download the ZIP or TAR versions of the file from the [XAP Apache site](#).

2. Unpacking to your web server document directory

Locate the document root for your web server. If you do not have a web server, you can download one for free from <http://httpd.apache.org/>. The document root for Apache web servers is indicated in the `conf` file with the `DocumentRoot` variable, and is called `htdocs` by default. For IIS the document root is under `INetPub\wwwroot`.

Unpack the contents of the downloaded file into a directory under your server document root.

For more information consult the documentation for your web server.

3. Running the initial example

Open up your web browser and navigate to the directory where you unpacked the archive. You can typically access the server on your machine using the URL <http://localhost/>. Locate `ajax-index.html` (in either `dist`, `samples`, or your root directory) and run it your browser. This is a basic XAP application. There should also be `html` `README` files in these directories.

Important Notes

Image Caching and IE 6

The dynamic user interface powered by Ajax can require the frequent use of images; this involves getting and setting images rapidly on the fly, and works better if any repeatedly used images are locally cached. IE 6 does not handle image caching well - it does not check the cache for previous-loaded images, instead getting the image from the server every time. This is a well-known bug affecting any application using images (see, for example, [here](#), or [search for the terms "IE image cache"](#)). The workaround is to configure the web server not to add an "expires" HTTP header.

Cached XAL files

XAL files can be cached by any of the popular browsers. By contrast with the case of images under IE 6, this can be *more* caching than we'd want, because in this case refreshing your application page will not get the latest copy of a XAL file, even if it's been edited in the meantime. As with image caching, this behavior can be controlled by the web server setup. Another way would be to add an additional header to the page, which is possible if you're using a server side scripting technology such as JSP or PHP. In PHP this can be done by calling: `header ('cache-control: no-cache');`

Deploying XAL files to IIS

IIS does not recognize files with a `.xal` file extension and does not send them by default. This can be worked around by setting a mime type for XAL files to be `text/xml`; another workaround is to rename the files with the extension `.xml`. This behavior is not found with the Apache web server or with any application server.