
File URL scheme

Autor:

Data de publicació: 24-01-2014

File-URL-scheme

File:///d:CmpD_St_mydocdown_Dst0DownloadsGrandezas_de_Tarragona.pdf#page=536 File:///d:CmpD_St ebay historyOertling-MA1984.pdf#page=42 File:///d:CmpD_St_mydocdown_Dst0DownloadsOertling-Bilge-pump.pdf#page=42
File:///d:CmpD_St_mydocdown_Dst0DownloadsA_descriptive_and_historical_account_of.pdf#page=241
File:///d:CmpD_St_mydocdown_Dst0DownloadsCrisi_de_Cataluña_hecha_por_las_nacione.pdf#page=87
File:///d:CmpD_St_mydocdown_Dst0DownloadsCrisi_de_Cataluña_hecha_por_las_nacione.pdf#page=91 http://www.google.es/search?q=burr+pump&hl=ca&tbm=bks&tbm=bks&gs_l=heirloom-serp.12...1761466.1775889.0.1778488.6.6.0.0.0.0.191.765.0j6.6.0...0.0...1c.1.ZbcMsz4e_T0&oq=burr+pump http://books.google.es/books?id=FoVkfkbV1_8C&pg=PA16&dq=Mariano+Jacopo+Taccola&hl=ca#v=onepage&q=Mariano%20Jacopo%20Taccola&f=false

File URL scheme The file URL scheme is a URL scheme specified in RFC 1630 and RFC 1738, typically used to retrieve files from within one's own computer.

Format

A file URL takes the form of

file://host/path

where host is the fully qualified domain name of the system on which the path is accessible, and path is a hierarchical directory path of the form directory/directory/.../name. If host is omitted, it is taken to be "localhost", the machine from which the URL is being interpreted. Note that when omitting host you do not omit the slash ("file:///foo.txt" is okay, while "file://foo.txt" is not, although some interpreters manage to handle the latter).

Meaning of slash character

The slash character (/), depending on its position, has different meanings within a file URL.

The // after the file: is part of the general syntax of URLs. (The double slash // should always appear in a file URL according to the specification, but in practice many Web browsers allow you to omit it)

The single slash between host and path is part of the syntax of URLs.

And the slashes in path separate directory names in a hierarchical system of directories and subdirectories. In this usage, the slash is a general, system-independent way of separating the parts, and in a particular host system it might be used as such in any pathname (as in Unix systems).

Examples

Unix

Here are two Unix examples pointing to the same /etc/fstab file:

file://localhost/etc/fstab

file:///etc/fstab

Windows

Here are some examples which may be accepted by some applications on Windows systems, referring to the same, local file c:WINDOWSclock.avi

file://localhost/c:/WINDOWS/clock.avi
file:///c:/WINDOWS/clock.avi
file://localhost/c:/WINDOWS/clock.avi

Here is the URI as understood by the Windows Shell API:[1]

file:///c:/WINDOWS/clock.avi

Things to consider

Windows

On Microsoft Windows systems, the normal colon (:) after a device letter has sometimes been replaced by a vertical bar (|) in file URLs. This reflected the original URL syntax, which made the colon a reserved character in a path part.

Since Internet Explorer 4, file URLs have been standardized on Windows, and should follow the following scheme. This applies to all applications which use URLMON or SHLWAPI for parsing, fetching or binding to URIs. To convert a path to a URL, use `UriCreateFromPath`, and to convert a URL to a path, use `PathCreateFromUrl`. [1]

To access a file "the file.txt", the following might be used.

For a network location:

file://hostname/path/to/the%20file.txt

Or for a local file, the hostname is omitted, but the slash is not (note the third slash):

file:///c:/path/to/the%20file.txt

This is not the same as providing the string "localhost" or the dot "." in place of the hostname. The string "localhost" will attempt to access the file as `localhost:pathtothe file.txt`, which will not work since the colon is not allowed in a share name. The dot "." results in the string being passed as `.c:pathtothe file.txt`, which will work for local files, but not shares on the local system. For example `file:///./sharename/path/to/the%20file.txt` will not work, because it will result in `sharename` being interpreted as part of the DOSDEVICES namespace, not as a network share.

The following outline roughly describes the requirements.

The colon should be used, and should not be replaced with a vertical bar for Internet Explorer.

Forward slashes should be used to delimit paths. note that while DOS requires backslashes, the Win32 API will accept either a forward or back slash.

Characters such as the hash (#) or question mark (?) which are part of the filename should be percent-encoded.

Characters which are not allowed in URIs, but which are allowed in filenames, must also be percent-encoded. For example, any of "{}^" and all control characters. In the example above, the space in the filename is encoded as %20.

Characters which are allowed in both URIs and filenames must NOT be percent-encoded.

Must not use legacy ACP encodings. (ACP code pages are specified by DOS CHCP or Windows Control Panel language setting.)

Unicode characters outside of the ASCII range must be UTF-8 encoded, and those UTF-8 encodings must be percent-encoded.

Use the provided functions if you can. If you must create a URL programmatically and you cannot access SHLWAPI.dll (for example from script, or another programming environment where the equivalent functions are not available) the above outline will help.

Legacy URLs

To aid the installed base of legacy applications, the `PathCreateFromUrl` recognizes certain URLs which do not meet these criteria, and treats them uniformly. These are called "legacy" file URLs as opposed to "healthy" file URLs. [2]

In the past, a variety of other applications have used other systems. Some added an additional two slashes. For example, `emotehostsharedirfile.txt`, would become `file:///remotehost/share/dir/file.txt` instead of the "healthy" `file://remotehost/share/dir/file.txt`.

Web pages

File URLs are rarely used in Web pages on the Internet, since they make the assumption that such a file exists on the client's computer. The host specifier can be used to retrieve a file from an external source, although no specific file-retrieval protocol is specified; and using it should result in a message that informs the user that no mechanism to access that machine is available.

Browser behaviour

Common browsers

Google Chrome

use extension "LocalLinks"[2]

Microsoft Internet Explorer

Add site to Trusted Sites (Internet Options->Security-> Trusted Sites).[3]

Mozilla Firefox

within directory c:\Users<<USERNAME>>\AppData\Roaming\Mozilla\Firefox\Profiles<<***>>\default[4] create file user.js which contains the following:[5]

```
user_pref("capability.policy.localfilelinks.checkloaduri.enabled", "allAccess");
user_pref("capability.policy.localfilelinks.sites", "http://intranet http://wiki http://bugzilla HTTP://EXAMPLE");
user_pref("capability.policy.policynames", "localfilelinks");
```

or use addon "Local Filesystem Links"[6]

Opera

click right – "save..."

Other browsers

The original Web browser, WorldWideWeb, provided editing of resources in file space [3]. Amaya still has this ability.

References

- Jump up ^ Risney, Dave (2006). "File URIs in Windows". IEBlog. Microsoft Corporation. Retrieved 31 July 2013.
- Jump up ^ <http://chrome.google.com/webstore/detail/locallinks/jllpkdkcdjndhggodimiphkghogcpida>
- Jump up ^ http://answers.microsoft.com/en-us/ie/forum/ie8-windows_xp/after-recent-ms-update-can-no-longer-open-local/46bdd620-196f-453e-9f4f-f4a92aad6c2a?msgId=6438de4f-d8ff-4feb-8ba6-8eab5524e659
- Jump up ^ http://support.mozilla.org/en-US/kb/profiles-where-firefox-stores-user-data#w_how-do-i-find-my-profile
- Jump up ^ http://kb.mozillazine.org/Links_to_local_pages_do_not_work
- Jump up ^ <http://addons.mozilla.org/de/firefox/addon/local-filesystem-links>