10. Multilingual Web Addresses

Multilingual Web Addresses

Presently the Web addresses are typically expressed using Uniform Resource Identifiers or URIs. The URI syntax defined in RFC 3986 STD 66 (Uniform Resource Identifier (URI): Generic Syntax) restricts Web addresses to a small number of characters. These characters are upper and lower case letters of the English alphabet, European numerals and a small number of symbols.

The original reason for this was to aid transcribability and usability both in computer systems and in non-computer communications, to avoid clashes with characters used conventionally as delimiters around URIs, and to facilitate entry using those input facilities available to most Internet users.

With the development of Multilingual Computing, user demands to use characters from his/her native language in Web addresses also. A Web address in one s own language and alphabet is easier to create, memorise, transcribe, interpret, guess, and relate to. It is also important for brand recognition. This, in turn, is better for business, better for finding things, and better for communicating.

Recent developments in the area of Internationalized Domain Names activity begin to make this possible. Imagine, how easy it would be for you to remember and type the web address such as shown below in your own language.

http://भारतीयभाषा.टीडीआईएल.भारत

Basics of Multilingual Web Addresses:

Web addresses that allow the use of characters from a wide range of scripts are known as Internationalized Resource Identifiers or IRIs. There are four main requirements for IRIs to work:

1. The syntax of the format where IRIs are used (eg. HTML, XML, SVG, etc) must support the use of non-ASCII characters in Web addresses.
2. The application where IRIs are used (eg. browsers, parsers, etc.) must support the input and use of non-ASCII characters in Web addresses.
3. It must be possible to carry the information in an IRI through the necessary protocol (eg. HTTP, FTP, IMAP, etc.)
4. It must be possible to successfully match the string of characters in Web address against the name of the resource being pointed to on the file system or registry where it is stored.

Various document formats support IRIs. Examples include HTML 4.0, XML 1.0 system identifiers, the XLink href attribute, XMLSchemas anyURI datatype, etc.

Since most of the protocols developed, using ASCII, many protocols do not allow IRIs to pass through unchanged. Typically they require that the address be specified using the ASCII characters defined for URIs. There are well specified ways to do this conversion process.

The last requirement for IRI to work properly requires that a string of characters be matched against a target whether or not those characters are represented by the same encoding. This is achieved by using UTF-8 encoding. he following fictitious Web address will be used to explain the concept.

http://	भारतीयभाषा.टीडीआईएल.भारत.in/	dirA/आईडीएन.html
Scheme	Domain Name in Hindi	Path

This is a simple IRI that is composed of three parts.

The http:// contains information about the scheme to be used. Non-ASCII characters are not used here. The next part, भारतीयभाषा.हिन्दी.in/, is the domain name. The rest of the address is a path that indicates the actual location of the resource being pointed to from the server root.

The domain name भारतीयभाषा.हिन्दी.in/ is read as BhartiyaBhasha.Hindi (meaning Indian Language) dot Hindi (meaning Hindi Language) dot in (Indian country code). The path reads dirA slash IDN dot html .

When it comes to dealing with requirements two to four above, there is one solution for the domain name and a different solution for the path.

Domain names are allocated and managed by domain name registration organizations spread around the world. A standard approach to dealing with multilingual domain names was agreed by the Internet Engineering Task Force (IETF). It is defined in RFCs 3490, 3491, 3492 and 3454, and is based on Unicode 3.2.

The domain name registrar fixes the list of characters that people can request to be used in their country or top level domains. However, when a person requests a domain name using these characters they are actually allocated the equivalent of the domain name using a representation called Punycode. Punycode is a way of representing Unicode code points using only ASCII characters.

The multilingual web address is typed in the users (requester s) address bar using the relevant native characters. When user clicks on the link, the browser needs to convert any native script characters in the web address to punycode representations.

The user clicks on a hyperlink or enters the IRI in the address bar of the browser. At this point the IRI contains non-ASCII characters that could be in any character encoding. Here is the domain name that appears in the example above.

भारतीयभाषा.हिन्दी.in/

If the string that represents the domain name is not in Unicode, the user agent converts the string to Unicode. It then performs some normalization functions on the string to eliminate ambiguities that may exist in Unicode encoded text.

Normalization involves such things as converting uppercase characters to lowercase, reducing alternative representations (eg. Converting ड+़ to ड़, etc.), eliminating prohibited characters (eg. spaces), etc.

Next, the user agent converts each of the labels (ie. pieces of text between dots) in the Unicode string to a punycode representation. A special marker (xn   ) is added to the beginning of each label containing non-ASCII characters to show that the label was not originally ASCII. The web address in the example now becomes:

xn   h2brbhd9b4bebp.xn   j2bd4cyah0f.in

In the next step, the punycode is resolved by the domain name server into a numeric IP address. Finally the browser sends the request for the page. Since punycode contains no characters outside those normally allowed for protocols such as HTTP, there is no issue with the transmission of the address. This should simply match against a registered domain name.

Presently the top-level country codes, for example, the .in at the end of **भारतीयभाषा.हिन्दी**.in/ still have to be in Latin characters.

References:

www.w3.org