

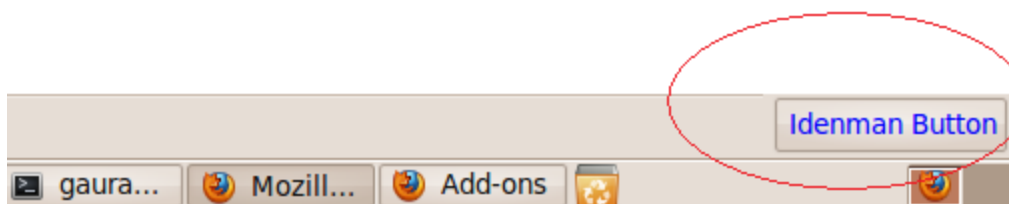
Idenman Progress Report (December 1, 2009)

Implementation of the Project involves development of the Library functions required at the Client as well as Server side level. At the Client side level, work involves development of a Firefox Plug-in which would interact with the User. At the Server side level, we have to implement a Shared Object (.so) Library. The .so file would be installed in the Apache web server and it will be responsible for interacting with the Firefox Plug-in and providing relevant service to it.

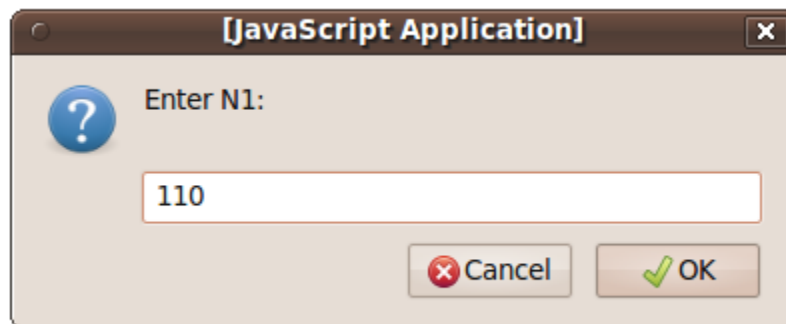
Initial Challenge was to develop the infrastructure for basic data-transfer between the Firefox Plug-in and the Apache Module.

I started to work on the Apache Web Server initially and understood how to make a simple module for it. Documentation was sparse on this topic and I took Binh's help to get around it. While implementing the library functions, I realized that the data would be passed from the Client side using either GET or POST functionality for which AJAX (XMLHttpRequest) would be required. I decided to handle GET for the time being. While using GET, URL Encoding at the browser level has to be handled for which I had to parse the data for ASCII characters which would be coming in from the Client side.

To test my functionality developed at the Server side, I implemented a Firefox Plug-in which would take User Input as required.



Upon click of the above button, a series of Dialog menu opens up to fetch information from the User.



```

/**
 * GET
 * Retrieve an URL synchronously (false) or asynchronously (true). Note that
 * this is synchronous, that is, it will block (stall) the browser
 */
function getURL(url, asynchronously)
{
    var xmlhttp;
    if(window.XMLHttpRequest)
    {
        xmlhttp = new XMLHttpRequest();
    }
    else
    {
        alert('Your browser does not support XMLHttpRequest');
        return;
    }
    xmlhttp.onreadystatechange = function() {
        if(xmlhttp.readyState == 0) {
            alert('Server is not setup');
            return;
        }
        if(xmlhttp.readyState == 4)
        {
            var response = xmlhttp.responseText;
            if(response.length > 0)
                alert('Server response: '+response);
            if(request.status != 200)
                throw("Unable to get URL " + url + " status code " + request.status);
        }
    }
    xmlhttp.open("GET",url,asynchronously);
    xmlhttp.send(null);
}

```

The above shown functionality is used for performing synchronous/asynchronous GET form submissions using AJAX. Similar functions have been implemented for POST submission as well.

An Example for URL for GET form submission is

```
var url = "http://localhost/idenman?N1="+N1+"&C1="+C1+"&C2="+C2;
```

Where N1, C1 and C2 would be the values fetched from the User while generating the Pseudonym.

Once the data is received at the Server end, the above mentioned URL has to be broken down into tokens and I save each of the entries in a Hash Table structure as <key,value> pair.

There are numerous Apache Portable Runtime (APR) Library Functions which I have used for various operations as per need. Also, there was a requirement for data to be sent to the Client side for which I used the `ap_rprintf()`. In this example, r and N2 have to be submitted to the Client side during generation of the Pseudonym.

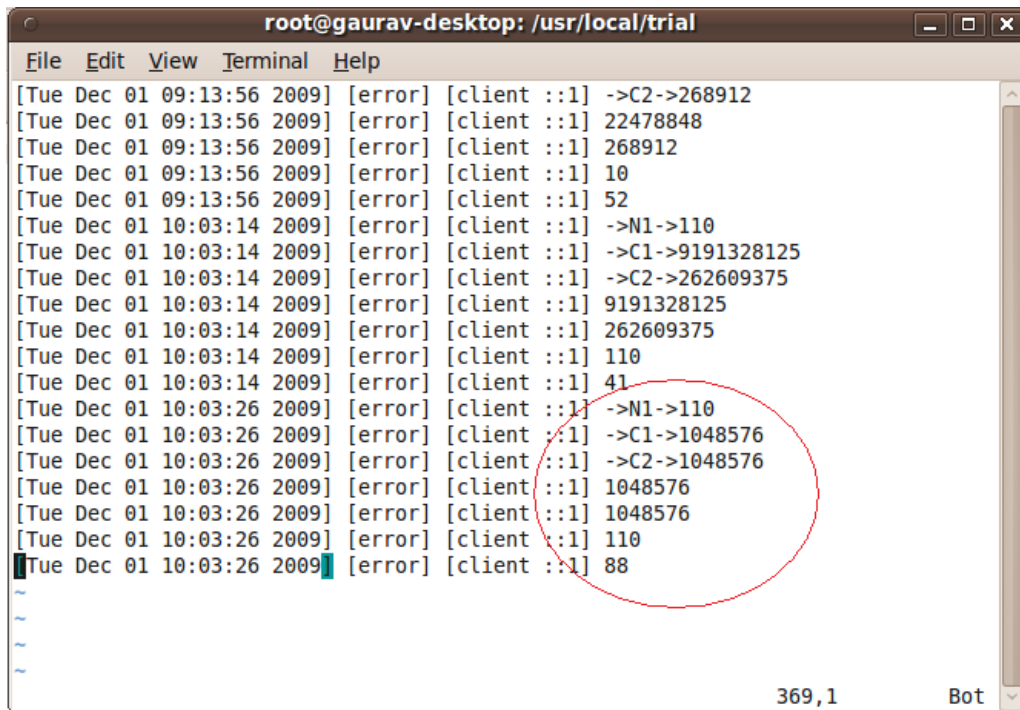
```
ap_rprintf(r, "r: %d\n",R);
ap_rprintf(r, "N2: %d\n",N2);
```

Once the data reaches the client side, the XMLHttpRequest provides the data through 'responseText' property. Sample is shown below:



So, the infrastructure for data transfer between the Apache Web Server and Client side Firefox Plug-in has been established properly.

The testing work at the Server part has been performed by passing debug messages to the Error Log.



Presently, I am working on the latter part of the Generation of Pseudonym functionality.

I have used dummy data at present for some functionality and range of numbers has not been set yet. I would like to meet Binh and discuss some of the doubts which I am coming across.

I have created a Google Code Repository for managing Apache Web Server compiled data at present. (Source Files has been attached)