Check List For Front End optimization-Xforms

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Area** | **Possibilities** | **Applicability** |
| **Add an Expires or a Cache-Control** **Header** | **server** | 1. **Content expiry headers** |  |
| **Gzip Components** | **server** | 1. **We server configurations** |  |
| **Minimize Http request** | **Content** | 1. **Combined files** 2. **CSS Sprites** 3. **Image maps** 4. **Inline images** |  |
| **Minify JavaScript and CSS** | **Content** | 1. **Use the script minified for js &css** 2. **Repack libraries** |  |
| **Make Ajax Cacheable** | **content** | 1. **To minimize the HTTP request** |  |
| **Post-load Components** | **content** | **For runtime performance after load.** |  |
| **Preload Components** | **content`** | **Unconditional preload**  **Conditional preload**  **Anticipated preload** |  |
| **Reduce the Number of DOM Elements** | **content** | 1. **Reduced number of DOM elements in the documents** 2. **Split complex forms to smaller ones and load on demand** |  |
| **No 404s** | **Content** | 1. **Remove unused references** |  |
| **Minimize the Number of iframes** | **content** |  |  |
| **Remove Duplicate Scripts** | **JavaScript** | 1. **Remove Unnecessary HTTP request and execution** |  |
| **Put Scripts at the Bottom** | **JavaScript** | **Script references will be added at the top,Script Tab content will be added at the bottom.** |  |
| **Make JavaScript and CSS External** | **Javascipt** |  |  |
| **Minimize DOM Access** | **JavaScript** | 1. **Cache references to accessed elements** 2. **Update nodes "offline" and then add them to the tree** 3. **Avoid fixing layout with JavaScript** |  |
| **Optimize Event Handlers** | **JavaScript** | 1. **reduce number of handlers on DOM elements with event delegation** |  |
| **Put Stylesheets at the Top** | **css** | **1.style sheets added at the top by default** |  |
| **Avoid Filters** | **css** |  |  |
| **Avoid CSS Expressions** | **css** |  |  |
| **Optimize Images** | **css** | 1. **Gif Optimization** 2. **Gif/jpeg tp png for better compression** 3. **Optimize pngs** |  |
| **Optimize CSS Sprites** | **css** | 1. **Arrange images horizontally** 2. **Use multi color sprites for filters** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Done |  | Changes not required |  |
| Yet to Do |  | Can be done if required |  |

1. **Content Expiry/Cache Control Headers**

Set content expiry on HTTP headers and the **max-age** directive of the **Cache-Control** HTTP header in server responses. This holds the information about the validity of document and persistence. If cached, content will be fetched from browser cache instead of firing an HTTP request until the expiry time is passed.

Content expiry can be set to either time of last modified or the client access time

*Note:By Default content expiry is enabled only for JavaScript files. This can be enabled for HTML and css files as well.*

Apache: <http://httpd.apache.org/docs/2.0/mod/mod_expires.html>

IIS : <http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/0fc16fe7-be45-4033-a5aa-d7fda3c993ff.mspx?mfr=true>

|  |
| --- |
| File: <installation> \components\webgateway\Apache\cordys\_apache.conf  <Directory "D:\CordysInstallation\fp1\_cu5/Web">  AllowOverride All  #AuthName "Cordys"  #AuthType SSPI  #SSPIAuth On  #SSPIAuthoritative On  Allow from all  ##Require valid-user  Anonymous anonymous  <IfModule mod\_expires.c>  ExpiresActive on  ExpiresDefault "access plus 1 month"  ExpiresByType text/css "access plus 1 month"  ExpiresByType image/\* "access plus 1 month"  ExpiresByType text/html "access plus 1 month"  ExpiresByType application/javascript "access plus 8 hours"  ExpiresByType application/x-shockwave-flash "access plus 1 month"  </IfModule> |
| C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf  Un-comment the line  LoadModule expires\_module modules/mod\_expires.so |

1. [**Gzip Components**](#GzipComponents)

* Note: Gzip compression is enabled in the Web server level.

1. **Minimize Http request**

* All the css files are combined to one file-common.css
* All Common JavaScript code is kept to one file. –common.js
* All Xforms are having one JavaScript file with the corresponding business logic.[*Round trip of this has to be analyzed and can be merged later to the corresponding Xforms if round trip time is more*]

1. **Minify JavaScript and CSS.**

* This can be done for deployment. We can use YUI compressor.

Note: *Disadvantage of this approach will be difficulty in maintaining code after deployment.*

***Online Minifcation tools***

<http://www.refresh-sf.com/yui/#output>

<http://yui.2clics.net/>

1. **Make Ajax Cacheable**

* This is for pages loads more than once and the script files will be cached. Content expiry for JavaScript files is already set. This will be done with review comments implementation

1. **Post-load Components**

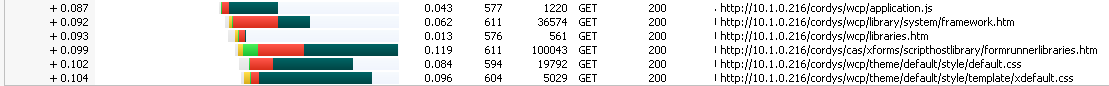
* For optimized performance of pages after the application loaded. Fire the requests in advance.

1. [**Preload Components**](#PreCatcher)

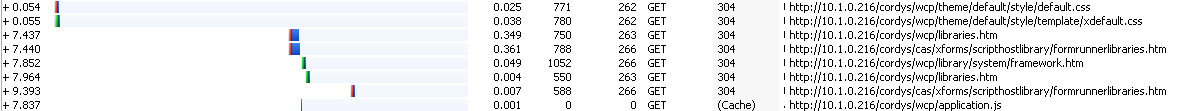
* Note:- Primary goal is to improve the loading time performance.

Pre Caching without content expiry will not avoid net work round trips .It still sends request to server and receives 304 status

Example: Preacher loads mandatory libraries in advance refer the screen shots for details



X form loading with pre cached elements(304 for html/css) Cache for js



Pre caching with content expiry will load the contents in no time.

**Note:-** The maintenance scenarios have to be considered. Example content modified in between the pre caching time. Preferably we can set 10 minutes content expiry from accessing time.

|  |
| --- |
| **Pre catcher code.**  **function catchURL(url,bSync)**  **{**  **var xmlhttp;**  **if (window.XMLHttpRequest)**  **{**  **xmlhttp=new XMLHttpRequest();**  **}**  **else**  **{**  **xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");**  **}**  **xmlhttp.open("GET",url,bSync);**  **xmlhttp.send();**  **}**  catchURL("/cordys/wcp/application.js",true);  catchURL("/cordys/wcp/theme/default/style/default.css",true);  catchURL("/cordys/wcp/theme/default/style/template/xdefault.css",true);  catchURL("/cordys/wcp/library/system/framework.htm",true);  catchURL("/cordys/wcp/libraries.htm",true)  catchURL("/cordys/cas/xforms/scripthostlibrary/formrunnerlibraries.htm",true); |

1. **No 404s**

* All Unused references must be removed (scripts and css files).This will be done with implementation of review comments. There are repeated references and unused references in for script and css files in the Xforms

1. **Remove Duplicate Scripts**

* Remove the repeated references from Xforms(script and css files).Also the repeated scripts. This will be done with implementation of review comments.

1. **Make JavaScript and CSS External**

* This is to avoid round trips in form which loaded more than ones. Content expiry must be set to get the better performance.

1. **Minimize DOM Access**

* For optimized performance of pages after load. Most of the optimizations will be done with code review comments.

1. **Optimize Event Handlers**

* For optimized performance of pages after load. Most of the optimizations will be done with code review comments.

1. **Avoid Filters**

* No filters used in the custom css files

1. **Avoid CSS Expressions**

* It has been removed from common.css after the review comments implementation. If style problem persists corresponding css classes to be tweaked.

1. **Optimize Images**

* Only background image for form body is used (logo2.jpg) which is optimized. Sprite.png for status Indicator is optimized.

1. **Optimize CSS Sprites**

* Done for sprite.png for status indicator.

**Xform server configurations for better performance**

Content Expiry is set to every time.If form not modified it will give 304/and will be read from cache.(nearly 2 sec advancement in performance).

**Check list for final results**

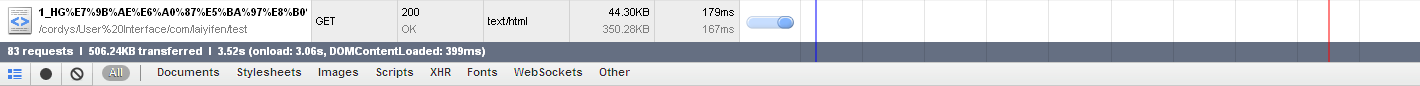
|  |  |  |
| --- | --- | --- |
| Review comments implementation(1&2)   * Indentation and comments removal * Caching of values in variables * Reusing the script |  | Runtime performance & script loading time |
| Enable content expiry for HTML , css & images at web server | Done |  |
| Use the pre catcher to load components in advance | Done | Load all js,htm,image content in advance |
| Content expiry on Xforms service container. | Done |  |
| Repack libraries.htm with required libraries |  | Since we pre cache it is not required. |
|  |  |  |

**Observations on Loading: [Fluctuations in network not giving identical results]**

First impression after the review comments implementation without preaching on the sample page with xform server configurations are as in the below figure. It gives better results in non-IE browsers (in chrome and Firefox pages are loaded in less than 4 seconds.). To get the proper results we need to analyze the application in the test server after the review comments.

**Chrome:**





**IE:**

****

****

**Further improvement points.**

After Implementations all requests are cached except the following and which could not be avoided.

* To Read user preferences and translation web services to read the dictionary file
* To get the xform
* To Validate the form
* To Get composite object.

Forms are relatively complex and complete rendering and data binding is happening on loading itself. The possibility for further improvements falls in to the following options

**Reduce the DOM load: -** Split the UI into smaller parts and load on demand. (The group Box contents can be loaded In different iframe as different forms)

**Reduce the JavaScript Execution time on load: -** Now data is binding on all controls on form load with get composite object method. Instead of this cache the data and the binding of data to individual models and refreshing views can be done on onexpand of group boxes. If we use the individual forms(previous method for reduce DOM load) to implement the same with composite object model ,the model object belongs to the group boxes has to be in the child form .To push data to model and refresh the views , model objects can be read from the window object of the child from using DOM navigation.

**Note: as per the discussions since it requires reworks on all the forms we have avoided these options. It can be implemented if further improvements are required sample forms are available in the test folder.**

**Code Snippets**

**Load Script Files at runtime**

|  |
| --- |
| var scriptTag = document.getElementsByTagName('head')[0].appendChild(document.createElement('script'));  scriptTag.setAttribute('type','text/javascript');  scriptTag.setAttribute('src', '/cordys/wcp/application.js'); |

**To stop back space action on document body:**

This should not be done since it is browser behavior.

|  |
| --- |
| function Form\_Init(eventObject)  {  cordys.addDOMListener(document.body,"onkeydown",stopBackSpace())  }  function stopBackSpace()  {  return function(eventObject)  {  var eventObejct = eventObject?eventObject:window.event;  if(eventObejct.keyCode == 8)  {  cordys.stopPropagation(eventObject);  cordys.preventDefault(eventObject);  }  }  }  function Form\_BeforeClose(eventObject)  {  cordys.removeDOMListener(document.body,"onkeydown",stopBackSpace())  } |

**Status Indicator**



**How to use**

To create the indicator

**Comment:** the following method will create status indicator in the xform inside any given htmlElement in the form. We need to pass three parameters html Element where the status indicator required. and the warm point and the hot point. Note:if not using the tool bar need to mention the corresponding UI element and position it accordingly.

|  |
| --- |
| **Syntax:**  var indicator = new StatusIndicator(htmlElement,warm\_point,hot\_point); */****/****to add the indicator to window so it will be available globally*  toolbar.ownerDocument.defaultView["Indicator1"]= indicator ;    **Example:**  var indicator = new StatusIndicator(toolbar*);//default points are70,90* toolbar.ownerDocument.defaultView["Indicator1"]= indicator ; |

To refresh the container with the status values use refresh method pass the  start date end date and current date in the following format.( **2011-12-16T09:30:00.0**)

|  |
| --- |
| **Syntax:**  Indicator.refresh(startDate,dueDate,currentDate);    **Example:**    Indicator1.refresh(“2011-12-16T09:30:00.0”, “2011-12-26T09:40:59.0”, 2011-12-25T09:31:01.0) |

To get the dates use

**Workflow.getTask()//***this will return all the task related information*

**Configuring**

If using in all files put the script to common.js and css to common.css

Uplod image **sprite.png**  and replace the url (**background-image:url(/cordys/battery/img/sprite.png);)**of the container in css with new image url.