Quios Internationalization & Co-branding

Process Specifications Document

Company Confidential

Produced for
Quios.com

Prepared by
Wes Chester, Jengis Alpar, Bryan Scholz

Version.1.3
8/22/00

For internal use only

Table of Contents:

Release History

Version	Author	Date
1.0	W. Chester, J. Alpar, A. Scholz	July 14, 2000
1.1	A. Scholz	7/31/00
1.2	A. Scholz	8/10/00
1.3	A. Scholz	8/22/00

Location and Dependencies

Document Location	http://intranet.quios.net/Docs/docs.shtml
Code Location	
Dependencies	

1. Architectural Overview and Motivation

This document has been written as a guideline for all Internationalization and Co-branding efforts within Quios. It is an architectural overview of the *Quios Tag System,* which describes Quios' proprietary Tag-based architecture for generating dynamic HTML content, as well as an overview of the system, and a step-by-step for the usage of standard HTML tagging in all matters of co-branding and localization.

When you are finished reading this document, if it serves its purpose, you should understand:

- **How the Quios standard HTML tagging system works**
- **What theQuios Tagging System is**
- **How we use them both to help us with our Internationalization and Co-branding efforts**
- **Where they affect our Web site in terms of files and Database**
- **How you should use them**
- **How you should not use them**

1.1 Motivation

The Quios Tags System and the standard HTML tagging system were designed to solve a very well defined set of related problems:

- **Internationalization**
- **Co-branding**

A side benefit, but not a primary goal of the *Quios Tag System,* one which must be used carefully, is that it allows:

- **Small, modular pieces of dynamic content logic appearing in many places using Perl tags.**

> *You should speak to your manager or an experienced Quios developer before adding any new Perl content tags, which actually store executable Perl instead of HTML: they are powerful but are more difficult to maintain and debug.*

Management at Quios has decided to take a very aggressive approach to the problems of Internationalization and co-branding: the goal is to allow actual switching of brands and languages dynamically at runtime, possibly even on a per-user basis. *Quios Dynamic Tags* were designed to solve this problem. This is an incredibly useful, powerful, and potentially profitable feature of the Quios system, but one that must be implemented with great care. This is not so much due to the inherent complexity of the feature as the fact that it must scale to handle a potentially large number of languages multiplied by the number of brands multiplied by the number of language- or brand-specific content entities; the product of these three is quickly growing to be a very large number.

2. Quios Tags 101

2.1 Brand / Language

The key is a two dimensional matrix with the core engine. All text, English or otherwise, must be in tags. We use an XML tag like system that sits in the HTML tags. In the tags table in our database is a matrix where only languages have to be translated or changed. This is also true of any pictures or features that would be potentially localized, as well as anything that is brand related. Any of these changes can be done online.

The two dimensions of the matrix are therefore language and brand. Every gif or text string or object has been set within a framework of:

- **default / default**
- **default / brand**
- **default / language**
- **brand / language**

This means that when the page is loading the server will know what to grab. For example, a Quios logo might be set in default/default, because it is the same regardless of the web site, and the engine knows to grab the default/default directory. Next the engine will know to grab at the language, because it is text on a French site, and then maybe brand because it is hosted by Excite.

2.2 Session

When you first enter a site your session "bag" is in a sense empty. First the language is determined, and then the brand. The server will look at the URL accessed as well as the language setting on the browser, and sign up specifications from Quios. Often this information is hidden in the form of an index session identifier in the URL.

For security's sake the information is encrypted. In the event that this is a second session id and a cookie of a registered user has been placed, you will be allowed to log on more quickly. For example:

/FBXYZL2/hme tmpl: the FB…. Is an index to the dam which allows you from page to page.

2.3 Languages

For the moment we can handle any single byte languages which in essence are the main European languages that make use of the Latin alphabet. When we migrate to a UTF database we will be able to work in any multi byte languages such as Arabic or Japanese.

2.4 Scalability

Everything is easily changed while maintaining site integrity. This is what is allowing us to roll out Excite in 7 languages in half as many weeks, and to form partnerships easily.

Our partners, not to lose sight of the big picture, are one of our main sources of revenue. The mere fact that we can get a partner a simple site up in a day might sometimes clinch a deal. If they want something more complicated that takes four weeks, we can do that too, but often a small partner does not want all the bother nor the associated costs.

2.5 What Does a *Quios Dynamic Tag* Look Like?

To start with something specific, here is how *Tags* appear in HTML:

```
<- NAME [Param_A|Param_B|Param_N] ->
```

That is the general form of a Quios Dynamic Tag. Here are some examples:

- **With no parameters:**
  ```
  <- PAGE_CONTENT_FOOTER ->
  ```
- **With one parameter:**
  ```
  <- TOP_MENU_BAR contact ->
  ```
- **With multiple parameters:**
  ```
  <- SELECT_HOUR MEASUREMENT_SYSTEM|hour|HOUR_VAL ->
  ```

2.6 What files do they affect?

At this time, Tags are used in files with the **.tmpl** and the **.mtpl** extensions. The **.tmpl** files at Quios are HTML with Quios Dynamic Tags in them; the **.mtpl** files are actual Perl programs that output HTML with Quios Dynamic Tags in them. Files with the extension **.tpl** almost always accompany **.mtpl** files. A file or output set may not have any tags, in which case it is treated as ordinary HTML. No other file extensions or output sets are subject to Tag Replacement.

2.7 What happens to Tags at output time: *Tag Replacement*

Tags are never served up to browsers directly, because they are not really HTML. They are replaced with the intended brand- and/or language-specific content in a process known as *Tag Replacement.* The appropriate brand/language is found in session info via $SESSION{Brand} and $SESSION{Language}. Sessions are how we keep track of Web users.

What happens with Tags in your HTML output is that special hooks into Apache are activated at a late stage in the Request cycle. These hooks do a regular expression search on the output, find all the tags, and replace them with the appropriate content. The hooks are in Quios::Apache::Tagger and Quios::Apache::PipeTagger.

The appropriate replacement follows two possible paths: HTML tags are mapped at Apache startup from a database table called TAG_TRANS and into an in-memory hash of references to anonymous subroutines.

There is no database activity generated by actual tag replacement, so performance is very good. The TagsTom.pm module determines how the tag is processed, using primary keys for Name, Brand and Language. These three columns uniquely identify a particular key from which the Data column is extracted and inserted into the HTML. See examples in the following table:

TAG_TRANS database table column names	**(example A)**	**(example B)**
NAME	HELLO_TAG	HELLO_TAG
F_BRAND_NAME	quios	quios
F_LANGUAGE_NAME	english_us	spanish
F_CODE_TYPE_NAME	html	html
F_TAG_TYPE_NAME	content	content
DATA	<P>hello</P>	<P>hola</P>

Table 1: Tag Translation table examples

Perl tags are pre-compiled and references to the compiled code are stored in the same in-memory hash. The code for each Perl tags lives in /usr/local/quios/lib/tags/. The file name becomes the Perl tag name, and the .plt extension stands for “Perl Tag”. The extension does not become part of the name. Should a name conflict arise between an HTML Tag and a Perl Tag, the tag replacement code in Quios::Apache::TagsTom will favor resolution of the HTML tag over the Perl tag.

2.7.1 Tag Replacement Work Flow

What really happens in the Tag Replacement workflow is something like this:

- At Apache startup time, all HTML tags are read from the database and all Perl tags are read from the file system. Both are wrapped into anonymous subroutines, references to which are stored in a big hash called %Quios::Apache::TagsTom::tagsref; this becomes an in-memory cache.

- When a Tag replaced file—extensions “tmpl” and “mtpl”—gets served up, two possible cases exist:

  - If the file is an .tmpl file, it is handled by a special handler in Quios::Apache::Tagger. This module reads the requested file, performs the tag replacment and then outputs the results to the brower.

  - If the file is an .mtpl file a stacked-handler chain is created consisting of Quios::Apache::PipeTagger and Apache::Registry. Quios::Apache::PipeTagger accumulates standard output from Apache::Registry and performs tag replacement on the complete contents after Apache::Registry has finished with the request. This means that any Perl print statements generated by an .mtpl file will be tag replaced. Please also be aware that calls to $r->print will bypass this mechanism entirely.

- The output is searched for tags, This is where the regexp search gets done. and the tags are accumulated in a hash.

- Each completely unique tag call (tag name and arguments) are processed in the order in which they found.

    - Arguments are parsed

    - The tagsref hash is searched for that tag using the name, brand, and language criteria. If the coderef is found, it executes and the results, are saved. The results will be the DATA column in the tag_trans table in the case of HTML tags, or the results of the Perl code executed at that moment.

    - The the first argument of the given Tag call is the exact string 'RECURSE' the the routine will recurse and call itself against the contents of the tag just evaluated. This allows for 'Tags within Tags' and is, like Perl tags, incredibly powerful and very easy to abuse.

> *Note: Quios::Apache::PipeTagger @ISA Quios::Apache::Pipeline. Quios::Apache::Pipeline is a base class modeled in a substantial way from the examples beginning on page 178 of the O'Rielly* Writing Apache Modules with Perl and C *(Eagle Book). The modifications allow filters to execute on the entire output of multiple writes to STDOUT in a single pass. Quios::Apache::Pipeline could serve as the basis for other filters should we decide that we need additional filters in the future.*

3. Tag Naming Conventions

The purpose of this process is to replace portions of each .tpl and .tmpl file with a flag to mark where the appropriate localized content for that section should be added. The intent is to dynamically add the appropriate language or images at the time a user is browsing our site. Please note that it is possible the file you are editing will already have a tag created for a section of text.

Note that standard HTML tags, due to Macintosh file naming restrictions, may only be 26 characters in length.

Currently we have three tag types. Content, Image and Font tags, each of which may also have a brand.

3.1 Content tags

Format: <- [<*BRAND_ID*>]_<*IMG_FILENAME*>_BODY_CONTENT_<*DESCIPTIVE_ID*> ->

<BRAND_ID> - Tags are named first by brand. Currently, if there is no brand it is assumed to be Quios.

<IMG_FILENAME> - Leave off the file extension.

<DESCIPTIVE_ID> **-** A unique identifier for the tag which should be described briefly.

3.2 Image tags

Format: <- [<***BRAND_ID***>]_ **IMAGE_<IMG_FILENAME>**_[<***SRC***>]->

<BRAND_ID> - Tags are named first by brand. Currently, if there is no brand, it is assumed to be Quios.

<IMG_FILENAME> - Name of the image file. If the filename contains two or more words, separate them with an underscore.

Ex: IMAGE_BT_ON MSN_HOME_BODY_CONTENT_SIGN_UP

<SRC>If the image is used as an INPUT button, add _SRC to the tag name

Ex: IMAGE_BT_ON_SRC

Note: image tags that are flagged SRC should be used in the following manner:

<INPUT TYPE=IMG <- MSN_IMAGE_BUTTON_ON_SRC -> ALT="click here" >

Note: Only the HEIGHT WIDTH SRC and ALT attributes may only be used within in a tag of this type.

3.3 Global Tags

Global tags are used for text, not for images. They are also branded with among others MSN and HOTBAR.

Some examples are:

<- GLOBAL_HELP ->

<- MSN_GLOBAL_PERSONAL_PROCESS ->

3.4 Tag Extensions For each File Type

.tmpl: indicates a static page with HTML and localization tags.

.tpl: indicates a template page to be used with HTML::Template PERL module. It includes HTML, Input Type and HTML::Template tags

.mtpl: PERL script without any tags.

.plt: PERL backend script.

4. Tools and Configuration for Standard HTML

Both Mac and PC tools consist of a browser and a text editing tool. It is important to remember that all codes must be checked on both platforms.

4.1 PC

- Netscape 4.7 and IE. 5.0.
- We use HomeSite as our Windows based text editor, though you may prefer to use another.

4.2 Macintosh

- Netscape Communicator v4.7 and IE 4.5.
- BBedit v5.1.1 is the Mac text editor of choice.

BBedit needs to be configured by following the procedure below:
Filing: 'Edit' --> 'Preferences' --> 'Filing'
Select the Unix button in the **Default line breaks** section. You will also want to uncheck the **Force CR at End** box in the **When Saving** section.

HTML Markup: Select 'Edit' --> 'Preferences' --> 'HTML Markup'
Select the **HTML Tags upper case** button.
Select the **Quote Tag Attributes only if required** button.

Languages: 'Edit' --> 'Preferences' --> 'Languages'

Change no suffix to HTML by selecting **<no suffix>** in the **File suffix mappings** box, and selecting **HTML** in the **Language** pull down menu. Click the Replace button to make changes. Add these suffixes with their appropriate languages: .tpl=HTML, .mtpl=PERL, .plt=PERL,.tmpl=HTML. When changes have been made, click the Save button at the top window.

Default Font: 'Edit' --> 'Preferences' --> 'Editor'

Click the Set button to the right of the **Default font** box. Change the **Set tabs every _ spaces** to *2* instead of *4*. Click the Save button.

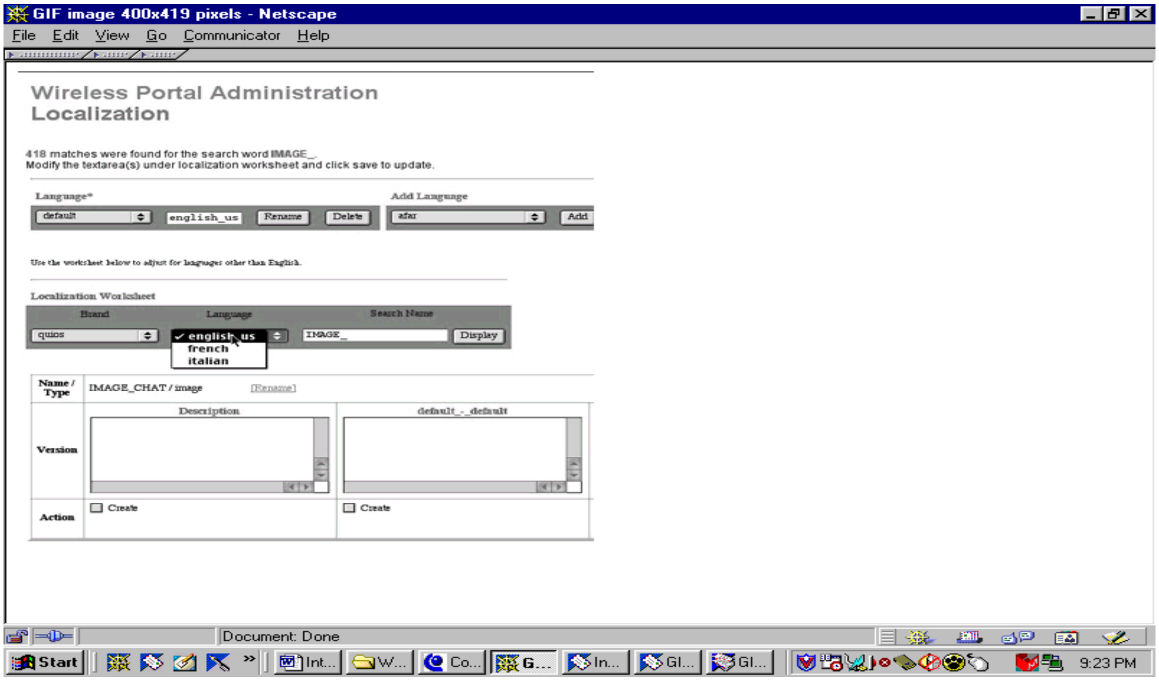
5. Localization Pages

There are two pages that deal with the standard HTML tagging for Internationalization and Co-branding:

The Localization Page allows you to add a new language to be used in the localization engine, or to delete or to rename a currently available language.

The Tags Page allows you to search for a tag by name. If the string is found in more than one tag, all the matching tags are displayed. Note that tag types are also displayed. You can use the drop-down menu next to the **Name** field to sort by type of tag.

5.1 Localization



GIF image 400x419 pixels - Netscape
File Edit View Go Communicator Help

Wireless Portal Administration
Localization

418 matches were found for the search word IMAGE_.
Modify the textarea(s) under localization worksheet and click save to update.

Language*: default | english_us | Rename | Delete

Add Language: afar | Add

Use the worksheet below to adjust for languages other than English.

Localization Worksheet

Brand	Language	Search Name
quios	✓ english_us french italian	IMAGE_ Display

Name / Type	IMAGE_CHAT / image [Rename]	
Version	Description	default_-_default
Action	□ Create	□ Create

URL: http://adm-dev.quios.net/trans/localization.mtpl

The result of a search returns every matching tag. Each tag has four fields: one language and brand independent, one language dependent and brand independent, one language independent and brand independent and finally one language and brand dependent.

The fields are called as follows:

default-default: bare tag with the text in plain English (no slang or local expression) and without any reference to the brand. This field is used as a reference to create the other fields.

<language>-default: this is the default-default field plus the language specific data only.

default-<brand>: this is the default-default field plus the brand specific data only.

<language>-<brand>: this is the default-default field with all the language and brand data or you can say it is the merge of the <language>-default field with the default-<brand> field.

Example incorporating language english_us:

default-default:	This is the home page
english_us-default:	This is the home page
default-quios:	This is the home page of QUIOS
english_us-quios:	This is the home page of QUIOS

Example incorporating language French:

default-default:	This is the home page
french-default:	Ceci est la homepage
default-quios:	This is the home page of QUIOS
french-quios:	Ceci est la homepage de QUIOS

You can edit and modify the tags as follows:

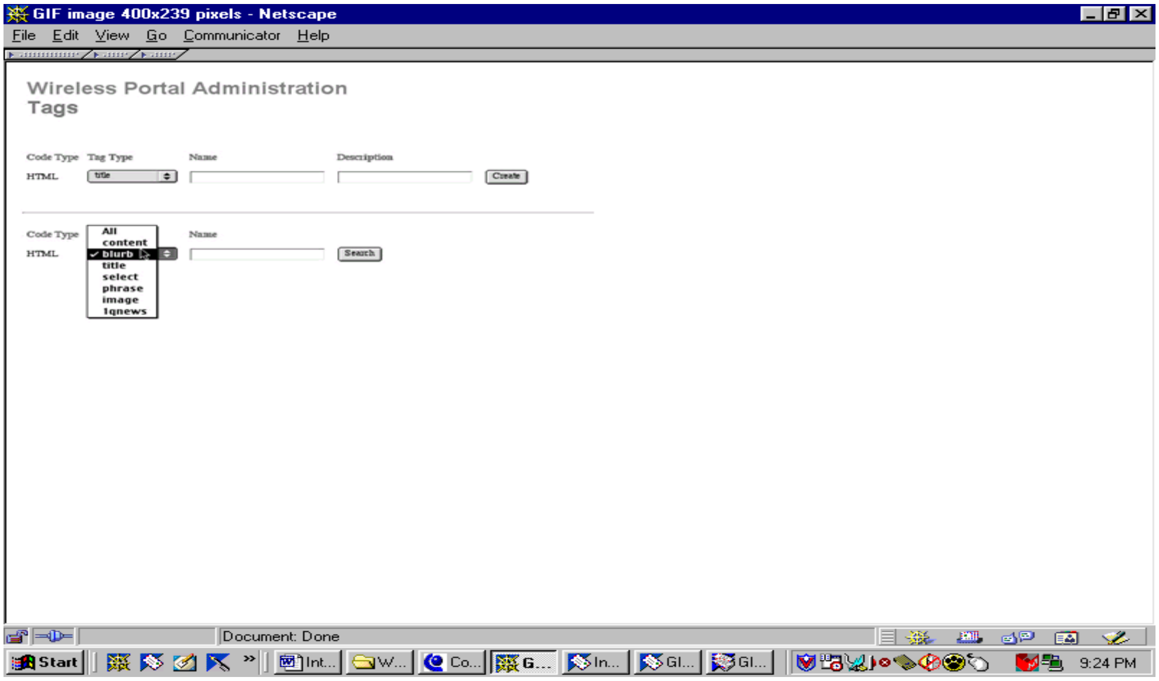
If a field has not yet been created, you can create it by filling it with the appropriate data and selecting the Create checkbox.

If a field has already been created, you can modify it by changing the data and selecting the Update radio button or you can delete it by selecting the Delete radio button.

Once every change is done, click on the Save button at the bottom of the page to update all the changes.

6. Tags Page

6.1 Steps for Content Tagging



GIF image 400x239 pixels - Netscape

Wireless Portal Administration
Tags

Code Type	Tag Type	Name	Description
HTML	title		

Code Type		Name
HTML	✓ blurb	

URL: **http://adm-dev.quios.net/trans/tags.mtpl**

The steps for content tagging should be used in conjunction with the Quios HTML guideline. Although the Internationalization process does not describe how to write good code, there are other tags that you may need to understand. These other tags are outlined within the HTML guideline.

1. Open appropriate file in BBEdit or text editor. You can do this through CVS.
2. Launch the Netscape browser and go to the tagging page. This page allows you to look up and create tags that represent the appropriate HTML sections to be localized in the .tpl files.
3. Select the content to be tagged.
4. Copy the content to be tagged onto the clipboard by highlighting and hitting CMD+C / CNTL+C.
5. Switch to your browser.
6. Select the appropriate code type for the tag you are creating from the **Tag type** popup menu. For our needs, this will be either **Title** or **Content**.

7. Enter the name of the tag to create in the **Name** field using the format mentioned in the Tag Naming Conventions section of this document. Remember to use all upper case characters, and to separate each section of the tag name with an underscore.
    a. **Page Tags**: If you are creating a tag for content, the first set of characters will be the brand, next the file name, third the phrase BODY_CONTENT, last the unique name of that tag. Where brand is not specified, it is assumed to be Quios.

       Ex: Quios Brand HELLO_BODY_CONTENT_HELLO

       Ex: Other Brand QNN_GOODBYE_BODY_CONTENT_BYE

    b. **Global Tags**: If you are creating a global tag, the first set of characters will be the brand, next GLOBAL, and lastly a short description.

       Ex: QNN_GLOBAL-SUBMIT
8. Enter description in Description field.
9. Click the Create button to create the tag.
10. Verify that you successfully created the tag. Choose the type of tag.
11. Type the name of the tag you just created into the **Search Name** entry field on the admin page. Click the Search button. Your tag should appear at the bottom of the page along with any other matches.
12. Click the Edit button for the tag you just created.
13. On the **Wireless Portal Admin Localization** page, you will see contents of tag displayed.
14. You can only display one Brand at a time.
15. Paste the contents from the clipboard into the Brand Code combination box.
16. Click on Create box underneath the text field
17. Click the Save button. For each field that you changed you will receive a confirmation.
18. Go back to step 4 until you no longer have any files that need tagging.

NOTE: To edit the default or English tags you must us the radios buttons beneath the text fields. Once you start editing the tag. Change the desired text and click on the update radio button.

6.2 Steps for Image Tagging

URL: http://adm-dev.quios.net/trans/tags.mtpl

1. Open appropriate file in BBEdit or text editor. You can do this through CVS.
2. Launch Netscape browser and go to the tagging page.
3. Look in your file for the image to be tagged.
   Ex: **<IMG SRC="/images/foo.gif" HEIGHT=100 WIDTH=150 HEIGHT=30 >**
   and
   <INPUT TYPE="image" SRC="images/foo.gif"…> should also be tagged. A tag of this type will contain only the SRC, WIDTH, HEIGHT and ALT attributes. This will be referred to as a "Input Type tag".
4. Copy code for the image to be tagged onto the clipboard by highlighting and pressing CMD+C / CTRL+C. The Input Type tag should contain the following attributes of the html tag: HEIGHT, WIDTH, SRC and ALT.
   Ex: Html tag is <INPUT SRC="images/templates/bt_on.gif" HEIGHT=30 WIDTH=10 BORDER=0> then the Input Type tag should be SRC="images/templates/bt_on.gif" HEIGHT=30 WIDTH=10 BORDER=0
5. Switch to your browser on the dev server.

6. Enter the name of the tag to create in the **Name** field using the format mentioned in the tag naming conventions section of this document. Remember to use all upper case characters, and to separate each section of the tag name with an underscore character.
   a. **Image tags**: If you are creating a tag for image, the first set of characters will be the Brand name. The second will be "IMAGE", followed by the name of the file.
      Ex: Quios brand IMAGE_FOO
      Ex: Other brand MSN_IMAGE_FOO
   b. **Input Type Tags**: If you are creating a tag for an **<INPUT TYPE="image">**, follow the format for image tags and add SRC to the end of the name.
      Ex: MSN_IMAGE_BAR_SRC
7. Click the Create button to create the tag.
8. Verify that you successfully created the tag.
9. Type the tag name you just created into the **Search name** entry field on the admin page.
10. Click the Search button.
11. Type the name of the tag you just created into the **Search name** entry field on the admin page. Click the Search button. Your tag should appear at the bottom of the page along with any other matches.
12. Click the Edit button for the tag you just created.
13. On the **Wireless Portal Admin Localization** page, you will see contents of tag displayed.
14. You can only display one Brand at a time.
15. Click the **Save** button. For each field that you changed you will receive a confirmation.
16. Go back to step 4 until you no longer have any .tpl files that need tagging.

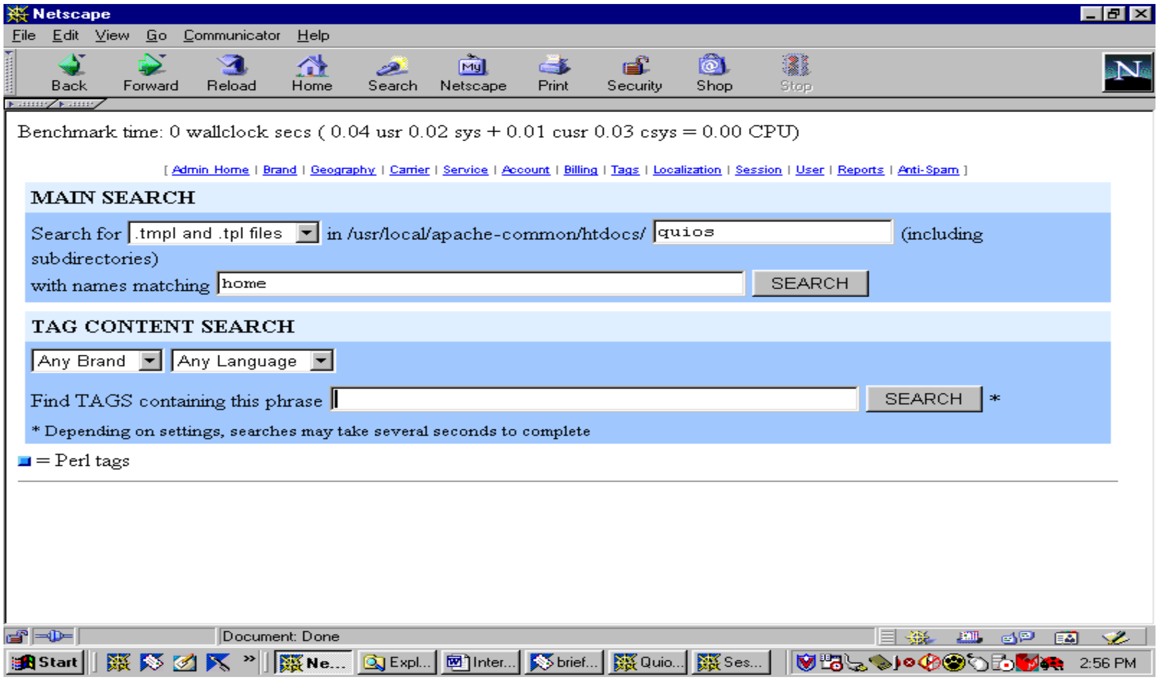
7. The Superlist Page

URL: http://adm-dev.quios.net/superlist.mtpl

There are two main tools on the superlist page.

7.1 Main Search

Users can get a listing of pages and the tags used on those pages. Users have an option of searching by the name of a file [.tmpl or tpl] or by the name of a tag [tags] by setting the pulldown option. The default directory to be searched is /quios, but can be easily changed to search other directories such as Excite or MSN. The search string goes into the text field right before the Search button. The search is case insensitive, so it doesn't matter if the user enters uppercase or lowercase. Each Search button corresponds to each search field.


Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Document: Done
Start Ne... Expl... Inter... brief... Quio... Ses... 2:56 PM

Benchmark time: 0 wallclock secs (0.04 usr 0.02 sys + 0.01 cusr 0.03 csys = 0.00 CPU)

[Admin Home | Brand | Geography | Carrier | Service | Account | Billing | Tags | Localization | Session | User | Reports | Anti-Spam]

MAIN SEARCH

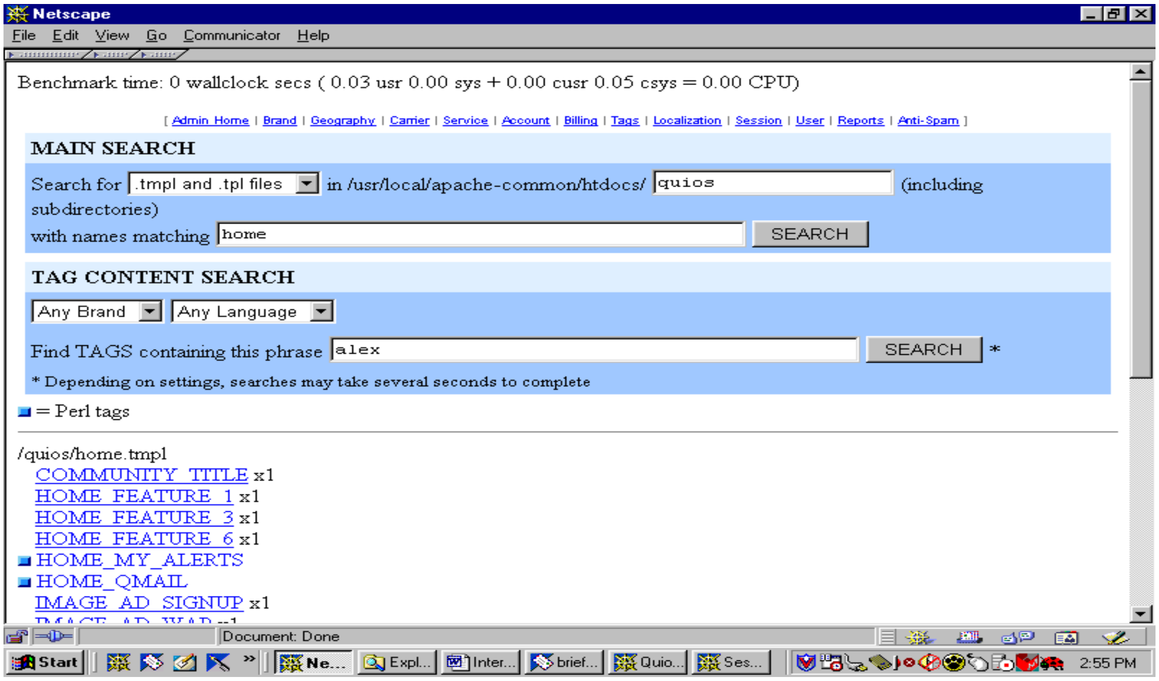
Search for .tmpl and .tpl files in /usr/local/apache-common/htdocs/ quios (including subdirectories)
with names matching home SEARCH

TAG CONTENT SEARCH

Any Brand Any Language
Find TAGS containing this phrase SEARCH *
* Depending on settings, searches may take several seconds to complete

■ = Perl tags

Clicking a tag name in the search results will take the user directly to the localization page where they can edit the current tag.


Netscape
File Edit View Go Communicator Help
Document: Done
Start Ne... Expl... Inter... brief... Quio... Ses... 2:55 PM

Benchmark time: 0 wallclock secs (0.03 usr 0.00 sys + 0.00 cusr 0.05 csys = 0.00 CPU)

[Admin Home | Brand | Geography | Carrier | Service | Account | Billing | Tags | Localization | Session | User | Reports | Anti-Spam]

MAIN SEARCH

Search for .tmpl and .tpl files in /usr/local/apache-common/htdocs/ quios (including subdirectories)
with names matching home SEARCH

TAG CONTENT SEARCH

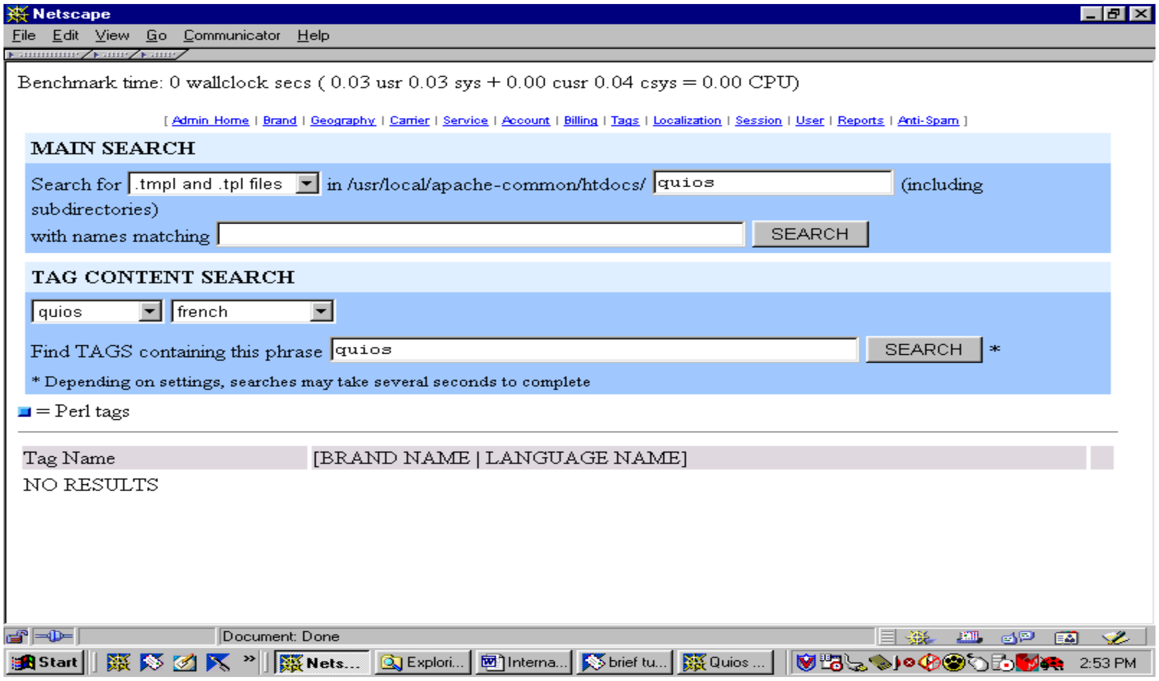
Any Brand Any Language
Find TAGS containing this phrase alex SEARCH *
* Depending on settings, searches may take several seconds to complete

■ = Perl tags

/quios/home.tmpl
COMMUNITY_TITLE x1
HOME_FEATURE_1 x1
HOME_FEATURE_3 x1
HOME_FEATURE_6 x1
■ HOME_MY_ALERTS
■ HOME_QMAIL
IMAGE_AD_SIGNUP x1
IMAGE_AD_WAP x1

7.2 Tag Content Search

Users can search for a specific string and find all tags whose content matches that string. By selecting a brand from the pulldown menu and selecting a language from the pulldown menu, users can refine and expedite their searches. Leaving the Any Brand and Any Language settings will allow the users to search all existing brands and languages that contain the specified string, but will take much longer to process and complete. (10-20 seconds is not unlikely) This search is also case insensitive.



Benchmark time: 0 wallclock secs (0.03 usr 0.03 sys + 0.00 cusr 0.04 csys = 0.00 CPU)

[Admin Home | Brand | Geography | Carrier | Service | Account | Billing | Tags | Localization | Session | User | Reports | Anti-Spam]

MAIN SEARCH

Search for .tmpl and .tpl files in /usr/local/apache-common/htdocs/ quios (including subdirectories)

with names matching SEARCH

TAG CONTENT SEARCH

quios french

Find TAGS containing this phrase quios SEARCH *

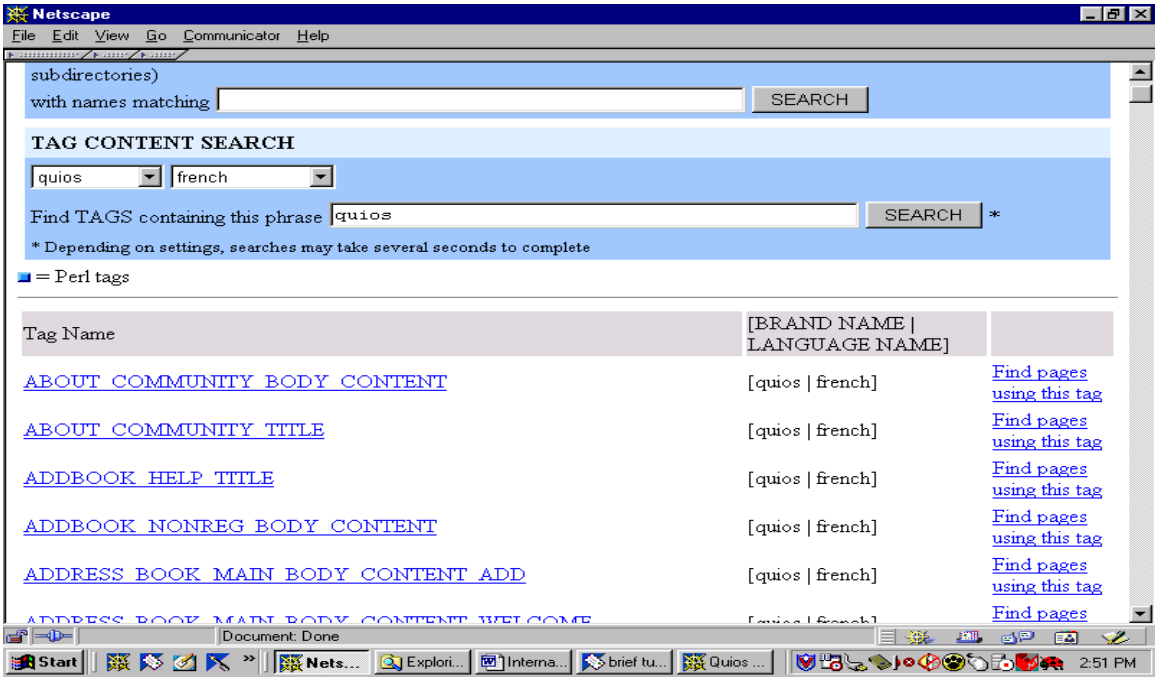
* Depending on settings, searches may take several seconds to complete

🟦 = Perl tags

Tag Name [BRAND NAME | LANGUAGE NAME]

NO RESULTS

Clicking a tag name in the search results will take the user directly to the localization page where they can edit the current tag.



subdirectories)
with names matching [ ] SEARCH

TAG CONTENT SEARCH

quios ▾ french ▾

Find TAGS containing this phrase [quios] SEARCH *

* Depending on settings, searches may take several seconds to complete

🟦 = Perl tags

Tag Name	[BRAND NAME | LANGUAGE NAME]	
ABOUT_COMMUNITY_BODY_CONTENT	[quios | french]	Find pages using this tag
ABOUT_COMMUNITY_TITLE	[quios | french]	Find pages using this tag
ADDBOOK_HELP_TITLE	[quios | french]	Find pages using this tag
ADDBOOK_NONREG_BODY_CONTENT	[quios | french]	Find pages using this tag
ADDRESS_BOOK_MAIN_BODY_CONTENT_ADD	[quios | french]	Find pages using this tag
ADDRESS_BOOK_MAIN_BODY_CONTENT_WELCOME	[quios | french]	Find pages

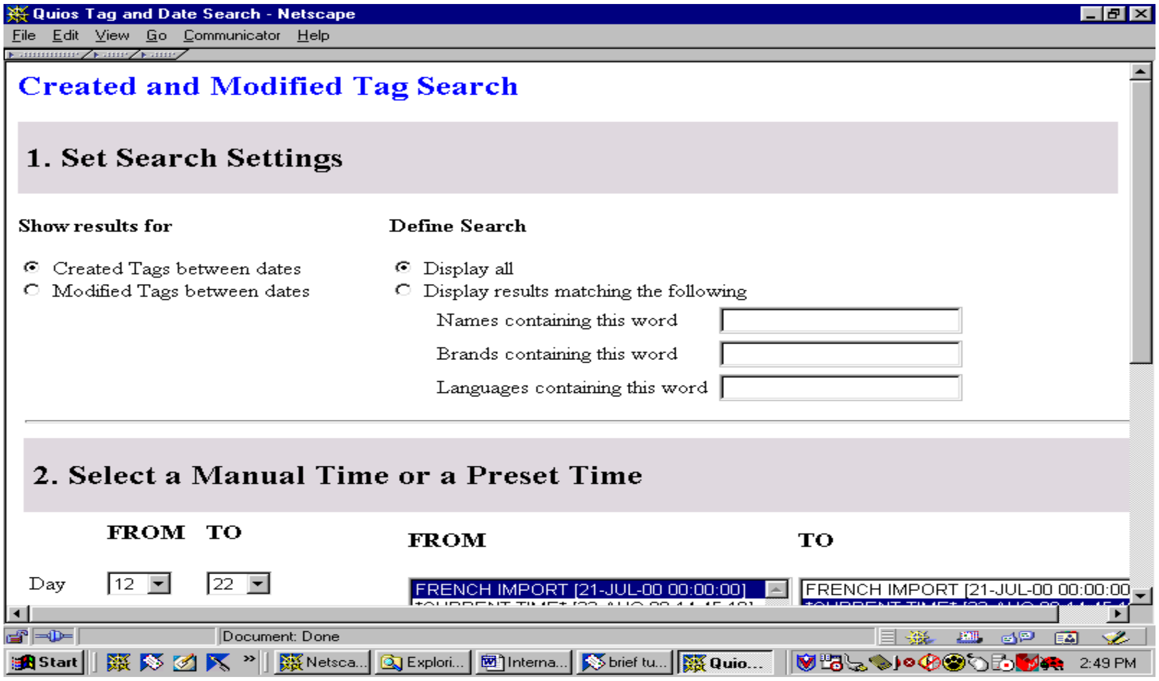
8. Created and Modified Tags Search

URL: http://adm-dev.quios.net/time.mtpl

Users can easily find what tags were created between two dates.

8.1 Selection of Tag Type

User selects search options first. They can either searched for tags that have been created or tags that have been modified. Secondly, they can refine their search to only display names, brands, languages that match a certain pattern.



Created and Modified Tag Search

1. Set Search Settings

Show results for

● Created Tags between dates
○ Modified Tags between dates

Define Search

● Display all
○ Display results matching the following

Names containing this word
Brands containing this word
Languages containing this word

2. Select a Manual Time or a Preset Time

	FROM	TO
Day	12	22

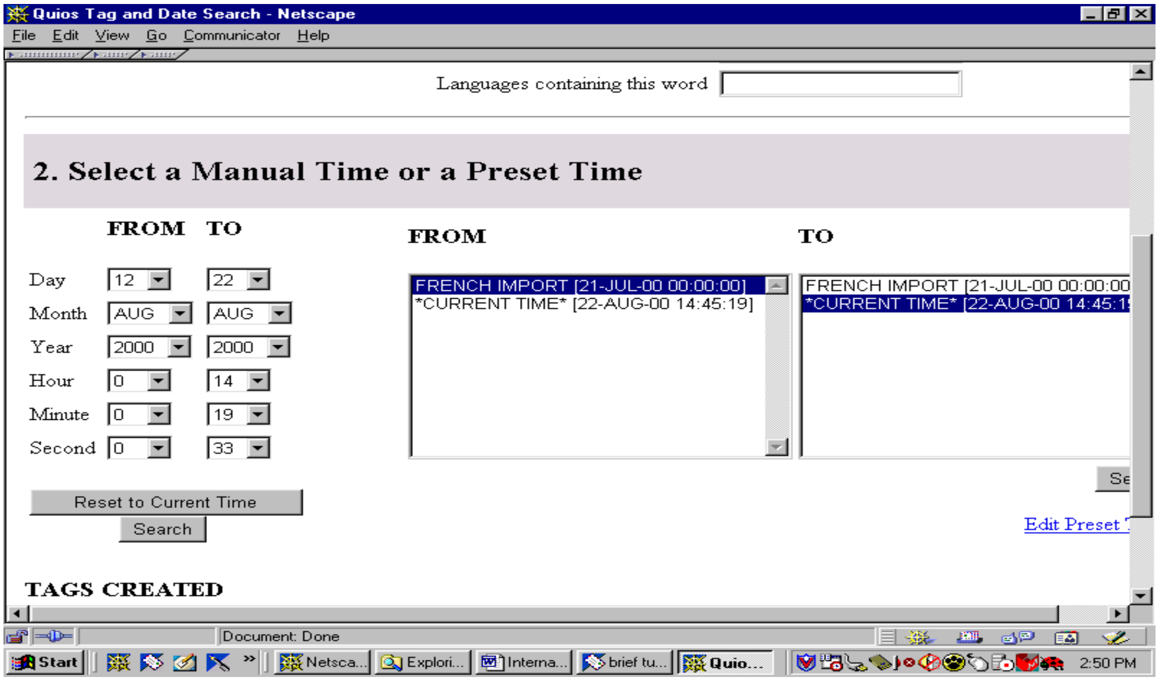
FROM: FRENCH IMPORT [21-JUL-00 00:00:00]

TO: FRENCH IMPORT [21-JUL-00 00:00:00

8.2 Selection of Time and Interval

User selects the time interval to search between. They can manually set the FROM and TO times with the pulldown options (the FROM and TO fields are defaulted to the beginning of the day to the exact second the user enters the page). A second alternative to manually selecting times is to use preset times. The preset times are simply names associated with a point in time.

For example, users could view tags that were created between BUILD 351 to BUILD 352 without having to remember the exact time of BUILD 351 and BUILD 352. All they would have to do is select BUILD 351 from the FROM field and BUILD 352 from the TO field and click search.



Languages containing this word

2. Select a Manual Time or a Preset Time

	FROM	TO
Day	12	22
Month	AUG	AUG
Year	2000	2000
Hour	0	14
Minute	0	19
Second	0	33

FROM

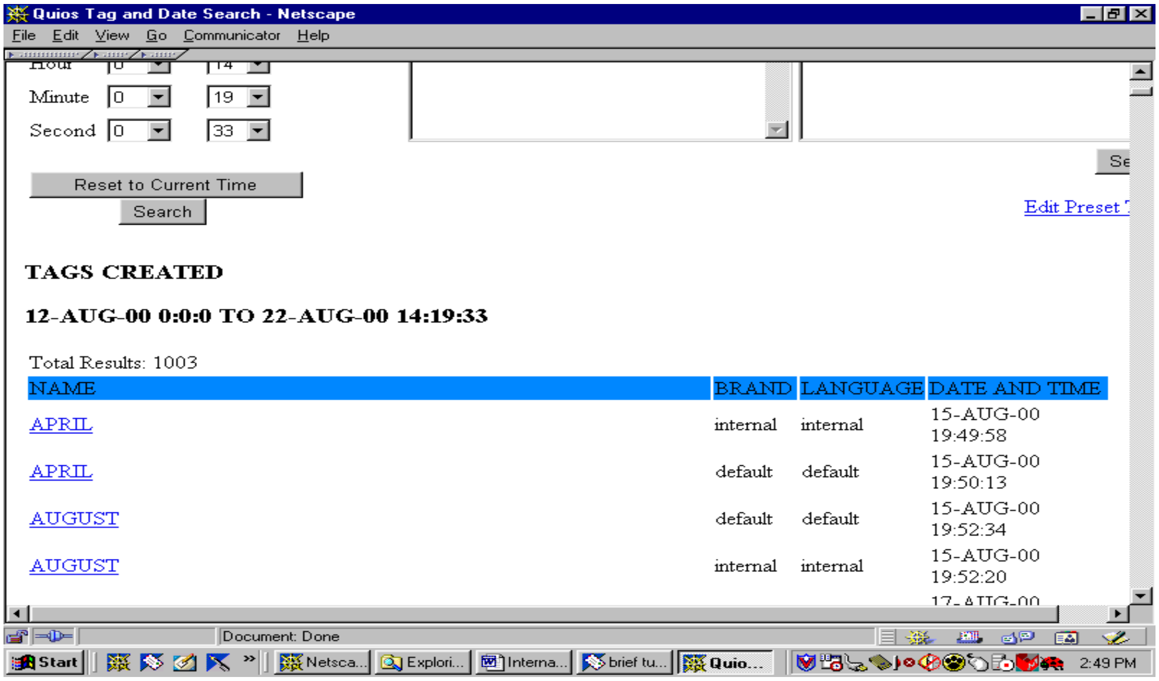
FRENCH IMPORT [21-JUL-00 00:00:00]
CURRENT TIME [22-AUG-00 14:45:19]

TO

FRENCH IMPORT [21-JUL-00 00:00:00
CURRENT TIME [22-AUG-00 14:45:1

TAGS CREATED

The Search button corresponds only to the field that it is near. In the example below, the Search button corresponding to the Manual Time has been clicked.



Hour 0 14
Minute 0 19
Second 0 33

Reset to Current Time
Search

TAGS CREATED

12-AUG-00 0:0:0 TO 22-AUG-00 14:19:33

Total Results: 1003

NAME	BRAND	LANGUAGE	DATE AND TIME
APRIL	internal	internal	15-AUG-00 19:49:58
APRIL	default	default	15-AUG-00 19:50:13
AUGUST	default	default	15-AUG-00 19:52:34
AUGUST	internal	internal	15-AUG-00 19:52:20
			17-AUG-00

9. Appropriate Use of Quios Dynamic Tags

9.1 How Dynamic Tags Should Be Used

The primary appropriate use of Tags is for the purpose of providing a modular piece of Web content, such as a logo, a dollar value, or some text, in multiple brands and/or languages. A secondary use is for snippets of HTML that will appear in many places on the Web site.

The vast majority of Tags should contain static HTML in a specific brand and language, and one version of the Tag will exist for each brand and language. These are stored in the Database as CODE_TYPE_NAME "html" in the TAG_TRANS table.

A much smaller minority of Tags will be Perl tags, which contain actual Perl code. These can be used to reduce the multiplication of Tags stored in the database for purposes of filling in specific UI elements, such as fixed or drop-down lists, in specific brands and languages. Scripts inside Perl tags should be kept very short and simple.

A good use of Perl tags is to create partially dynamic interface elements, such as pull-down menus, that are generated from the many _trans tables in our database. Tags run in their own private package space and non-lexical variables will retain their values between evaluations. For this reason it is a significant performance advantage to use a package global to cache the results of a single database lookup at startup time, rather than making repeated identical database calls that will always return the same result.

9.2 How Dynamic Tags Should NOT Be Used:

Perl tags are intended **only** to solve Internationalization and Branding problems **and should never be used to implement site work flow or any task of any kind spanning more than 1 Web page delivery.** They should always be well commented and kept to as few non-commented lines of straightforward logic as possible. A senior developer needs to approve each and every new Perl tag, or set of Perl tags for a specific feature. Perl tag scripts fragment the code and make it very difficult to check program flow.

9.3 What Not to Tag – Standard HTML

Do not tag the following:

- Images that are inside template variables e.g. <A HREF=<TMPL_VAR NAME=COPYLINK>><IMG SRC="/images/alerts/bt_copy_settings>Template variables cannot be placed inside tags and vice-versa.
- <TABLE> tags. <TR> AND <TD> can be put into a HTML tag but we leave <TABLE> and </TABLE> outside of any tags.
- Spacer images: space.gif, grey.gif, s.gif are all 1 pixel gifs used for spacing tables. They do not need a tag.

In general, in tagging content, we try to tag just the text in a page. If there is quite a bit of text spanning a few table rows, then you can go ahead and put the <TR>'s and <TD>'s inside the tag. Remember that the translation company needs some context for doing the translation, so its probably a good idea to err on the side of too much information then not enough.