QAD

Web Site Technical Specifications Document Version 2.1

Prepared by
Rachele Forsythe
Web Developer
Internet Marketing

Web Team

Table of Contents

1. Introduction 1
2. Requirements 2
2.1 Color Palette *2*
2.2 Fonts *2*
2.3 Page Elements *3*
2.3.1 Top-Level Elements 3
2.3.2 Head Elements 3
2.3.3 Block-Level Elements 3
2.3.4 Table Elements 4
2.3.5 Inline Elements 4
2.4 Navigation *5*
2.4.3 Persistent Navigation Items 5
2.4.3 Left Navigation Items 5
2.4.3 Breadcrumb Script 5
2.5 Graphics *6*
2.5.1 Creating Graphics 6
2.5.2 Creating GIFs 6
2.5.3 Creating JPEGs 7
2.6 Cascading Style Sheets *7*
2.7 Server Side Include Files *7*
2.7.1 Page-Building Include Files 8
2.7.2 Navigation Include Files 8
2.7.3 Miscellaneous Include Files 8
2.8 Templates *8*
3. Guidelines 9
3.1 File Naming Conventions *9*
3.1.1 Files in a Series 9
3.2 Editing Code *10*
3.3 Image Tags *11*
3.5 Identifying Supplemental File Types and Sizes *12*
3.6 Opening New Browser Windows *12*
3.7 Date Format *13*
4. Page Layout 14
4.1 Main Navigation Area *14*
4.2 Left Navigation Area *14*
4.3 Content Area *14*
4.2 Right Navigation Area *14*
4.5 Footer Area *15*
5. Implementation 16
5.1 Directory Structure Scheme *16*
5.1.1 Section Folders 16

5.1.2 Includes Folder 17
5.1.3 Images Folder 17
5.1.4 Resources Folder 17
5.1.5 Source Folder 17
5.1.6 CSS Folder 18
5.1.7 JavaScripts Folder 18
5.1.8 Flash Folder 18
5.2 Naming HTML and Include Files 18
5.3 Naming Graphic and Supplementary Files 18
5.4 Insertion of Content 19
5.5 Main Navigation Links 20
5.6 Side Navigation Links 20
5.7 PDF Documents 21
5.8 CGI Files 22
5.9 Development Process 22
5.9.1 Web Request Process 23
5.9.2 Browser Testing 24
5.9.3 Development Server 24
5.9.4 Test Server 24
5.9.5 Live Servers 24
Appendix A – *Page Layout by Sections* 25
Appendix B – *Pixel Layout* 26
Appendix C – *Code Layout* 27
Appendix D – CSS Defined 28

1. Introduction

This document will serve as the guideline for the creation and maintenance of QAD's corporate Web presence – www.qad.com. The attempt has been made to define every element and area of each type of file so as to keep consistency and uniformity across the domain.

Please refer to the appendices for visual examples of page layouts including diagramming of page sections and Styles.

- **Requirements**
- **Guidelines**
- **Page Layout**
- **Implementation**
- **Appendices**

2. Requirements

The follow subsections will detail the page-level conditions for each page constructed on qad.com.

2.1 Color Palette

Use of hexadecimal color codes is mandatory to ensure proper color display. The format for using colors includes using the # symbol and no quotation marks. Use of quotation marks leads to instances of discoloration in Netscape. Text colors should only be defined in the Style sheet.

#000000 black	#666666 dark gray	#663399 purple	#000066 dark blue	#660033 Burgundy
#FFFFFF white	#999999 medium gray	#99CC33 bright green	#003366 dark blue	#CC0033 bright red
#FF9900 orange	#CCCCCC light gray	#669933 medium green	#336699 medium blue	#009999 teal

Note: This hexadecimal color palette does not pertain to graphics.

2.2 Fonts

Definition: Fonts are complete typesets of one face and its varients.

The true type font family used for all text is

Arial, Helvetica, San-Serif.

Graphics are restricted to the Gill Sans font family unless others are approved by MarCom.

2.3 Page Elements

Definition: Page Elements, or tags, in HTML provide formatting specifications for everything within the document. The tags mentioned in the follow subsections are by no means a definitive list of allowed tags; they are simply tags that need special mention for formatting specifications.

2.3.1 Top-Level Elements

<head>
The head tag contains the page title, most JavaScript and Style sheet information.

<frameset>
The decision to use frames on a section will be made on a case-by-case basis. The code standard frameset can be found in the Source directory.

2.3.2 Head Elements

<meta>
Meta tags, keywords and content, appear on every page. There is one set of tags for the home page and one set per main navigation section. These tags are primarily for indexing the site with external search engines. In some cases, ht://Dig-specific Meta tags may be added to a page to increase its ranking internally.

```
<META NAME ="keywords" CONTENT="life, universe, mankind, plants,
relationships, the meaning of life, science">
<META NAME="description" CONTENT="This page is about the meaning of
life, the universe, mankind and plants.">
```

<title>
Page titles should be concise and directly reflect the content on the page; the same text will appear in the breadcrumb script.

2.3.3 Block-Level Elements

 and
Use of the tag is problematic, mainly because it always creates a line break in most browsers. Instead of using either or , define alignment within a <td> tag or <p> tag, if possible..

2.3.4 Table Elements

<table>
When creating a new table, it is imperative to define it in the follow manner:

```
<table width="(≤ 411 or %)" border="0" cellpadding="0" cellspacing="0">
```

The content area of every page is 100% and has a maximum defined width of 411 pixels. Cellspacing, cellpadding and border widths can be altered in special circumstances but should always be defined as "0" by default.

<td>
When creating a new <td> cell, define the width, align and valign parameters and ALWAYS place the </td> at the end of the content. DO NOT BREAK THE LINE as this has a tendency to cause unwanted extra lines following the end of the cell.

```
<td width="100%" align="left" valign="top">Call for Presenters</td>
```

2.3.5 Inline Elements

<a>
The <a> tag is defined within the Style sheet. When linking a file path, the use of ../ is deprecated in favor of the full file path from / (root). Use quotes around the file path.

<font> or <p>
These tags should be used to define text by referencing the appropriate class.

<img>
Images must have a defined height, width, border and alt tag (use of the title tag is optional but recommended). When linking a file path, the use of ../ is deprecated in favor of the full file path from / (root). Use quotes around the file path.

2.4 Navigation

Definition: Within a Web site, the navigation is how information is organized and the process through which users move from one page to the next.

The navigation has the following main areas:

- **Solutions**
- **Industries**
- **Company**
- **Services (a.k.a. Support or ServiceLinQ)**
- **Alliances**

2.4.3 Persistent Navigation Items

- Home (QAD logo)
- Accounts
- Worldwide
- Support
- Contact
- Search
- Site Map
- Copyright
- Privacy Policy
- Terms and Conditions

2.4.3 Side Navigation Items

Some 2nd – 5th level pages may require left navigation, which is definable per main navigation section.

2.4.3 Breadcrumb Script

Each page contains a call to the breadcrumb JavaScript, which dynamically creates a hierarchy of links at the top of the content area by parsing the directories in the URL.

`http://www.qad.com/services/support_tools/documentation/`

This URL would appear as the breadcrumb

Home > ServiceLinQ > Support Tools > Documentation

The last item in the series is not a link; it is the text located within the <title> tag of the page.

NOTE: Within this JavaScript file, folders can be eliminated from showing up in the breadcrumb or given an alias.

2.5 Graphics

Definition: Graphics are image elements in a Web document specified with the <img> tag; they are usually GIF (graphics interchange format) or JPEG (Joint Photographic Experts Group) formats. File size for images is limited to 100k each and should optimally be 10-15k each on prominent pages such as the home.

2.5.1 Creating Graphics

When creating graphics, first save the source file (.psd in PhotoShop or .png in Fireworks). This allows the graphic to be altered at any time without losing any quality that would result from trying to alter a lower resolution file such as a .gif or .jpg.

After creating a graphic in PhotoShop or Fireworks, evaluate the needs of that graphic so as to determine in which format to save it.

NOTE: Creating optimized Web graphics is an art. The following guidelines for creation of graphics are meant only to assist in their creation; experimentation for proper look and feel is encouraged. Additionally, MarCom should be consulted for large-scale and prominent graphics. Contact jsv@qad.com.

2.5.2 Creating GIFs

If a graphic has few colors or contains text, “save for web” as a .gif. GIFs have a maximum of 256 colors and work well for smaller, less detailed images. Try different color amounts and dithering options for the best look to file size ratio.

2.5.3 Creating JPEGs

JPEG is the format of choice for larger, more-detailed images such as photographs. When "saving for web," try several setting that modify the quality of the JPEG preview; it is rarely necessary to have high quality, as this will add to the file size but not always to the visible quality; JPEG can achieve acceptable quality at a large compression rate.

2.6 Cascading Style Sheets

Definition: Cascading Style Sheets (CSS) are files that contain statements about page elements and are linked to Web documents by the <style> tag within the <head> tag. CSS Styles are called using `class="parameter"`. Making a change to a Style within a CSS file reformats any elements defined using that same Style anywhere on the Web site.

All Styles used must be included within a Style sheet, including redefined HTML elements as well as any custom Styles.

NOTE: Netscape renders some Styles differently than IE, for example hover Styles for links are not supported in Netscape 4.x.

NOTE: Classes within other classes take precedence.

2.7 Server Side Include Files

Definition: A Server Side Include is a file containing a set of data that is called on the page and executed by the server before displaying the page to the client. Includes make it possible to have the same data appear in several places but globally modifiable by a single change in one file.

```
<!--#include virtual="/includes/us/mainnav.inc"-->
```

NOTE: As a rule of thumb, any element organized to appear on two or more pages should be incorporated as an include file if possible.

2.7.1 Page-Building Include Files

These files contain the basic HTML that creates the basic structure of each page on the site. They are considered globally applicable and should not be altered.

2.7.2 Navigation Include Files

These files contain all the data for the top, main, left and footer navigations. They are applicable per language site.

2.7.3 Miscellaneous Include Files

These files and their variants contain the data for other modular elements, for example the search form or PDF help link, and are easily moved to different areas on a page.

2.8 Templates

Definition: A template is a file that sets up default parameters for a file that can then have data added to it.

The templates are simply files with all parameters defined for each page or file type and are located in the Source directory. Download the needed source file and rename it to the proper new file name. The page title, headings and content should then be added to the newly saved file.

3. Guidelines

This section is intended to provide concrete usage rules for style and implementation including explanations of preferred methods and instructions for usage of newly accepted technologies.

3.1 File Naming Conventions

All file names should be lower-case ***without*** spaces.

```
qad_automotive_brochure.pdf
```

NOT

```
QAD Automotive Brochure.pdf
```

Capitalization, especially in combination with QAD's case-sensitive Apache servers, is problematic in a Web environment. Additionally, Netscape does not always process file names containing spaces.

While the second file name works well on a personal computer, it is disorganized in a Web environment where capitalization can affect access to files.

Note: *The underscore is your friend*.

3.1.1 Files in a Series

Files which are sequential by number should be named

```
01_firstfile.html
02_secondfile.html
03_thirdfile.html
```

NOT

```
1_firstfile.html
2_secondfile.html
3_thirdfile.html
```

The second list will result in files being out of order in the directory folder after 10 files.

Note: If the series of files is greater than 99, use 001, 002, 003, etc.

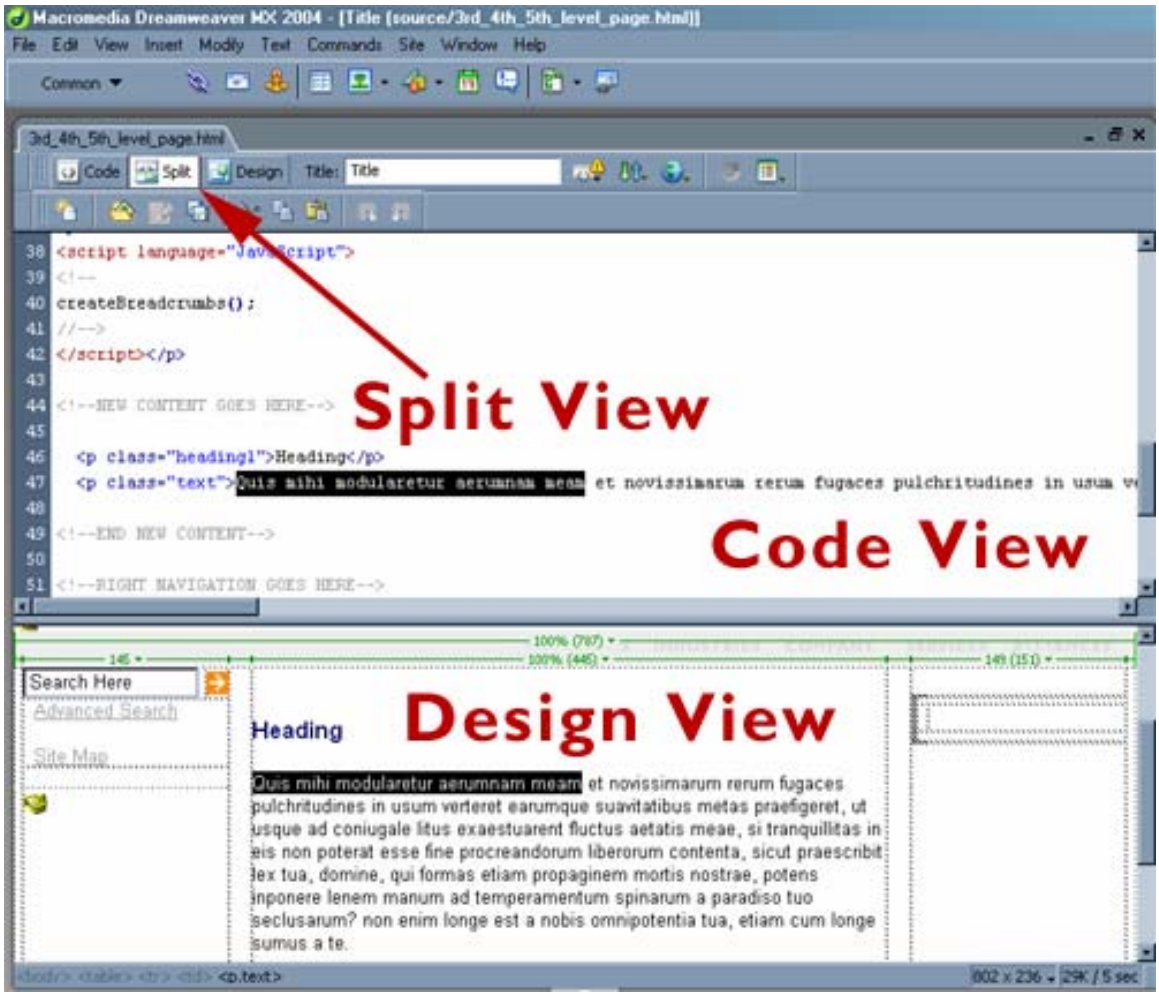
3.2 Editing Code

Use of the “Design” view should be avoided. When changes are made in the Design view, Web development tools attempt to write HTML code for the user, which is generally not optimized and makes working with the Code view afterwards extremely tedious.

If working strictly with the code is not feasible, Dreamweaver has a split view where the Design and Code views can be displayed simultaneously. Segments of the document can be highlighted in the Design view for easy reference but *should only* be altered in the Code view to eliminate extraneous tags and to achieve as much **precision** as possible.

Use of the Design view is also not conducive for trouble-shooting design discrepancies between Netscape and Internet Explorer; it does not accurately render the page and small errors, such as a broken </td>, are not always shown. This makes it impossible to correct these errors in the Design view.

Only by manipulation of the HTML in the Code view the proper optimization of HTML be achieved.



Dreamweaver MX: Code and Design views can be shown simultaneously.

Dreamweaver is the development tool that QAD has purchased. It is by no means the only tool that can or should be used to edit code for any QAD Web site. Other editors including BB Edit, Notepad, Homesite, etc. are permissible but the stipulation of only working directly with the HTML to alter code still remains.

3.3 Image Tags

Most images should have text within the alt attribute of the image tag. This text should descriptively indicate the content and purpose of the image. The alt attribute will give information to a client that may be using a reading device or has image viewing disabled.

Because the QAD Web presence is designed for use in Netscape 6, the use of the title attribute as well as the alt attribute within image tags is strongly recommended as Netscape 6 does not display alt and Netscape 4.x does not support title:

```
<img src="/images/qad.gif" width="10" height="10" alt="QAD"
title="QAD">
```

3.5 Identifying Supplemental File Types and Sizes

The need for a standard convention of identifying supplemental file types and sizes is necessary to keep a consistent look and method for identification of these files located within the QAD Web presence.

After a link to a supplementary file, an icon linked to a help file followed by the file size should be inserted using the code below; this example uses a PDF file:

```
<font class="sidebar">
[<!--#include virtual="/includes/us/pdf_help.inc"-->
 <!--#config sizefmt="kilobytes"-->
 <!--#fsize virtual="/company/resource_library/qad.pdf"-->]
</font>
```

1. The appropriate include file is inserted which contains the identifying icon and a link to its help page.
2. The script that displays file size is inserted. Put the file name of the file being linked inside of `#fsize file=""`.

QAD Automotive Brochure [208k]

This script works whether the file and the page on which the script appear are in the same directory or not. The relative path to the file should be used.

The above procedure should also be used for those links that require a Web account.

3.6 Opening New Browser Windows

It may be necessary to open new windows when:

- Linking to a URL not part of qad.com

- Linking to a supplementary file such as a PDF or video file
- Opening an application that does not carry the standard look and feel seamlessly
- When pushing out a new piece of the Web site that may temporarily not carry the standard look and feel

In addition to use of the `target="_blank"` attribute, the new window must be opened with a new window JavaScript function. Most new windows will be a standard 640x480 size and can be opened using the function **nw**, located in head.js.

```
<a href="http://www.gartner.com/" target="_blank"
onclick="nw('http://www.gartner.com/');return false;">Gartner</a>
```

Other popup scripts can be written and used as needed.

3.7 Date Format

Dates should appear as dd month yyyy, i.e. 11 December 1999. This is a more global format and less confusing than 12/11/99, which to much of the world means November 12.

4. Page Layout

This section will explain the basic layout of a page, specifically those elements which are similar to all pages. Most of these "areas" are contained within include files.

NOTE: The useable area of every page is 783 pixels, which is based on the 800x600 Web-standard resolution for development. According to the design, 760 is the maximum page width in pixels, though the entire site is defined using 100%.

4.1 Main Navigation Area

The main navigation area is everything at the top of the page – QAD logo and images, top navigation and main navigation. Additionally, there may be an image banner defining the section in which the page resides.

4.2 Left Navigation Area

This area is **145 pixels** wide and is where the search box, advanced search link and site map links are located.

4.3 Content Area

This area has a maximum width definition of **411 pixels** but is defined using 100% in order to span the entire width of the screen. The area contains the breadcrumb script, page heading and page content.

4.2 Right Navigation Area

This area is **149 pixels** wide and is located as a column on the right side of the page. It is used to display the side navigation as well as to list certain other links that pertain to the page or section but are not part of the persistent or left navigations. It is an optional area and may be removed by altering include files.

If the right navigation area is not used, the maximum usable area of the content area becomes **577 pixels**.

4.5 Footer Area

The footer appears beneath the main defining table so that it will appear last on the page and contains the company tagline and links to copyright, privacy policy and terms and conditions information.

5. Implementation

Implementation refers to the process of organizing and structuring all data and files appearing within QAD's Web presence.

5.1 Directory Structure Scheme

The home page is located at `/ (root)` level along with other specialty file and section directories. Housing these folders at `/ (root)` severely lessens the occurrence of multiple copies of a file existing in buried subdirectories.

NOTE: Subfolders should be created only with the intent to use another index.html page or if it houses special files.

NOTE: Files should be referenced with a full file path:

```
/includes/us/mainnav.inc
```

NOT

```
../mainnav.inc
```

HTML files for regional sites should be housed within a directory according to the ISO two-letter country code for that country.

AU – Australia, BR – Brazil, DE – Germany, FR – France, IT – Italy, MX – Mexico, NL – Netherlands, PL – Poland, UK – United Kingdom

All other files should be housed first in the directory related to the file type, i.e. images, then inside of a regional directory for the country's two-letter country code.

5.1.1 Section Folders

Section folders are located at `/ (root)` level. Subsections of main navigation sections will not have subfolders unless they will house more than one file. As far as the breadcrumb script is concerned, having a directory with an index file versus just a file named doesn't affect its output.

`http://www.qad.com/services/incident_management/`

versus

`http://www.qad.com/services/incident_management.html`

Both of these will return the same breadcrumb, as the script never displays the last directory or file automatically; that data is pulled from the <title> tag.

Home > ServiceLinQ > Incident Management

5.1.2 Includes Folder

The includes folder is located at `/ (root)` level and contains every include file used on the site. Subfolders should be made only in special circumstances, such as housing include files for a regional site, e.g. /includes/br/.

5.1.3 Images Folder

The images folder is located at `/ (root)` level and contains every image file used on the site. Subfolders should be made only in special circumstances, such as housing image files for a regional site, e.g. /images/br/.

5.1.4 Resources Folder

The resources folder is located at `/ (root)` level and contains files which are neither images nor publications and would appear in several locations such as the site map and search pages.

5.1.5 Source Folder

The source folder is located at `/ (root)` level and contains source files for items such as images and html templates.

5.1.6 CSS Folder

The Style sheets folder is located at `/ (root)` level and contains every Style sheet used on the. Subfolders should be made only in special circumstances.

5.1.7 JavaScripts Folder

The JavaScripts folder is located at `/ (root)` level and contains files which are linked to on every page of the site and contain the JavaScripts used in the navigational elements. . Subfolders should be made only in special circumstances, such as housing .js files for a regional site, e.g. /javascripts/br/.

5.1.8 Flash Folder

The flash folder is located at `/ (root)` level and contains every Flash file used on the site. Subfolders should be made only in special circumstances, such as housing image files for a regional site, e.g. /flash/br/.

5.2 Naming HTML and Include Files

To reduce the confusing environment created by a multitude of subfolders containing files with the same name, file names should be as intuitive as possible. When naming HTML files, spell out the full name of the page heading or subheading. Some leeway can be made in this area, such as using known abbreviations, i.e. sv for Supply Visualization.

The appropriate file extensions for HTML documents is **`.html`**

The appropriate file extension for includes is **`.inc`**

5.3 Naming Graphic and Supplementary Files

Graphics and supplementary files should be named intuitively. If a graphic is of a button that says 'PDF' on it, name the graphic pdf_button.gif instead of button1.gif.

Having file names for frequently downloaded publications that fully identify the contents of the document is more helpful to users who save may these files for later reference.

5.4 Insertion of Content

The following steps should be taken when creating, transferring or reconstructing content:

1. The appropriate source file for the section where the page will be located on the server should be downloaded and used as the shell for the content, i.e. `/source/3rd_4th_5th_level_page.html` would be used to create a new events page.

2. Create the page title accordingly: EXPLORE 2004 Call for Presenters. This text will also appear as the indicating title at the end of the breadcrumb script. Only a description of the page should appear. Do not include the section or subsection.

3. Add the appropriate page heading; it should be roughly the same as the page title.

4. Alter the side navigation JavaScript, if applicable, according to the section of the site the file will appear beneath.

5. Insert or begin creating content in the space indicated by **<!--NEW CONTENT GOES HERE-->** and insert new body content ONLY beneath this comment. Do not insert content after the **<!-- END NEW CONTENT -->** comment.

6. The **maximum** definable size of any table located in the content area is 411 pixels. The use of percentages is also an option. If the option to remove the right navigation area is used, the maximum definable size increases to 577 pixels.

7. All HTML elements must be formatted according to the Style Sheet; therefore, all text **MUST** be formatted with the appropriate class. It is also important to note that Style tags take precedence over non-Style information.

8. If the content contains links to other pages, files or images, verify the file paths of those links and upload all dependent files.

5.5 Main Navigation Links

The main navigation drop-down is generated using Coalesys WebMenu Studio. Do not attempt to update these JavaScript files manually. Log a Web Request with the needed changes so that the files can be regenerated.

5.6 Side Navigation Links

One JavaScript file is used to call the side navigation for each of the five main sections of navigation, i.e. `/javascripts/us/services_sidenav.js` is used for every 2nd – 5th level page appearing under the Services section that requires left navigation.

The following navigational JavaScript references appear in the <head> tag of every 2nd – 5th level page located in one of the five main sections:

```
<script language="JavaScript" src="/javascripts/sidenav_functions.js">
</script>

<script language="JavaScript"
src="/javascripts/us/services_sidenav.js">
</script>
```

In side navigation area of each page, the following JavaScript is used to call the appropriate array to display navigational items for a 3rd, 4th or 5th level page:

```
<script language="JavaScript">
<!--
UseArray = Array_SupportTools;
createMenu('00');
//-->
</script>
```

If no side navigation links are required, these JavaScript calls should be removed.

Use the following steps to create and call the desired link:

1. All side navigations are arrays created in the .js file corresponding to the main section in which it will appear. Each 3rd level section requires a new array to be created for its corresponding links, i.e. Support Tools would require the creation of an array in the Services left navigation file.

   Note: Link text should be in all mixed case: Documentation.

2. Parent links begin with an arbitrary integer and inherit children by using periods: 03.01 might represent the 4th level link "QXtend" beneath the 3rd level link "Documentation" inside the array "Array_SupportTools."

```
//SUPPORT TOOLS
var Array_SupportTools = new Array (
[ "Knowledgebase","01","http://primus.qad.com/","_blank" ],
[ "Incident Management","02","/services/incident_management/","" ],
[ "Documentation" , "03" , "/services/documentation/" , "" ],
   [ "QXtend" , "03.01" , "/services/documentation/qxtend/", ""],
```

3. Once navigation items have been placed into the appropriate files, the page-level JavaScript needs to be altered according to which left navigation choice should be highlighted on each particular page.

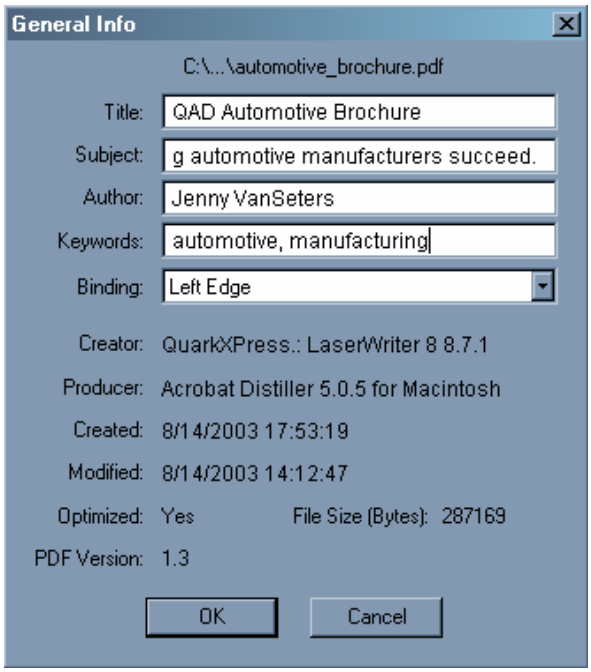
   Parameter ('00') is used when nothing in the side navigation is highlighted, that is, when the particular page is not one of the choices in the left navigation; this occurs only on 2nd level pages, i.e. `/services/ index.html`

```
<script language="JavaScript">
<!--
UseArray = Array_SupportTools;
createMenu('03.01');
//-->
</script>
```

5.7 PDF Documents

In order for the site search engine to display optimal results for PDF files, it is necessary to fill out the Meta data fields within the file using a PDF editor. These fields at minimum include Title, Subject (description) and Keywords (comma separated values); author may

not be known and is optional. The content of the title field is equivalent to the <title> tag on HTML documents.



To edit these fields, in Adobe Acrobat 4, choose File > Document Info > General; in Adobe Acrobat 5, choose File > Document Properties > Summary.

5.8 CGI Files

PERL scripts can be uploaded to the cgi-bin directory for execution on pages. Ht://Dig, error messages and the redirect page are some of the current scripts in use. These files should be downloaded in ASCII format for updating.

Note: Do not use cgi sendmail for form submission. All forms where data is being collected should be created in Domino Designer or Aprimo.

5.9 Development Process

All files for production to the extranet most go through a series of checks and verification before being published.

5.9.1 Web Request Process

The Web Request Process is used to update current content and request new content development as well as to alert the Web Team about problems on existing pages.

- Add the Web Team Database to your desktop.
- To log a New Request, begin by opening the Web Team Database.
- Click the "New Request" button on the tool bar; a New Request window will appear with several blank fields.
- SHORT DESCRIPTION: Give a short, title-like description summarizing the scope of the New Request.
- REQUEST TYPE: Choose an option from the dropdown menu that best describes the purpose of the New Request.
- PROPOSED PUBLICATION DATE: Enter a date when the New Request should be published ONLY if it is time-sensitive.
- WEB TEAM ASSIGNEE: All New Requests are "Unassigned;" when a New Request is approved, a Web Team administrator will assign it to the appropriate Web Team Member or a Web Team Members may choose to assign the requests to themselves.
- PROJECT & SUB-PROJECT: These fields should remain blank and will be filled in by a Web Team administrator if appropriate.
- REQUEST STATUS: Notice that the request is marked as "New Request;" when the status of the New Request changes, you will be notified by e-mail of its progress. It is the responsibility of the Web Team Assignee to update the status of the request as work progresses.
- DOMAIN: Check the box(es) of the domain(s) where the files are located for the New Request.
- LISTING OF URLs: Be sure to document the exact URL(s) of all pages and files that the New Request will affect.
- REQUESTOR COMMENTS: Provide all necessary information about the nature of the New Request; if the New Request is for a fix or change to existing content, please document the specific changes that need to take place. If the New Request is for new content, please insert or attach the content.
- Be sure to SAVE the New Request before exiting.

- You may view the status of the New Request at any time by returning to the Web Team Database, clicking the "Web Request" option in the side bar and searching for your name under "By Requester.

5.9.2 Browser Testing

All Web pages must be reviewed in **IE 4.x or higher**, **Netscape 4.x** and **Netscape 6.x or higher** for basic consistency. Netscape will *usually* display a little differently than IE but making sure nothing is outright broken in either Netscape is important.

NOTE: Both a 4.x and 6.x version of Netscape can be installed at the same time.

5.9.3 Development Server

www-dev.qad.com

All new files are created and uploaded to the development server for initial testing. Test each file in each browser and make all necessary changes to the files ***BEFORE*** uploading them to the test server.

5.9.4 Test Server

www-test.qad.com

Once a file is determined to be working properly in all three browsers, it can then be promoted to the test server.

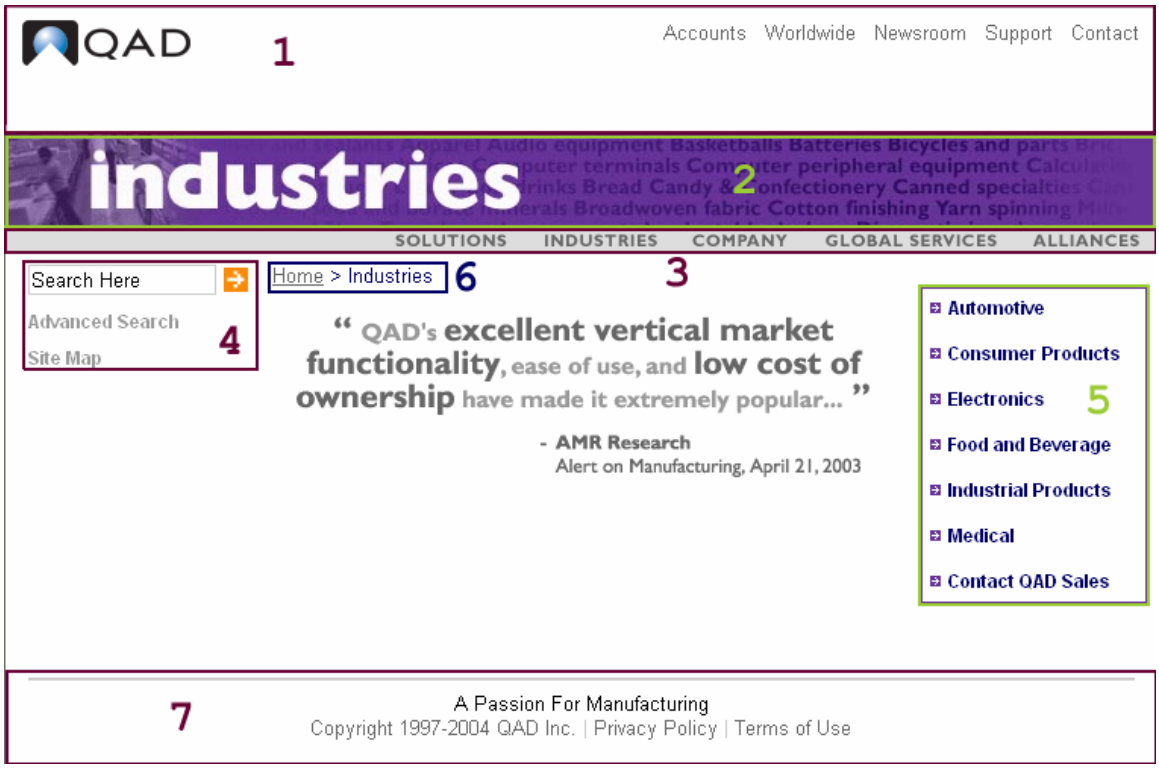
Note: Promote dependent files like images and PDFs.

The test server should, in theory, be an exact mirror of the live server. If anything breaks in test, remove it and return to the development server to troubleshoot the problem.

5.9.5 Live Servers

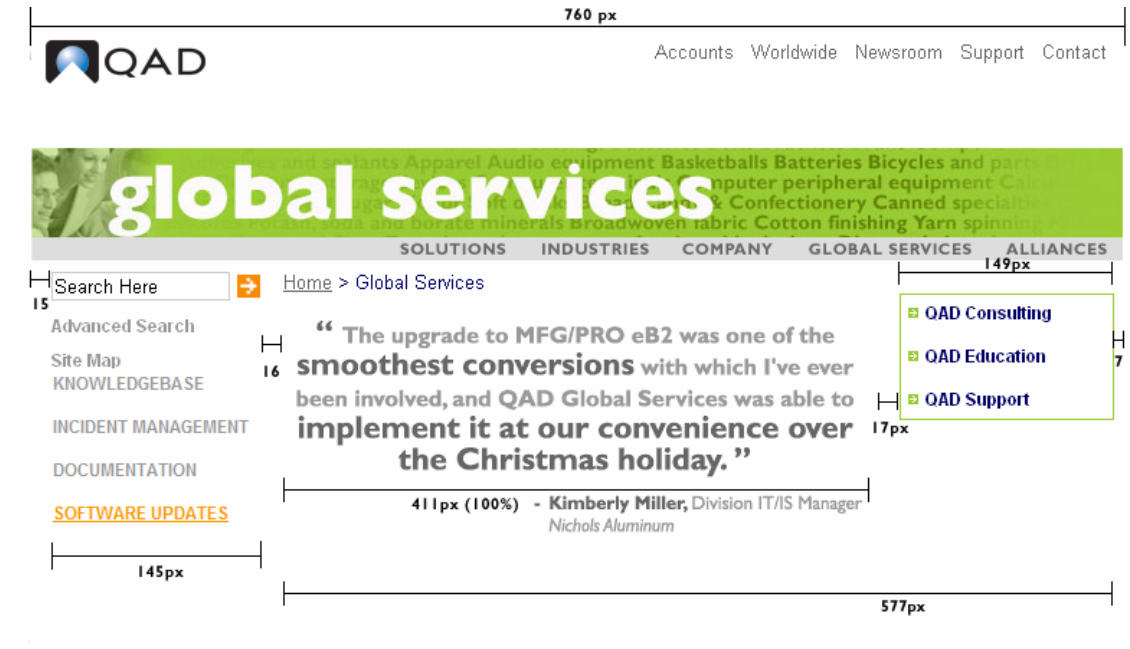
After a file performs successfully on the test server, it can be promoted to the live servers. It is important to check the file again in all three browsers after it goes live, making sure all elements are functional, especially links.

Appendix A – *Page Layout by Sections*



1. Header links
2. Section image
3. Main navigation
4. Search box
5. Side navigation
6. Breadcrumb script
7. Footer

Appendix B – *Pixel Layout*



Page width:	760px	
Left Gutter:	15px	
Left Navigation Area:	145px	
Content Gutter:	16px	
Content Area:	411px (100%)	577px w/o right
Resource Gutter:	17px	
Right Navigation Area:	149px	
Right Gutter:	7px	

Appendix C – *Code Layout*

```
<!--#include virtual="/includes/global_vars.inc"-->
<!--#include virtual="/includes/us/global_vars_REPLACE-ME.inc"-->
<html>
<head>
<title>Title</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<!--#include virtual="/includes/us/REPLACE-ME_keywords.inc"-->
<script language="JavaScript" src="/javascripts/head.js">
</script>
<script language="JavaScript" src="/javascripts/us/breadcrumb.js">
</script>
<script language="JavaScript" src="/javascripts/sidenav_functions.js">
</script>
<script language="JavaScript" src="/javascripts/us/REPLACE-ME_sidenav.js">
</script>
<link rel="stylesheet" href="/css/qad.css" type="text/css">
</head>
<!--#include virtual="/includes/body.inc"-->
<!--#include virtual="/includes/us/header.inc"-->
<!--#include virtual="/includes/us/section_header_large.inc"-->
<!--#include virtual="/includes/us/mainnav.inc"-->
<!--LEFT NAVIGATION GOES HERE-->
<!--#include virtual="/includes/us/search.inc"-->
<!--END LEFT NAV-->
<!--#include virtual="/includes/begin_content.inc"-->
<!--NEW CONTENT GOES HERE-->
  <h1>Heading</h1>
<!--END NEW CONTENT-->
<!--#include virtual="/includes/begin_rightnav.inc"-->
<!--RIGHT NAVIGATION GOES HERE-->
<script language="JavaScript">
<!--
UseArray = Array_REPLACE-ME;
createMenu('00');
//-->
</script>
<!--END RIGHT NAV-->
<!--#include virtual="/includes/begin_promo.inc"-->
<!--#include virtual="/cgi-bin/promo.pl?id=0"-->
<!--#include virtual="/includes/begin_footer.inc"-->
<!--#include virtual="/includes/us/footer.inc"-->
</body>
</html>
```

Appendix D – CSS Defined

.text - black
This style is used for most content text.

h1 – dark blue
This Style is used for page headings – 12pt.

h2 – dark blue
This Style is used for page headings – 11pt.

h3 – dark blue
This Style is used for page headings – 10pt.

h4 – dark blue
This Style is used for page headings – 8pt.

.sidebar - black
This Style is used when smaller text is needed.

.navtxt – dark blue
This Style is use for the text within navigators.

.breadcrumb – dark blue
This Style is used for the breadcrumb text.

a – default link

Link	Active	Visited	Hover
dark gray	orange	dark gray	orange

a.one – top navigation and footer links

Link	Active	Visited	Hover
dark gray	orange	dark gray	orange

a.two – search link

Link	Active	Visited	Hover
dark gray	orange	dark gray	orange

a.three – side navigation links

Link	Active	Visited	Hover
dark blue	orange	dark blue	orange

* Other styles are defined within the site style sheet and should be referred to if one of the above styles does not appear to meet requirements.