LawPort Application

Developer Guide

Version 3.7

SV Technology, Inc.

Contractor/manufacturer is:
SV Technology, Inc.
301 Howard Street, Suite 1410
San Francisco, CA 94108
415.817.0000

LawPort Application Developer Guide, Release 3.7, September 1, 2014

Table of Contents

About This Guide

LawPort is a comprehensive, browser-based, information management system that is structured around views and applications. This guide assists developers in their goal to modify and customize their firm's LawPort application by extending and supplementing its value with the functionality described in the chapters of this guide.

Audience

This guide is written for LawPort client developers.

Structure of this guide

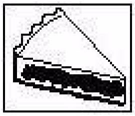
This guide consists of the following chapters:

- Chapter 1, "Simple API," shows developers how to integrate properties from the LawPort Simple API with their own external applications.
- Chapter 2, "Designing Custom URL Components," shows developers how to design custom URL components for LawPort.

- Chapter 3, “Web Services,” provides an overview of web services, defines each web service, and lists all of the request parameters and return values for each web service.
- Chapter 4, “Web Services for Internet Module,” discusses how developers can create their firm’s Internet site with the Web Services for Internet Module, shows how to set up the Module’s Internet templates, and lists all of the request parameters and return values for each Internet web service.
- Chapter 5, “Customizing LawPort: Developer Guidelines,” provides guidelines on how to customize LawPort, noting what types of customization approaches are supported by SV Technology (i.e, customizations that will not be lost during upgrades).
- Chapter 6, “Customizing LawPort Style Classes,” points out how, with regard to principles such as class precedence rules, developers can customize LawPort, includes customization-related FAQs, and ends with a section that details all of the LawPort Style classes, including their respective class names and class attributes.
- An Index completes this guide.

Conventions used in this guide

The following is a list of the typographical and text frame conventions used in this guide.

Blue Text	Cross-references are distinguished by blue text and are used to guide readers to further useful sections.
`Courier`	Courier is used to distinguish code examples.
Best Practices	This text frame is used to convey helpful hints to LawPort users. These hints try to illustrate how one can obtain the best performance from LawPort.
	This text frame contains information that is meant only for users of the Traditional Intranet Slice. Most often these frames will indicate which sections in this guide do not apply to the Traditional Intranet Slice.

Technical Support

For technical assistance with your LawPort application, please contact your firm’s LawPort administrator.

Your comments on this guide

Please send your comments on this guide to techpubs@svtechnology.com. Your comments are useful to us when we revise this guide.

1 Simple API

Chapter Contents:

Overview of the LawPort Simple API

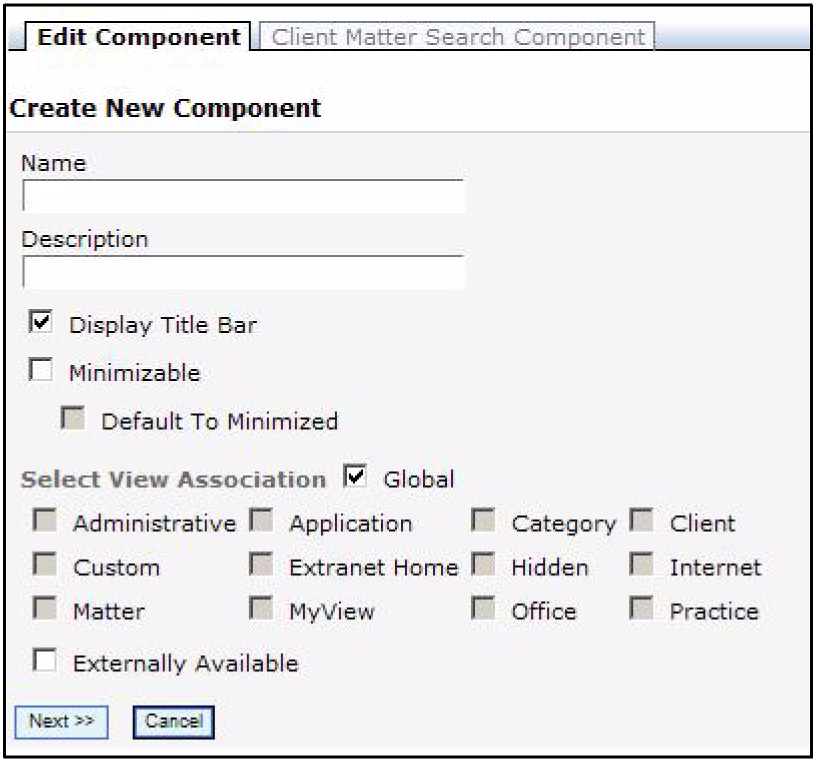
By creating a custom URL component using LawPort's component administration, developers can include properties from the LawPort Simple API in their own external applications. When you create a custom URL component, you can assign it a set of LawPort properties. These properties allow the LawPort application to sync with external applications. You can then include these newly configured external applications in views throughout LawPort. For more information on the available component properties, refer to Table 1-1 in the Component Properties section of this chapter.

Creating a custom URL Component

Follow the steps below to learn how to integrate LawPort properties with your own applications.

1. Navigate to the URL component page by choosing System>New Components>URL. The figure below displays the Create New Component form.

Figure 1-1 Create New Component form



Create New Component

Name

Description

☑ Display Title Bar

□ Minimizable

□ Default To Minimized

Select View Association ☑ Global

□ Administrative	□ Application	□ Category	□ Client
□ Custom	□ Extranet Home	□ Hidden	□ Internet
□ Matter	□ MyView	□ Office	□ Practice

□ Externally Available

2. Give your new component a name and description by filling in the Name and Description fields.

3. Deselect Display Title Bar if you do not want your component to display a title bar. If you deselect this option the Minimizable and Default to Minimized will not be selectable, as a title bar is needed for minimizing components.

4. Check the Minimizable option, if you want your component to be minimizable in the view. If you check this option, the Default to Minimized option will be selectable. Check this option if you want your component to appear minimized in the view.

5. In Select View Association leave Global checked, or uncheck it and select one or more view associations.

6. Select a display height from the list in the Display Height drop-down.

7. Click Next.

8. Enter the URL of your external application in the URL field.

9. Select Post from the Method drop-down. This will transfer the properties you select to your application.

10. Select the properties you would like to include from those listed in the Additional Component Properties group. Refer to the section below for more information on component properties.

Note: *There is no limit to how many properties you can select.*

11. Click Submit when you are finished.

Note: *If you do not wish to retain the selections you just made for your new component, click the reset button to clear the fields and start over.*

Component Properties

The following table describes all of the component properties and their conceivable uses.

Table 1-1 Component Properties

Parameter	**Description**	**Use**
LPBillingID	Unique ID for billing that is associated with the logged-in user.	Use the Billing ID to get more info about the logged-in LawPort user and external financial applications.
LPContactID	Unique ID for the currently logged-in user.	Use the ContactID to get more information about the logged-in LawPort user.
LPNetworkID	Unique ID for the current user's network. Also, can be used as the login name for the user. Name user uses to login to LawPort.	Use the NetworkID to identify the currently logged-in user to other applications that use network logins.
LPViewName	The name of the current view.	Used to indicate to the user which view is currently displayed.
LPViewTypeID	The unique ID for the view within its ViewType. Combined with a ViewType, this will uniquely identify a view.	Combine the ViewTypeID with a ViewType to request more information from LawPort about the view, such as the component layout, the color scheme or the contact list.
LPPubTargetType	The target type of the publishing target to which this view subscribes.	Combine the PubTargetType with the PubTargetID to request more information from LawPort about the PubTarget to which this view subscribes, such as a list of published content.
LPEmployeeID	The ID that represents this user in the firm's HR system.	Use the EmployeeID to get more information about the user from the firm's HR system.
LPPreferredName	The logged-in user's name.	Used to display the currently logged-in user's name.

Table 1-1 Component Properties

Parameter	Description	Use
LPViewType	The type of view for which the possible values are: site, category, client, matter, MyView and application.	Combine the ViewTypeID with a ViewType to request more information from LawPort about the view, such as the component layout, the color scheme or the contact list.
LPPubTargetName	The name of the publishing target to which this view subscribes.	Used to display the current view name.
LPPubTargetID	The ID of the publishing target to which this view subscribes.	Combine the PubTargetType with the PubTargetID to request more information from LawPort about the PubTarget to which this view subscribes, such as a list of published content.

2 Designing Custom URL Components

Chapter Contents:

Overview of custom URL components

This section is designed to help developers create custom URL components for LawPort. It provides suggestions on how to prepare your component’s design; offers code samples for you to incorporate into your custom URL component; and ends with a section on troubleshooting, providing you with a checklist to help you resolve any issues you may have when integrating your custom component into LawPort.

Planning and designing custom URL components

To help you in the component creation process, we recommend that you write a specification that outlines your component’s requirements, such as its database dependencies, navigation and content. Your specification can include a wireframe(s), a visual prototype of your custom component, as well as a workflow: the anticipated number and order of interactions a user may have with your component’s user interface.

When designing a component’s interface, try to keep the component’s design simple; create components with interfaces that require little if any explanation or documentation and minimal user interaction.

Also, consider your component’s load-time; know that the view to which the component is loading loads only as fast as its slowest component.

Creating styles for your custom URL components

You can store any style classes you want to apply to your custom URL components in the CustomStyles.css. This style sheet can be found in your *“root”* folder under the *“includes”* folder. When you first open your CustomStyles.css, it should be empty. If you want your custom URL components to include some of the styles that the LawPort components include, you can copy and paste the ones you want from the lawport.css into your CustomStyles.css.

> ***Note:*** *You should not alter the styles in the lawport.css, as any changes you make to them will be overwritten during upgrades. In contrast, you can (i.e., add, edit and delete your style classes) the CustomStyle.css, as it is not overwritten during upgrades. Also, keep in mind that the classes in CustomStyle.css will override any corresponding classes in the lawport.css. Your components will always adhere to the style classes you designate in CustomStyle.css.*

Adding LawPort Styles to your custom components

This section provides you with the header and table HTML code that is used to format LawPort components. We included this code so that you could add it to your own custom URL components. By using the header HTML, your components will display the same header design that the LawPort components display, allowing you to maintain a consistent look among components throughout LawPort. In addition, you can apply the table HTML to your custom URL components search results. When you apply the table HTML to your component, your component’s results (if it is designed to output results) will behave and look like LawPort component search results.

LAWPORT HEADER CODE

To add the LawPort header to your custom URL component, apply the following HTML code:

```
<SPAN class="componentHeaderName">Component Title</SPAN>
```

LAWPORT TABLE CODE

To add the LawPort table sort to your custom URL component, apply the following HTML code:

```
<table border="0" class="tablesort" onClick="sortColumn(event)">
<thead>
<tr>
   <td>column1</td>
   <td>column2</td>
   <td>column3</td>
</tr>
</thead>
<tr>
   <td>4</td>
   <td>B</td>
   <td>2</td>
</tr>
<tr>
   <td>1</td>
   <td>F</td>
   <td>11</td>
</tr>
</table>
<script language="JavaScript">initSortTable();</script>
```

Customizing the behavior of your custom URL component

Custom URL components behave like LawPort components when they are added to views in LawPort. If a custom URL component is refreshed, which is generally initiated through an interaction between user and component, the entire view, correspondingly, is refreshed. Likewise, if a view is refreshed, all components, including custom URL components, are refreshed.

However, if you want your custom URL component to refresh without refreshing the view in which it is displayed, minimizing user disruption and overall load time, you can perform one of the following:

- Design a component that opens in a new browser window.
- Add your component to an IFRAME.
- Design a component that displays its results in the same browser window.

For information on how to design any of the component behaviors listed above, refer to the following sections.

New Browser Windows for custom components

This section provides you with the code needed to apply the new window functionality to your custom component, and provides you with a checklist to help you troubleshoot new window functionality.

NEW WINDOW CODE

To add new window code to your custom component, apply the following HTML code:

```
function openWin(winName){
newWin = window.open( ‘<href>’ , ‘<window name>’ , '
toolbar=no,location=yes,scrollbars=yes,width=380,height=410' );
    newWin.focus();
}
```

> ***Note:*** *The window open function requires an HREF for the page (String), a name for the window (String), and a series of options (String).*

NEW WINDOW CHECKLIST

When adding a new window to a custom URL component, you should ask the following:

- Does the new window close when LawPort browser window is closed?
- Should a scroll bar be applied to the new window?
- Is the new window resizable?
- Does the new window appear in front of the LawPort browser?

Components in IFRAMES

A component that is displayed in an IFRAME can be refreshed without causing the view in which it is displayed to be refreshed, minimizing user disruption and overall load time. The IFRAME component is self-contained, precluding the need for interaction between the component and the LawPort Server. The only time that the IFRAME component communicates with the LawPort server is when the view in which the IFRAME component is displayed first loads.

IFRAME CODE FOR COMPONENTS

To add the IFRAME to your custom URL component, apply the following HTML code:

```
<IFRAME SRC=”Content.html” WIDTH=100%>
Content.html Example:

Content.html
<HTML>
<link rel="STYLESHEET" type="text/css" href="http://serverURL/includes/
lawport.css">
    <HEAD>
    </HEAD>
    <BODY>Hello World
    </BODY>
</HTML>
```

Note: *Only the IFRAME is registered in LawPort. The IFRAME refers to the component which is stored elsewhere and only referenced in the IFRAME component.*

STYLE SHEET FOR COMPONENT THAT USE IFRAMES

Because the IFRAME separates the content from the page itself, it does not automatically include the style sheet. To include style sheet code refer to the following section.

STYLE SHEET CODE

To add the style sheet code to your custom URL component that uses an IFRAME, apply the following HTML code:

```
<link rel="STYLESHEET" type="text/css" href="http://serverURL/includes/
lawport.css">
```

```
<link rel="STYLESHEET" type="text/css" href="http://serverURL/clientfiles/
customLawportStyle.css">
```

IFRAME CHECKLIST

When adding an IFRAME to a custom URL component, you should ask the following:

- Are style sheets applied to the IFRAME?
- Does the IFRAME appear in a LawPort view?
- Are size values applied (height & width) to the IFRAME?
- Does the IFRAME refresh without refreshing the view?

Displaying component results in the same page

You can design a custom URL component in such a way that the results it generates display in the same page, also known as the referring view. This design allows you to show results in the same page without loosing the LawPort view's header and navigation. This section provides you with the same page form code to add to your custom component.

SAME PAGE FORM CODE

To add form code to your custom component so that your results display in the same page, apply the following HTML code:

```
<form name="userform" onsubmit="refreshLawportView('http://devserverURL/
example.cfm', 150, 'userfrom')" method="post">
<input type="text" name="test"><br>
<input type="submit" name="submit" value="submit">
</form>
```

REQUIREMENTS FOR SAME PAGE RESULTS

The following list outlines the requirements that need to be met in order for results to appear in the same page:

- The “post” option needs to be selected in the component administration options.
- The component code must contain a component name.

- Within the component code, the HTML form must use the post method.
- The HTML form's "action" attribute will not be used. The component's new URL will be determined by the JavaScript as described above.
- Displaying content using this method is done by referring to a JavaScript method that requires the following three parameters:
    - The form action URL.
    - The ID of the referring component, which can be obtained by referring to the component administration menu (insert screenshot of component administration that includes the ID column).
    - The name of the form.

Adding and displaying custom URL components in LawPort

After you create your custom URL component, you can then add it to LawPort. Refer to the URL component section of the LawPort Administrator Guide's Component Administration chapter for information.

Troubleshooting custom URL components

Refer to the following when troubleshooting a custom URL component:

- Has a copy of the component been added to the web server with share access to the LawPort server?
- Have authentication issues regarding the server where this component is hosted been resolved? The URL location must allow anonymous access.
- Has the custom URL component been added to LawPort through component administration as a URL type component?
- Have you tested to make sure that your component's URL can be accessed via a web browser?
- Is the URL path correct? Remember to include http://<server name>/<component name>.

- Has the component been associated with a template or view?
- Have references to style sheets been included?

3 Web Services

Chapter Contents:

Overview of web services

Web services are designed to provide access to parts of the LawPort Legal Object Model, a framework of business objects that encapsulates legal specific data, such as client and matter, practice groups, and legal documents. From your own applications, you can make calls to these web services to request information from LawPort and to publish content to LawPort. These web services communicate with your application using a SOAP standard, a language independent XML based protocol, allowing you to convert data returned from a web service into your own applications easily.

You can take advantage of these web services in many ways. For example, using the Matter web service, you can extract matter names associated with a particular LawPort view and incorporate those names into a matter lookup application. Also, rather than executing searches from within the LawPort application, you can use these web services to search LawPort from your own applications. Use these search services for both large, comprehensive extractions and for individual lookups on specific items.

This chapter provides an overview of web services, defines them, and provides a detailed list of all of the request parameters and return values associated with each web service.

Types of web services

Below is a list of all of the web services offered.

- Categories Search web service
- Client Search web service
- Group Search web service
- People Search web service
- PubItem Delete web service
- Matter Search web service
- PubItem Search web service
- Publish PubItem web service
- Site Hierarchy Search web service
- Site Search web service

Web services technology

Web services use the SOAP protocol to transmit data. As SOAP is platform and vendor neutral, and supported by major platforms (like Java and .NET), programmers need not know anything about SOAP to call the available web services.

How to authenticate your web service requests

For authentication purposes, all requests to LawPort web services must include a user name and password in the SOAP header.

AUTHENTICATION FORMAT IN THE SOAP HEADER

The client must specify the SOAP header for authentication in the following manner:

```
<authenticate>
  <username>user1</username>
  <password>password1</password>
</authenticate>
```

The values (“user1” and “password1”) specified in the header are verified against the values specified in the LawPort config file, which is under the “webservice-password-config” section in the LawPort config file.

SPECIFYING WEB SERVICE AUTHENTICATION IN THE LAWPORT CONFIG FILE

Each user and user’s password must be specified in the following manner:

```
<webservice-password-config>
  <user>
    <username>user1</username>
    <password>password1</password>
  </user>
  <user>
    <username>user2</username>
    <password>password2</password>
  </user>
  <user>
```

```
    <username>userX</username>
    <password>passwordX</password>
  </user>
</webservice-password-config>
```

Note: *A LawPort administrator, having rights to the LawPort config, can make the necessary authentication adjustments: add or modify user names and passwords.*

EXAMPLE OF AUTHENTICATION IN LAWPORT CONFIGURATION FILE

The example below shows LawPort administrators how to authenticate users in the LawPort config file:

```
<webservice-password-config>
 <user>
    <username>ssnead</username>
    <password>mycatsname</password>
  </user>
  <user>
    <username>jdoe</username>
    <password>mydogsname</password>
  </user>
</webservice-password-config>
```

Note: *There is no limit to the number of users who may be specified.*

Viewing WSDLs for LawPort web services

WSDL (Web Service Description Language) describes a web service. Many languages have tools to automatically generate client code from a WSDL file. To view a WSDL that is associated with a LawPort web service, perform the following:

1. Browse to: http://*yourservername*/LawPort/Services/

2. Click on a LawPort web service in the list of web service that appears.

The WSDL file for the web service you selected will appear.

Request parameters and return values

In the sections below, you will find a section for each one of the web services. Included in each section is a brief description of the web service followed by tables that list and describe the request parameters and return values for that web service.

This section contains the following:

Note: *Not all web services have both request parameters and return values.*

Category Search web service

Function: Gets the name and ID for a category and all of its child categories.

The sections listed below provide detailed information on the request parameters and the return values associated with the Category Search web service.

Request Parameters for Category Search

Table 3-1 lists and describes the request parameters associated with Category Search.

Table 3-1 Category Search request parameters.

Request Parameter	Format	Description
CategoryId	integer	Category’s unique integer ID in LawPort; if omitted returns the entire category tree.
validateParameters	boolean	Determines whether the request parameters will be validated before executing the service. If true, then invalid parameters will cause an error.

Return Values for Category Search

Table 3-2 lists and describes the return values associated with Category Search.

Table 3-2 Category Search return values.

Return Value	Format	Description
id	integer	Category’s unique integer ID in LawPort.
name	string	Name of the category.
parentId	integer	Unique ID of the parent category.

Client Search web service

Function: Searches for clients and returns a list of clients and their details.

The sections listed below provide detailed information on the request parameters and the return values associated with the Client Search web service.

- "Request Parameters for Client Search" on page 21
- "Return Values for Client Search" on page 22
- "Details of the Array for Client Search" on page 23

Request Parameters for Client Search

Table 3-3 lists and describes the request parameters associated with Client Search.

Table 3-3 Client Search request parameters.

Request Parameter	Format	Description
clientNumber	string	The Client number is imported from the firm's financial system. The search will get the Client that matches this number.
clientName	string	The search will get the Clients whose names that include this text.
clientIdList	string	A comma-delimited list of integer IDs. The search will get Clients whose IDs match these Client IDs.
contactIdList	string	A comma-delimited list of integer IDs. The search will get Clients whose Members include Contacts with these Contact IDs.
categoryIdList	string	A comma-delimited list of integer IDs. The search will get Clients associated to Categories that match these Category IDs.
similarClientIdList	string	A comma-delimited list of integer IDs. The search will get Clients that share at least one Category with the Clients whose IDs match these Client IDs.
matterIdList	string	A comma-delimited list of integer IDs. The search will get Clients whose Matters include Matters with these Matter IDs.

Table 3-3 Client Search request parameters.

Request Parameter	Format	Description
matterNumber	string	The Matter number is imported from the firm's financial system. The search will get the Clients that have Matters that match this number.
matterName	string	The search will get the Clients that have Matters whose names that include this text.
openDate	string	Gets Clients opened after the given date.
activeClientsOnly	boolean	Only get Clients that are flagged as Active
startRow = 1	integer	Rather than return the full results begin at a row other than the first row.
returnRows = -1	integer	The maximum number of rows to return. A value of -1 indicates to return all rows.
customFields	customField array	An array of name-value pairs that represent a search on Custom Fields. See Table 3-5 for this array's details.
MappedCustomField	string	Gets clients associated with a particular imported category. This argument requires this specific format: importSource=importedCategoryID.
validateParameters	boolean	Determines whether the request parameters will be validated before executing the service. If true, then invalid parameters will cause an error.

Return Values for Client Search

Table 3-4 lists and describes the return values associated with Client Search.

Table 3-4 Client Search return values.

Return Values	Format	Description
id	integer	Unique integer ID of the Client in LawPort.
name	string	Name of the Client.
number	string	Client number. Imported from the firm's financial system.
openDate	string	Date the Client was opened.

Table 3-4 Client Search return values.

Return Values	Format	Description
closeDate	string	Date, if any, the Client was closed.
customFields	customField array	List of custom fields and their values for the Client. See Table 3-5 for this array's details.

Details of the Array for Client Search

Table 3-5 provides the formats and descriptions of the values associated with customFields, which is both a request parameter and a return value.

Table 3-5 **customFields** values.

customFields Values	Format	Description
name	string	Name of the custom field for this client.
value	string	Value of the custom field for this client.

Matter Search web service

Function: Searches for matters and returns a list of matters and their details.

The sections listed below provide detailed information on the request parameters and the return values associated with the Matter Search web service.

- "Request Parameters for Matter Search" on page 24
- "Return Values for Matter Search" on page 25
- "Details of the Array for Matter Search" on page 26

Request Parameters for Matter Search

Table 3-6 below lists and describes the request parameters for Matter Search.

Table 3-6 Matter Search request parameters.

Request Parameters	Format	Description
matterIdList	string	A comma-delimited list of integer IDs. The search will get Matters that match these Matter IDs.
matterName	string	The search will get the Matters whose names include this text.
matterNumber	string	The Matter number is imported from the firm's financial system. The search will get Matters that match this number.
clientIdList	string	A comma-delimited list of integer IDs. The search will get Matters with Clients that match these Client IDs.
contactIdList	string	A comma-delimited list of integer IDs. The search will get Matters whose Members include Contacts with these Contact IDs.
categoryIdList	string	A comma-delimited list of integer IDs. The search will get Clients associated to Categories that match these Category IDs.
siteIdList	string	Gets Matters associated to Practice Groups and/or Offices with Site IDs in this list.
openDate	string	Gets Matters opened after the given date.
similarMatterIdList	string	A comma-delimited list of integer IDs. The search will get Matters that share at least one Category with the Matters whose IDs match these Matter IDs.
activeMattersOnly	boolean	Gets only Matters that are flagged as Active.
startRow = 1	integer	Rather than return the full results begin at a row other than the first row.
returnRows = -1	integer	The maximum number of rows to return. A value of -1 indicates to return all rows.
clientIdList	string	A comma-delimited list of integer IDs. The search will get Clients whose IDs match these Category IDs.

Table 3-6 Matter Search request parameters.

Request Parameters	Format	Description
customFields	customField array	An array of name-value pairs that represent a search on Custom Fields. See Table 3-8 for this array's details.
MappedCustomField	string	Gets matters associated with a particular imported category. This argument requires a specific format: importSource=importedCategoryID.
validateParameters	boolean	Determines whether the request parameters will be validated before executing the service. If true, then invalid parameters will cause an error.

Return Values for Matter Search

Table 3-7 lists and describes the return values associated with Matter Search.

Table 3-7 Matter Search return values.

Return Values	Format	Description
matterId	integer	The Matter's unique integer ID in LawPort.
number	string	The Matter number. Imported from the firm's financial system.
name	string	The Matter's name.
description	string	The Matter's description.
openDate	string	Date the Matter was opened.
closeDate	string	Date the Matter was closed, if any.
clientId	integer	ID of this Matter's Client.
attorneyGroupId	integer	ID of the group that contains the Matter's list of attorneys.
memberGroupId	integer	ID of the group that contains the Matter's list of members.
timekeeperGroupId	integer	ID of the group that contains the Matter's list of timekeepers.
customFields	customField array	List of custom fields and their values for this Matter. See Table 3-8 for this array's details.

Details of the Array for Matter Search

Table 3-8 provides the formats and descriptions of the values associated with customFields, which is both a request parameter and a return value.

Table 3-8 **customFields** values.

customFields Values	Format	Description
name	string	Name of the custom field for this matter.
value	string	Value of the custom field for this matter.

PubItem Search web service

Function: Searches for PubItems and returns a list of PubItems and their details.

The sections listed below provide detailed information on the request parameters and the return values associated with the PubItem Search web service.

- "Request Parameters for PubItem Search" on page 26
- "Return Values for the PubItem Search" on page 28
- "Details of the Arrays for PubItem Search" on page 29

Request Parameters for PubItem Search

Table 3-9 lists and describes the request parameters associated with PubItem Search.

Table 3-9 PubItem Search request parameters.

Request Parameters	Format	Description
pubItemIdList	string	A comma-delimited list of integer IDs. The search will get PubItems that match these PubItem IDs.
title	string	Get PubItems with a title that contains this text.
description	string	Get PubItems with a description that contains this text.
pubItemTypeList	string	A comma-delimited list of PubItem types (News, Event, etc.) Get PubItems of particular types.
categoryIdList	string	A comma-delimited list of integer IDs. Get PubItems associated to the Category IDs in this list.

Table 3-9 PubItem Search request parameters.

Request Parameters	Format	Description
clientIdList	string	A comma-delimited list of integer IDs. Get PubItems associated to the Client IDs in this list.
matterIdList	string	A comma-delimited list of integer IDs. Get PubItems associated to the Matter IDs in this list.
siteIdList	string	A comma-delimited list of integer IDs. Get PubItems associated to the Site IDs in this list.
authorId	integer	Get PubItems whose Author matches this Contact ID.
publisherId	integer	Get PubItems whose Publisher matches this Contact ID.
expirationStatus	string	Restrict result depending on if the PubItem is expired. The following values are allowed: 0 = unexpired only, 1 = all, 2 = expired only.
modifiedBeforeDate	string	Get PubItems modified before the given date.
modifiedAfterDate	string	Get PubItems modified after the given date.
storageType	integer	Get PubItems of a particular storage type. The following values are allowed: 1 = direct input, 2 = uploaded file, 3 = DMS, 4 = URL, 5 = DMS folder, 6 = InterAction folder
startRow = 1	integer	Rather than return the full results begin at a row other than the first row.
returnRows = -1	integer	The maximum number of rows to return. A value of -1 indicates to return all rows.
customFields	customField array	An array of name-value pairs that represent a search on Custom Fields. See Table 3-15 for this array's details.
validateParameters	boolean	Determines whether the request parameters will be validated before executing the service. If true, then invalid parameters will cause an error.
primarySortOrder	string	Indicates the primary sort order of the results.
primarySortDirection	string	Indicates the primary sort direction of the results.
secondarySortOrder	string	Indicates the secondary sort order of results.
secondarySortDirection	string	Indicates the secondary sort direction of results.

Return Values for the PubItem Search

Table 3-10 lists and describes the return values associated with PubItem Search.

Table 3-10 PubItem Search return values.

Return Values	Format	Description
id	integer	Unique integer ID for a PubItem in LawPort.
type	string	PubItem type of the PubItem.
title	string	Title of the PubItem.
description	string	Description of the PubItem.
content	string	Content of the PubItem.
authorId	integer	Contact ID of the PubItem's author.
publisherId	integer	Contact ID of the PubItem's publisher.
external	boolean	Is the PubItem available to External Contacts? (true/false)
creationDate	string	Date the PubItem was created.
modificationDate	string	Date the PubItem was last edited.
startDate	string	Date the PubItem will become visible in LawPort
endDate	string	Date the PubItem will cease to be visible in LawPort.
storageType	integer	An integer value that indicates the format of the PubItem's content. The following values are allowed: 1 = direct input, 2 = uploaded file, 3 = DMS, 4 = URL.
categoriesBeans	category-Bean array	Categories associated with this PubItem. See Table 3-11 for more details.
clientsBeans	clientBean array	Clients associated with this PubItem. See Table 3-12 for more details.
mattersBeans	matterBean array	Matters associated with this PubItem. See Table 3-13 for more details.
sitesBeans	siteBean array	Sites associated with this PubItem. See Table 3-14 for more details.
customFields	customField array	Custom fields associated with the PubItem. See Table 3-15 for more details.

Details of the Arrays for PubItem Search

The tables listed below provide the formats and descriptions of the values associated with each array (see Table 3-10) that is returned from, or, in the case of the customField array, requested by, People Search web service.

- Table 3-11 details the **categoriesBeans** array values.
- Table 3-12 details the **clientsBeans** array values.
- Table 3-13 details the **mattersBeans** array values.
- Table 3-14 details the **sitesBeans** array values.
- Table 3-15 details the **customFields** array values.

categoriesBeans array

Table 3-11 provides the formats and descriptions of the values associated with the catgegoriesBeans.

Table 3-11 **categoriesBeans** values.

categoriesBeans Values	Format	Description
id	integer	Unique ID number of category.
name	string	The category name.

clientsBeans array

Table 3-12 provides the formats and descriptions of the values associated with the clientsBeans.

Table 3-12 **clientsBeans** values.

clientsBeans Values	Format	Description
id	integer	Unique client ID.
name	string	The client name.
number	string	Client number.

mattersBeans array

Table 3-13 provides the formats and descriptions of the values associated with the mattersBeans.

Table 3-13 **mattersBeans** values.

mattersBeans Values	Format	Description
id	integer	Unique matter ID.
email	string	The matter name.
number	string	Matter number.

sitesBeans array

Table 3-14 provides the formats and descriptions of the values associated with the sitesBeans.

Table 3-14 **sitesBeans** values.

sitesBeans Values	Format	Description
id	integer	Unique ID number of site.
name	string	The name of site.

customFields array

Table 3-15 provides the formats and descriptions of the values associated with the customFields.

Table 3-15 **customFields** values.

customFields Values	Format	Description
name	string	Name of the custom field.
value	string	Value of the custom field.

Publish PubItem web service

Function: Publishes content to LawPort. The details of the PubItem, including its distribution and categorization information, are included in this service.

> ***Note:*** *"There are certain control strings which may be passed to LawPort via the Description field. These control strings are of the form*
>
> {LP_<name>=<value
>
> You should avoid the use of strings in this format in general use. For further information please contact your Engagement Manager"

Request Parameters for Publish PubItem

Table 3-16 lists and describes the request parameters associated with Publish PubItem.

Table 3-16 Publish PubItem request parameters.

Request Parameters	Format	Description
id	integer	ID of the PubItem that is being edited. If creating a new PubItem, the ID will be 0. Defaults to 0.
type	string	PubItem type of the PubItem.
title	string	Title of the PubItem.
description	string	Description of the PubItem.
content	string	Content of the PubItem.

Table 3-16 Publish PubItem request parameters.

Request Parameters	Format	Description
authorId	integer	Network ID of the PubItem's author. An invalid ID will cause an error.
external	boolean	Is the PubItem available to External Contacts? (true/false)
startDate	string	Date the PubItem will become visible in LawPort.
endDate	string	Date the PubItem will cease to be visible in LawPort.
storageType	integer	An integer value that indicates the format of the PubItem's content. The following values are allowed: 1 = direct input, 2 = uploaded file, 3 = DMS, 4 = URL.
content	string	The content of the PubItem. The following formats are required, depending on the PubItem's storage type. For storage type 1, the content must be valid HTML. For storage type 2, the content must be a valid path to a file. For storage type 3, the content must identify a document in a DMS, with the format "DatabaseName:DocumentNumber:DocumentVesion". For storage type 4, the content must be a URL, including the protocol (like http://).
categoryIdList	string	List of Category IDs that will be associated with this PubItem.
clientNumber	string	Number of the Client that will be associated with this PubItem. An invalid Client number will cause an error.
matterNumber	string	Number of the Matter that will be associated with this PubItem. An invalid Client number will cause an error.
siteIdList	string	List of Site IDs that will be associated with this PubItem.
validateParameters	boolean	Determines whether the request parameters will be validated before executing the service. If true, then invalid parameters will cause an error.
underRSSControl	boolean	Determines whether a PubItem that has been imported via the LawPort RSS (Rich Site Summary) module is still under RSS control. When this parameter is set to "true," all PubItem controls (e.g., editing) and inputs are disabled. When set to "false," PubItem con-trols and inputs operate normally (as the PubItem is no longer under RSS control).

Note: *The PubItem Publish web service does not include return values.*

Site Hierarchy Search web service

Function: Gets the entire tree of LawPort sites. Included with each site is its set of related categories.

The sections listed below provide detailed information on the request parameters and the return values associated with the Site Hierarchy Search web service.

- "Return Values for Site Hierarchy Search" on page 33
- "Details of the Array Return Values for Site Hierarchy Search" on page 34

> ***Note:*** *The Site Hierarchy Search web service does not include request parameters. To search for sites using request parameters, refer to "Site Search web service" on page 42.*

Return Values for Site Hierarchy Search

Table 3-17 lists and describes the return values associated with Site Hierarchy Search.

Table 3-17 Site Hierarchy Search return values.

Return Value	Format	Description
id	integer	View's unique integer ID in LawPort.
name	string	Name of the view.
memberGroupId	integer	Unique group ID of the site's member group.
adminGroupId	integer	Unique group ID of the site's admin group.
parentID	integer	Unique ID of the parent site.
type	string	Type of view (practice, administrative, office, etc.).
categoryBeans	category-Bean array	Describes a category.

Details of the Array Return Values for Site Hierarchy Search

Table 3-18 details the values associated with CategoryBeans that is returned from Site Hierarchy Search.

Table 3-18 Site Hierarchy Search CategoryBeans array.

CategoryBeans Values	Format	Description
id	integer	Unique ID of the site.
parentId	integer	Unique ID of the site's parent site.
name	string	Name of the site.

People Search web service

Function: Searches for people and returns a list of people and their details.

The sections listed below provide information on the request parameters and the return values associated with the People Search web service.

- "Request Parameters for People Search" on page 34
- "Return Values for People Search" on page 35
- "Details of the Array Return Values for People Search" on page 37

Request Parameters for People Search

Table 3-19 lists and describes the request parameters associated with People Search.

Table 3-19 People Search request parameters.

Request Parameters	Format	Description
returnRows = -1	integer	The maximum number of rows to return. A value of -1 indicates to return all rows.
startIndex = 1	integer	Rather than return the full results, begin at a row other than the first row.
barLicenseId	string	Search by bar license ID if available.
degreeId	string	Search by school degree ID if available.

Table 3-19 People Search request parameters.

Request Parameters	Format	Description
regionId	string	Search by region ID if available.
schoolId	string	Search by school ID if available.
primaryGroupOnly	boolean	Get only persons whose practice group is in the PracticeGroupIdList.
primaryDepartmentLead	boolean	Get only persons who are primary department leads in the DepartmentIdList.
practiceGroupLead	boolean	Get only persons who are practice group leaders in the PracticeIdList.
primaryOffice	boolean	Get only persons whose practice group is in the OfficeGroupIdList.
lastName	string	Last name of person.
phoneNumber	string	Phone number of person.
email	string	Email of person.
graduationYear	string	The year the person graduated.
practiceIdList	string	Practices that persons belong to.
departmentIdList	string	Departments that persons belong to.
firstName	string	First name of person.
phoneExtension	string	Phone extension.
netId	string	The Network ID (username) of the person.
jobTitleIdList	string	Job Titles that persons belong to.
officeIdList	string	Offices that persons belong to.
validateParameters	boolean	Determines whether the request parameters will be validated before executing the service. If true, then invalid parameters will cause an error.

Return Values for People Search

Table 3-20 lists and describes the return values, excluding the array return values (which are described in the following table), associated with People Search.

Table 3-20 People Search return values.

Return Values	Format	Description
firstName	string	First name.
title	string	Job title.
office	string	Office.
floor	string	Floor number.
defaultEmail	string	Primary email address.
netId	string	Network ID.
lastName	string	Last Name.
practiceGroup	string	Practice group name.
location	string	Location.
extension	string	Phone Extension.
defaultPhone	string	Primary phone number.
department	string	Department group name
addressInfo	addressinfo	List of addresses. See Table 3-21 for more details.
assignmentInfo	assignmentInfo array	List of secretarial assignments. See Table 3-22 for more details.
barLicenseInfo	barLicenseInfo array	List of bar licenses. See Table 3-23 for more details.
educationInfo	educationInfo array	List of education information. See Table 3-24 for more details.
emailInfo	emailInfo array	List of emails. See Table 3-25 for more details.
phoneInfo	phoneInfo array	List of phone numbers. See Table 3-26 for more details.

Details of the Array Return Values for People Search

The tables listed below provide the formats and descriptions of the values associated with each array that is returned from the People Search web service.

- Table 3-21 details the **addressInfo** array values.
- Table 3-22 details the **assignmentInfo** array values.
- Table 3-23 details the **barlicenseInfo** array values.
- Table 3-24 details the **educationInfo** array values.
- Table 3-25 details the **emailInfo** array values.
- Table 3-26 details the **phoneInfo** array values.

addressInfo array

Table 3-21 provides the formats and descriptions of the values associated with the addressInfo array.

Table 3-21 **addressInfo** values.

addressInfo Values	Format	Description
street1	string	Street name
street2	string	Street address line 2.
street3	string	Street address line 3.
city	string	City.
stateName	string	State.
countryName	string	Country.
postalCode	string	Postal/ZIP code.
stateOrProvinceName	string	State or Province name.
stateCode	string	State or Province name code.
countryCode	string	Country code.
region	string	Region.
mailingAddress	boolean	Returns “true” if this is the person’s mailing address.

assignmentInfo array

Table 3-22 provides the formats and descriptions of the values associated with the assignmentInfo array.

Table 3-22 **assignmentInfo** values.

assignmentInfo Values	Format	Description
name	string	Name of the Secretarial assignment.
extension	string	Phone extension of the assignee.

barlicenseInfo array

Table 3-23 provides the formats and descriptions of the values associated with the barlicenseInfo array.

Table 3-23 **barlicenseInfo** values.

barlicenseInfo Values	Format	Description
type	string	Type of license.
number	string	License number.
dateIssued	string	The date the license was issued.

educationInfo array

Table 3-24 provides the formats and descriptions of the values associated with the educationInfo array.

Table 3-24 **educationInfo** values.

educationInfo Values	Format	Description
school	string	Name of the school.
degree	string	Name of the degree.
graduationYear	string	Year graduated.

emailInfo array

Table 3-25 provides the formats and descriptions of the values associated with the emailInfo array.

Table 3-25 **emailInfo** values.

emailInfo Values	Format	Description
emailType	string	Type of email address.
email	string	The email address.

phoneInfo array

Table 3-26 provides the formats and descriptions of the values associated with the phoneInfo array.

Table 3-26 **phoneInfo** values.

phoneInfo Values	Format	Description
phoneType	string	Type of phone number.
phoneNumber	string	The phone number.
phoneExtension	string	The phone extension.

Group Search web service

Listed below are the two types of Group Search web services: the first allows you to search by Group ID and, the second, allows you to search by Group Name.

"Group Search by ID web service" on page 40

"Group Search by Name web service" on page 41

Note: *Both web services listed above have the same return values.*

GROUP SEARCH BY ID WEB SERVICE

Function: Searches for groups by ID and returns a list of groups and their details.

The sections listed below provide information on the request parameters and the return values associated with the Group Search by ID web service.

- "Request Parameters for Group Search by ID" on page 41
- "Return Values for Group Search by Name and by ID" on page 42

Request Parameters for Group Search by ID

Table 3-27 below lists and describes the request parameters associated with Group Search by ID.

Table 3-27 Group Search by ID request parameters.

Request Parameters	Format	Description
groupId	integer	The unique ID of the group
validateParameter	boolean	Determines whether the request parameters will be validated before executing the service. If true, then invalid parameters will cause an error.

GROUP SEARCH BY NAME WEB SERVICE

Function: Searches for groups by name and returns a list of groups and their details.

The sections listed below provide information on the request parameters and the return values associated with the Group Search by Name web service.

- "Request Parameters for Group Search by Name" on page 41
- "Return Values for Group Search by Name and by ID" on page 42

Request Parameters for Group Search by Name

Table 3-28 lists and describes the request parameters associated with Group Search by Name.

Table 3-28 Group Search by Name request parameters.

Request Parameters	Format	Description
groupName	string	The name of the group.
validateParameter	boolean	Determines whether the request parameters will be validated before executing the service. If true, then invalid parameters will cause an error.

Return Values for Group Search by Name and by ID

Table 3-29 lists and describes the return values associated with the Group Search by Name and by ID.

Table 3-29 Group Search return values.

Return Values	Format	Description
id	integer	The unique ID of the group.
name	string	The name of the group.
personIds	string	List of person IDs belonging directly to this group.
childGroupIds	string	List of child group IDs for this group.
allPersonIds	string	List of person IDs belonging directly to this group and all subgroups.

Site Search web service

Function: Searches for sites by site ID number and returns a list of sites and their details.

The sections listed below provide information on the request parameters and the return values associated with the Site Search web service.

- “Request Parameters for Site Search” on page 43
- “Return Values for Site Search” on page 43
- “Details of the Array Return Values for Site Search” on page 44

Request Parameters for Site Search

Table 3-30 lists and describes the request parameters associated with Site Search.

Table 3-30 Site Search request parameters.

Request Parameters	Format	Description
siteId	integer	The unique ID of the Site
validateParameters	boolean	Determines whether the request parameters will be validated before executing the service. If true, then invalid parameters will cause an error.

Return Values for Site Search

Table 3-29 lists and describes the return values associated with Site Search.

Table 3-31 Site Search return values.

Return Values	Format	Description
id	integer	The unique Site ID of the Site.
memberGroupId	integer	The Group ID for the Site’s member group.
parentId	integer	The unique Site ID of the Site parent Site.
adminGroupId	integer	The Group ID for the Site’s admin group.
name	string	The name of the site.
type	string	The type of site.
categoryBeans	category-Bean array	The list of categories associated with a site.

Details of the Array Return Values for Site Search

Table 3-32 details the values associated with the **CategoryBeans** array value that is returned from the Site Search.

Table 3-32 Site Search CategoryBeans array.

CategoryBeans Values	Format	Description
id	integer	The unique Category ID.
parentId	integer	The unique Category ID of the category’s parent.
name	string	The name of the category.

PubItem Delete web service

Function: Finds and deletes the PubItem(s) you want to delete.

Request Parameter for PubItem Delete.

Table 3-33 lists and describes all the request parameters associated with PubItem Delete.

> ***Note:*** *There are no return values associated with this web service.*

Table 3-33 PubItem Delete request parameters.

Request Parameters	Format	Description
pubItemId	integer	The unique Pubitem ID.
validateParameters	boolean	Determines whether the request parameters will be validated before executing the service. If true, then invalid parameters will cause an error.

4 Web Services for Internet Module

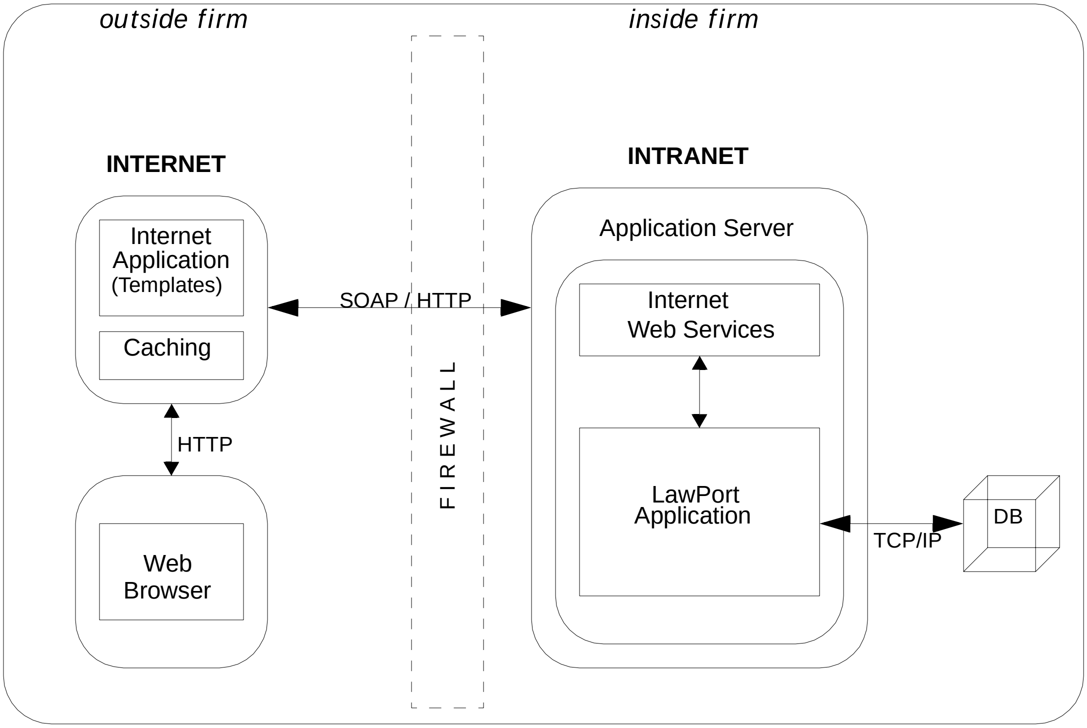
Chapter Contents:

Overview of the Web Services for Internet Module

The Web Services for Internet Module includes web services that allow you to leverage your intranet's data in the Internet site(s) you create. The Module's security settings make it possible for you to expose your intranet's data safely on the Internet. Because of the web services' flexibility, you can transmit intranet data to an Internet site hosted outside of your network on a different server anywhere on the Internet. The "out-of-the-box" ready Internet templates included in the Module can provide a comprehensive visual framework for an Internet site. You can adopt these templates explicitly, or you may decide to just use them as a guide when creating your firm's Internet site. View Figure 4-1 below for an illustration of intranet-to-Internet web services interaction.

This chapter introduces the Web Services Internet Module; describes each template included with this feature; shows how to set up and populate Internet views; and concludes with the Internet web services API, which provides detailed information about each web service.

Figure 4-1 Internet and intranet data sharing through web services



This chapter is written for the following audience:

- In-house developers
- Third party developers and design firms
- Existing LawPort 2.62 clients
- Existing LawPort 3.01 Internet clients

Getting Started with Web Services for Internet

This section provides you with the background information you need before you get started setting up your Internet site.

This section includes the following:

Internet views and the Web Services for Internet Module

The web services are at the core of the Web Services for Internet Module. These web services pull data from your intranet that has been secured to views of type Internet. So before you can view intranet data in your Internet site, you first need to create Internet views in your intranet site. You can create an Internet view and publish and add members to it just like you would with any other intranet view. Refer to "Who can access the Module?" below to see which role(s) you need to be a member of before you begin your set up.

Anatomy of the Web Services for Internet Module

The following list describes the pieces of functionality that together constitute the Web Services for Internet Module:

- **Internet Web Services.** Designed with particular attention to security, these services can safely expose your firm's intranet content in your firm's Internet site.
- **Internet Application.** A sample Internet web application made up of customizable templates that consume the data returned by the Internet web services. These templates are fully functional. When building your Internet site, you can adhere to these templates as they are, right out-of-the-box, or you may simply treat them as a starter-kit, modifying the templates as you see fit while you build. Written in Cold Fusion, these templates are easy to customize.

- **Intranet-based management tools.** A collection of tools contained in your firm's intranet that allow you to build and remotely secure the structure and content of your Internet sites. These intranet tools include:
    - **View Administration.** Allows you to create and administer Internet views.
    - **Publishing Wizard.** Allows you to publish content to Internet views.
    - **Role Administration.** Allows you to decide who has access to the roles described below.

Who can access the Module?

The following describes the roles that permit access to the various intranet tools associated with the Web Services for Internet Module:

- **Internet Publisher.** Members of this role can publish content to the Internet views of which they are a view publisher. In addition to belonging to this role, one must also be a member of the Publisher role so as to be able to publish to Internet views.

- **LawPort Administrator.** Members of this role can create, edit and delete Internet views; add, edit and delete any content published to any Internet view; and assign users to the Internet Publisher and View Publisher roles.

How are Internet web services secured?

The Web Services for Internet Module comes equipped with security that protects your firm's intranet data while allowing you to expose data on the Internet via the Internet web services.

Internet web services security includes two provisions: identification of Internet requests and Internet filters. They are described the following way:

One: Identification of Internet requests

- Your intranet will be configured to filter out all IP addresses that do not match your Internet server's IP address.

- Internet filters are applied to Internet web services to filter requests from your Internet server's IP address.

Two: Internet Filters

When called by the Internet application, the Internet web services will invoke a filter that behaves in the following ways:

- The SiteSearch web service will only return views of type Internet (i.e. Internet views).
- The PubItemSearch, PubItemFullTextSearch, and PubItemDetails web services will only return PubItems that have been published to views of type Internet. These web services also respect intranet PubItem security.
- The Person Search web service will only return persons that are members of views of type Internet.
- The NavBar Details web service will only return NavBar components that have been placed on views of type Internet.

Setting up your Internet site

This section commences with an introduction on the general layout of templates, and follows with detailed discussions of each template included in the Web Services Internet Module.

The templates are discussed in the following order:

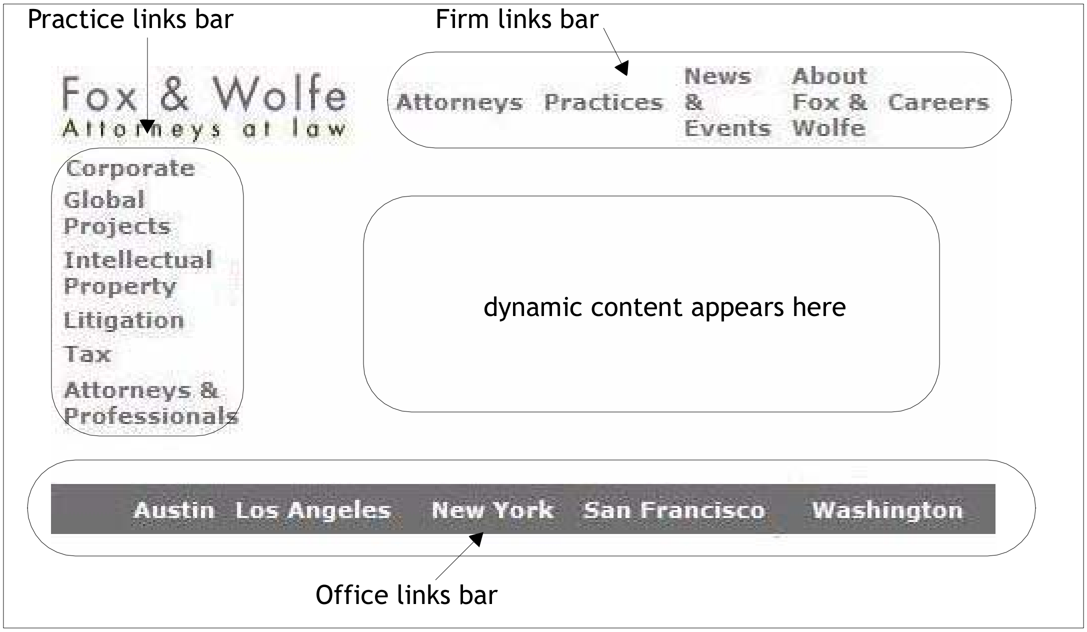
- “Home page template” on page 52
    - “Setting up a Home page” on page 53
    - “Internet Home page web services” on page 59
- “Practice page template” on page 60
    - “Setting up a Practice page” on page 61
    - “Practice page web services” on page 64
- “Office page template” on page 65
    - “Setting up the Office page” on page 66
    - “Office page web services” on page 68
- “People Search page template” on page 69
    - “Setting up the People Search page template” on page 70
    - “People Search page web service” on page 70
- “People Search Results template” on page 71
    - “Setting up the People Search Results template” on page 72
    - “People Search Results web service” on page 72
- “People Detail template” on page 73
    - “Setting up the People Detail template” on page 74
    - “People Detail web services” on page 74

The templates “global” layout

LawPort uses ColdFusion and Fusebox when implementing the Internet Module’s web services. Fusebox is a standard framework and methodology for building web-based applications. The templates discussed in this chapter adhere to Fusebox’s central concept of “global” layout. Figure 4-2 shows the basic structure of the templates. The top, left and bottom navigation bars make up the global layout of the templates; anything that appears inside this framework is dynamic. This layout repeats throughout all of the templates with the Home page, as it does not include left navigation, the only exception.

The left and bottom navigation is generated from the SiteSearch web service and the top navigation is generated from the NavBarDetails web service. The calls to these web services are located in the templates. A call uses Intranet ID numbers, such as Site ID and Container ID, to retrieve content that appears in the navigation bar.

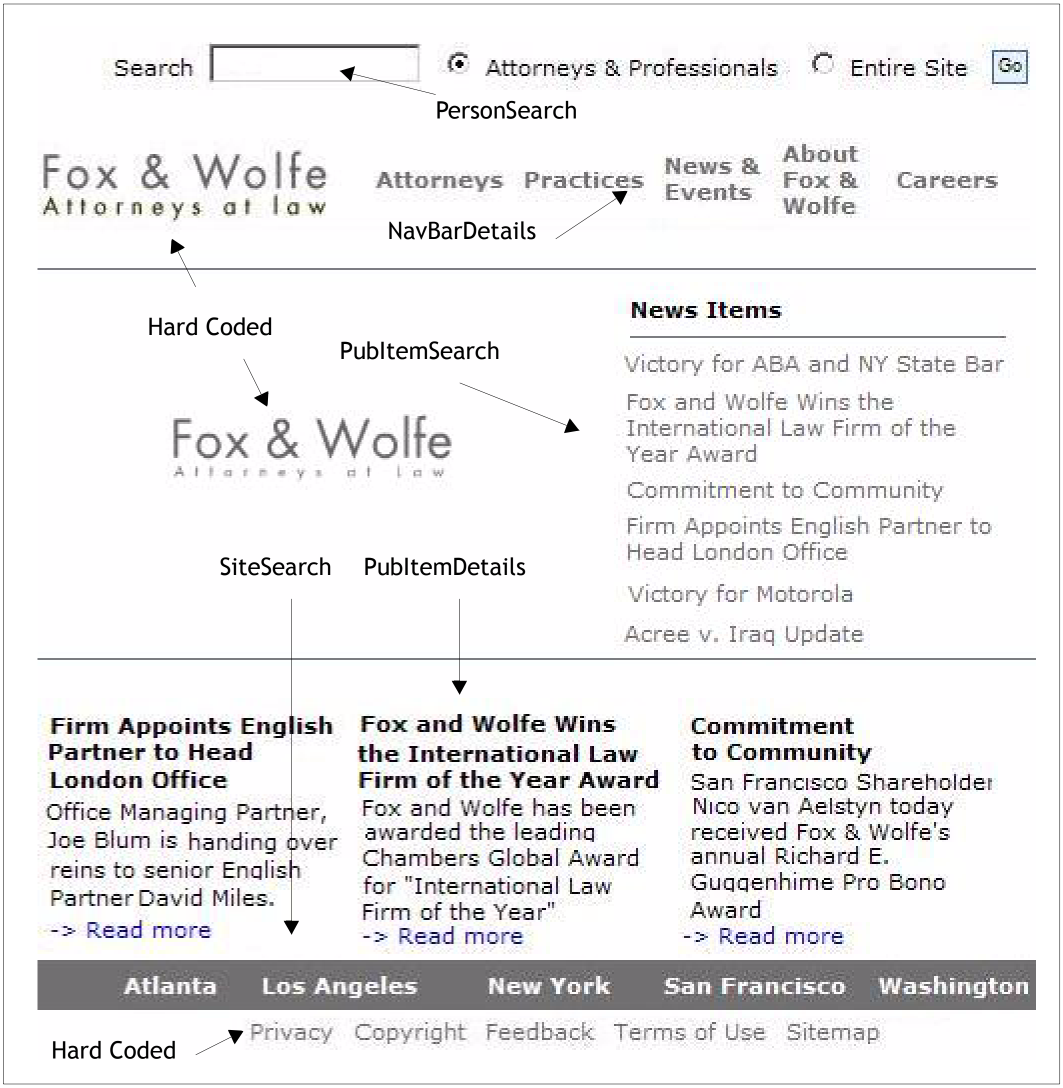
Figure 4-2 Global layout of templates



Home page template

Figure 4-3 displays a rendering of the Home page template. Arrows point to the web services that supply content to the page and to the hard-coded elements that you can modify to reflect your firm’s information.

Figure 4-3 Home page template



Search ● Attorneys & Professionals ○ Entire Site Go

Fox & Wolfe
Attorneys at law

Attorneys Practices News & Events About Fox & Wolfe Careers

Fox & Wolfe
Attorneys at law

News Items

Victory for ABA and NY State Bar

Fox and Wolfe Wins the International Law Firm of the Year Award

Commitment to Community

Firm Appoints English Partner to Head London Office

Victory for Motorola

Acree v. Iraq Update

Firm Appoints English Partner to Head London Office
Office Managing Partner, Joe Blum is handing over reins to senior English Partner David Miles.
-> Read more

Fox and Wolfe Wins the International Law Firm of the Year Award
Fox and Wolfe has been awarded the leading Chambers Global Award for "International Law Firm of the Year"
-> Read more

Commitment to Community
San Francisco Shareholder Nico van Aelstyn today received Fox & Wolfe's annual Richard E. Guggenhime Pro Bono Award
-> Read more

Atlanta Los Angeles New York San Francisco Washington

Privacy Copyright Feedback Terms of Use Sitemap

SETTING UP A HOME PAGE

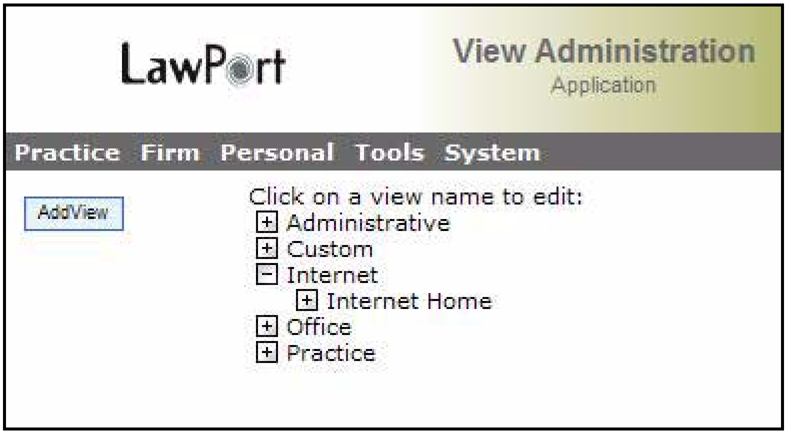
To set up your firm's home page using the Home page template, perform the following:

Creating your Internet Home view ...

To set up the intranet counterpart of your Internet Home page, perform the following:

1. In your intranet's view administration, define an Internet Home view, as in Figure 4-4. If you need help adding an Internet view, refer to Chapter 9 of the LawPort Administrator Guide.

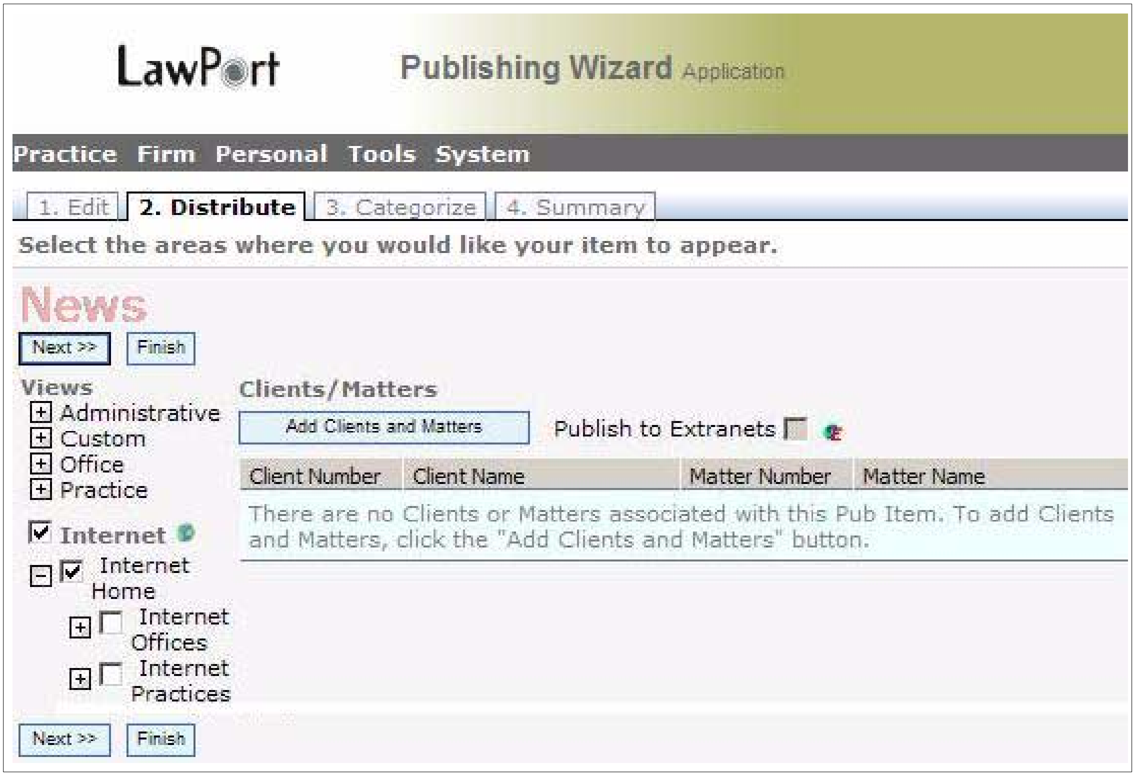
Figure 4-4 Defining an Internet Home view



Click on a view name to edit:
- ⊞ Administrative
- ⊞ Custom
- ⊟ Internet
  - ⊞ Internet Home
- ⊞ Office
- ⊞ Practice

2. To be able to display published items in your Home page, publish items to the Internet Home view you created. Figure 4-5 shows the Publishing Wizard's Distribute screen with the Internet Home view checked, specifying that the PubItem appear in the Internet Home view. For more details on how to publish content, refer to Chapter 5 of the LawPort User Guide.

Figure 4-5 Publishing content to the Internet Home view



Select the areas where you would like your item to appear.

Client Number	Client Name	Matter Number	Matter Name

There are no Clients or Matters associated with this Pub Item. To add Clients and Matters, click the "Add Clients and Matters" button.

Attorney and Professional search ...

To render the Attorneys and Professionals search, perform the following:

1. Add members and/or groups to your views of type Internet.

Note: *The Attorneys and Professionals search is a global search: it will search for members and groups that are added to any view of type Internet that you create.*

Entire Site search ...

To render the Entire Site search, perform the following:

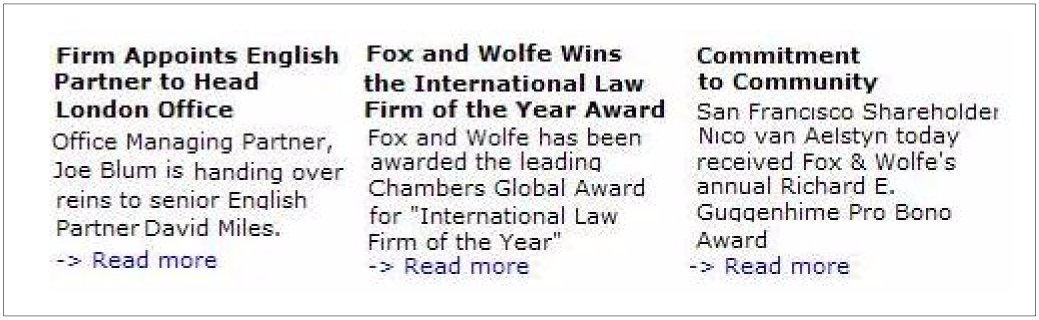
1. Publish content to views of type Internet. This search uses the PubItemFullTextSearch web service and will search for content published to all Internet views, not just Internet Home.

Displaying the details of published items ...

To display the details of PubItems, as in Figure 4-6 below, perform the following:

1. In the Home page template's call to the PubItemDetails web service, specify the three PubItem IDs that correspond to the three PubItems you want to display.

Figure 4-6 Home Page: Details of PubItems



Firm Appoints English Partner to Head London Office
Office Managing Partner, Joe Blum is handing over reins to senior English Partner David Miles.
-> Read more

Fox and Wolfe Wins the International Law Firm of the Year Award
Fox and Wolfe has been awarded the leading Chambers Global Award for "International Law Firm of the Year"
-> Read more

Commitment to Community
San Francisco Shareholder Nico van Aelstyn today received Fox & Wolfe's annual Richard E. Guggenhime Pro Bono Award
-> Read more

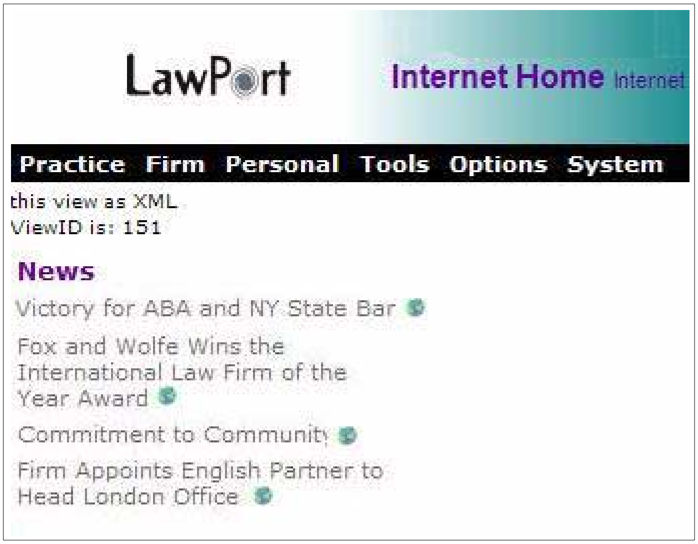
Displaying a list of published items ...

To render a list of News items, as in Figure 4-8 below, perform the following:

1. Publish PubItems of type News to your Internet Home view.

> ***Note:*** *To monitor the news exposed on your Internet site, as in Figure 4-8, you can add a News component to your Internet Home view, as in Figure 4-7 below. Also notice that the Internet Home view provides a Site ID (referred to as ViewID) in the top left of the view, which is where you can find Site IDs for all Internet views.*

Figure 4-7 News component in Internet Home view



this view as XML
ViewID is: 151

News

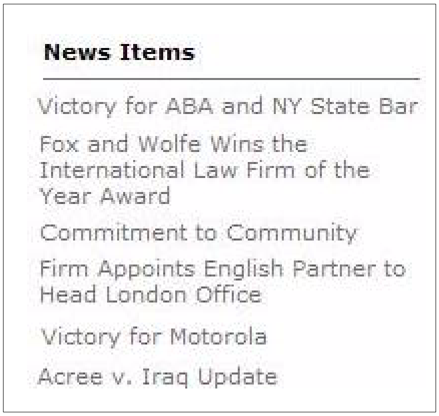
Victory for ABA and NY State Bar

Fox and Wolfe Wins the International Law Firm of the Year Award

Commitment to Communit

Firm Appoints English Partner to Head London Office

Figure 4-8 Home Page: List of News items



News Items

Victory for ABA and NY State Bar

Fox and Wolfe Wins the International Law Firm of the Year Award

Commitment to Community

Firm Appoints English Partner to Head London Office

Victory for Motorola

Acree v. Iraq Update

2. In your Home page template’s call to the PubItemSearch web service, specify the PubItemType (such as News) and the Site ID of the view from which you want content. However if you want content from all Internet views to appear, do not specify a Site ID.

Displaying Firm links ...

To render the Firm links across the top of the page, perform the following:

1. Add a Navigation Bar component to your Internet Home view. For information on how to create a custom navigation bar to add to your Internet Home view, refer to

2. Once your navigation bar is complete, you can specify the navigation bar’s Component ID in the Internet Home page template’s call to the NavBarDetails web service.

Note: *Besides including links to other Internet pages in your navigation bar, you can also include links to other destinations as well, such as a link to specific PubItem or a web site on the world wide web.*

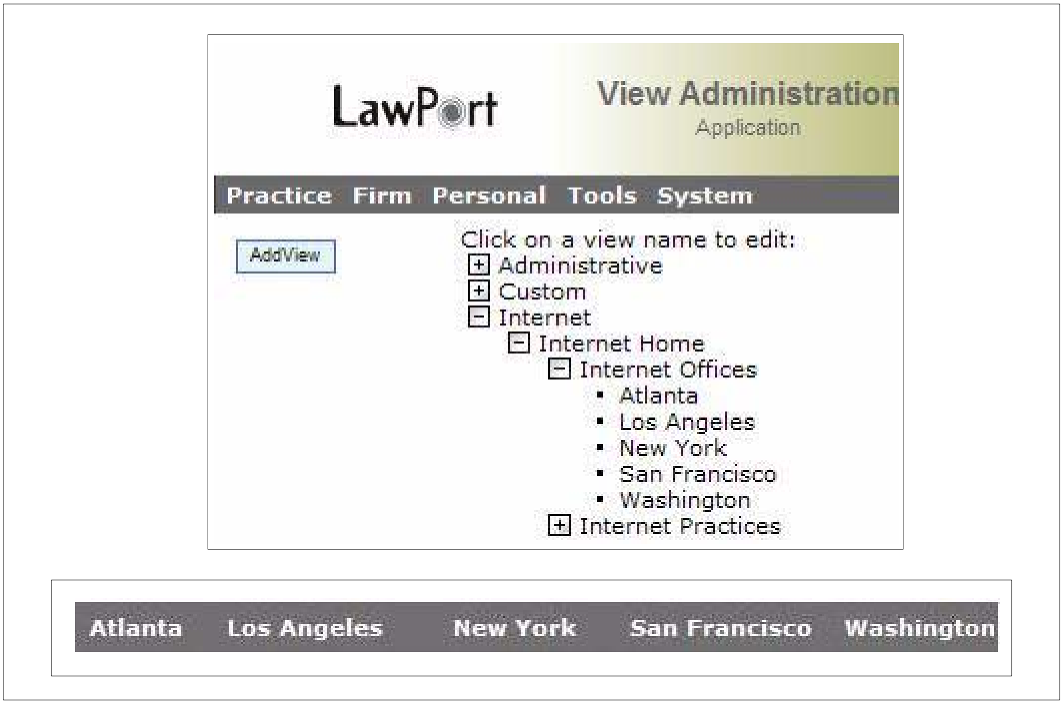
Displaying Office links ...

To render the Office links bar (displayed in the horizontal bar at the bottom of the page), perform the following:

1. In your intranet’s view administration, define an Internet Offices view and make it a child of Internet Home. If you need help adding an Internet view, refer to Chapter 3 of the LawPort Administrator Guide.

2. Also define an office view for each of your office links, making them a child of the Internet Offices view you defined in step 1. Figure 4-9 displays the Internet Office views in view administration in the top and the corresponding Office links in the bottom.

Figure 4-9 Defining Internet Office views in view administration



Click on a view name to edit:

- ⊞ Administrative
- ⊞ Custom
- ⊟ Internet
  - ⊟ Internet Home
    - ⊟ Internet Offices
      - Atlanta
      - Los Angeles
      - New York
      - San Francisco
      - Washington
    - ⊞ Internet Practices

3. In the Home page template’s SiteSearch web service, specify the Site ID for the Internet Offices view you defined.

Note: *The SiteSearch web service returns parent and child views.*

INTERNET HOME PAGE WEB SERVICES

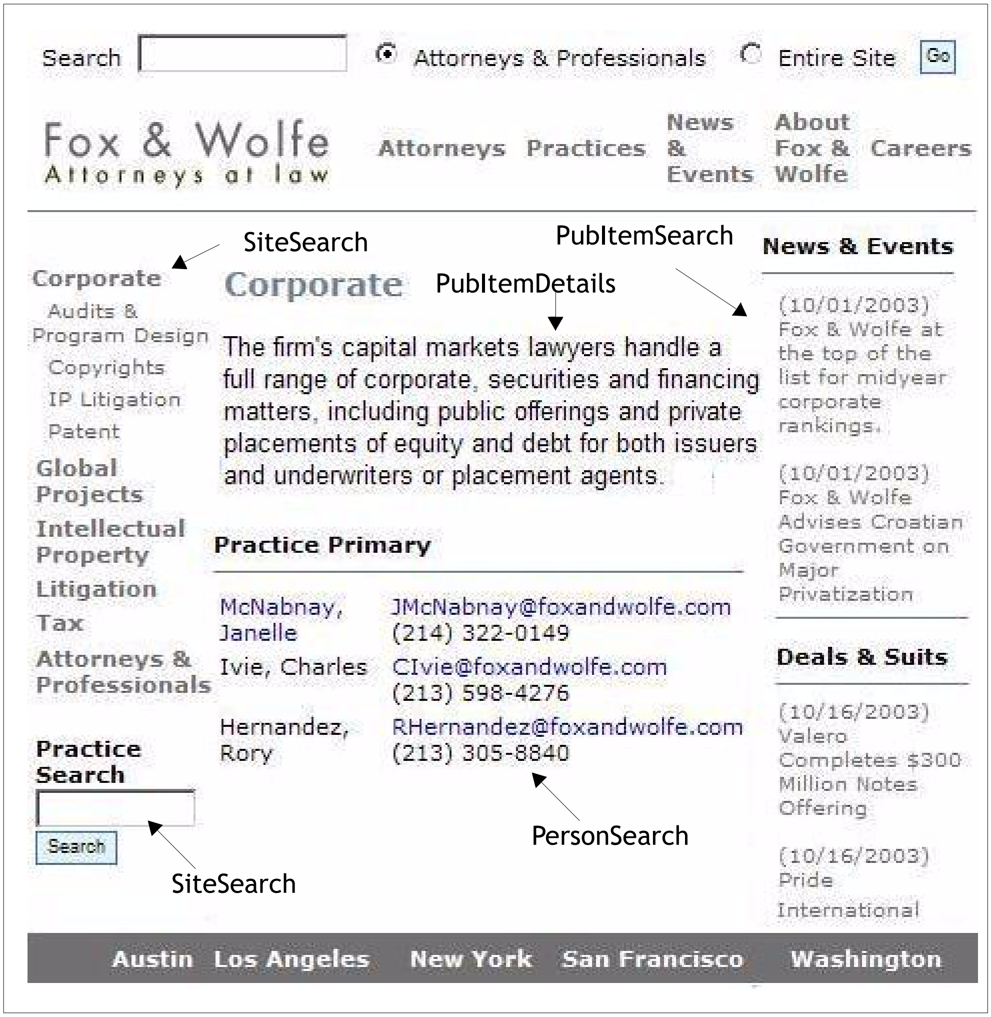
For additional information about the web services called by the Home page template, refer to the following sections:

- “NavBarDetails web service” on page 77.
- “PersonSearch web service” on page 79
- “PubItemDetails web service” on page 86
- “PubItemSearch web service” on page 81
- “SiteSearch web service” on page 87

Practice page template

Figure 4-10 below displays a rendering of the Practice page template; it allows you to spotlight your firm's practices and sub-practices. This template is set up to render a description of a practice, important practice contacts and a list of key published items.

Figure 4-10 Practice page template



Search ● Attorneys & Professionals ○ Entire Site Go

Fox & Wolfe
Attorneys at law

Attorneys Practices News & Events About Fox & Wolfe Careers

Corporate
Audits & Program Design
Copyrights
IP Litigation
Patent
Global Projects
Intellectual Property
Litigation
Tax
Attorneys & Professionals

Practice Search

Search

Corporate

The firm's capital markets lawyers handle a full range of corporate, securities and financing matters, including public offerings and private placements of equity and debt for both issuers and underwriters or placement agents.

Practice Primary

McNabnay, Janelle	JMcNabnay@foxandwolfe.com (214) 322-0149
Ivie, Charles	CIvie@foxandwolfe.com (213) 598-4276
Hernandez, Rory	RHernandez@foxandwolfe.com (213) 305-8840

News & Events

(10/01/2003) Fox & Wolfe at the top of the list for midyear corporate rankings.

(10/01/2003) Fox & Wolfe Advises Croatian Government on Major Privatization

Deals & Suits

(10/16/2003) Valero Completes $300 Million Notes Offering

(10/16/2003) Pride International

Austin Los Angeles New York San Francisco Washington

Setting up a Practice page

To set up your Practice page using the Practice page template, perform the following:

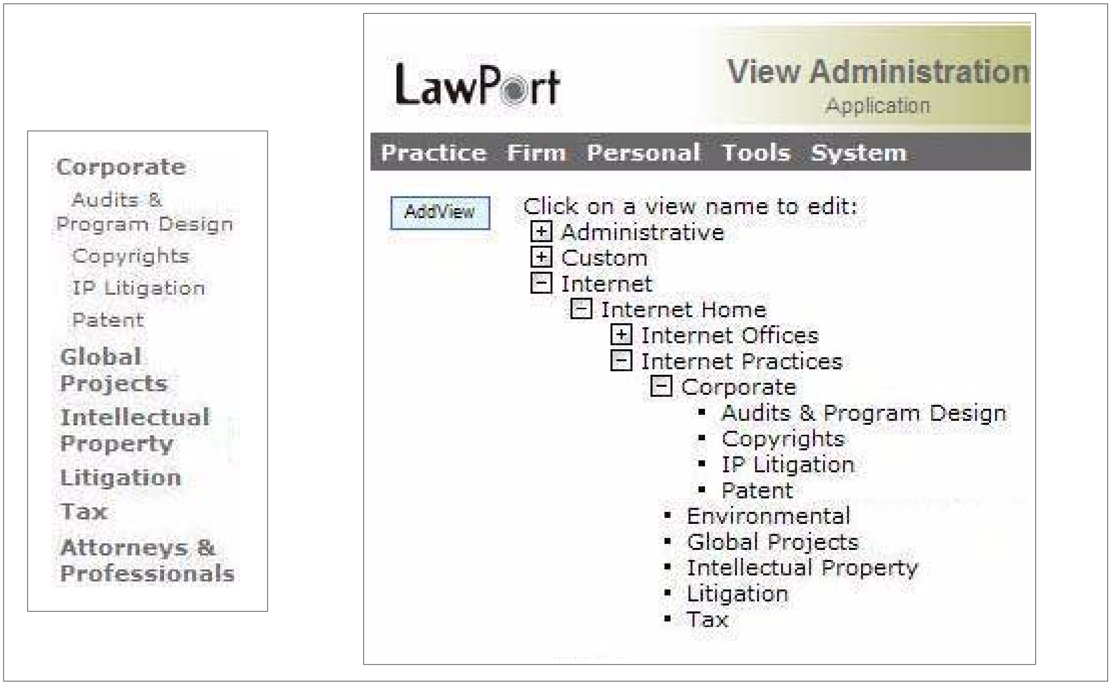
Displaying a Practice links bar ...

To render the Practice links bar, perform the following:

1. In your intranet’s view administration, define an Internet Practices view and make it a child of your Internet Home view. If you need help adding an Internet view, refer to Chapter 3 of the LawPort Administrator Guide.

2. Also, define a practice view for each of your practice links, making them a child of the Internet Practices view you defined in step 1 (and make your sub-practice views children of your practice views, as with the Copyrights view in Figure 4-11 on the right). Figure 4-11 displays the Practice links on the left and their corresponding Internet Practice views on the right.

Figure 4-11 Defining Internet Practices views in view administration



Corporate
- Audits & Program Design
- Copyrights
- IP Litigation
- Patent

Global Projects
Intellectual Property
Litigation
Tax
Attorneys & Professionals

Click on a view name to edit:

- ⊞ Administrative
- ⊞ Custom
- ⊟ Internet
  - ⊟ Internet Home
    - ⊞ Internet Offices
    - ⊟ Internet Practices
      - ⊟ Corporate
        - Audits & Program Design
        - Copyrights
        - IP Litigation
        - Patent
      - Environmental
      - Global Projects
      - Intellectual Property
      - Litigation
      - Tax

3. In the Practice page template's SiteSearch web service, specify the site ID of the Internet Practices view you defined.

Displaying a practice's description ...

To render a practice's description on a specific practice page (as with the Corporate description in Figure 4-10 above), perform the following:

1. Publish a PubItem containing a practice description to the corresponding Internet practice view. For example, a description of the Corporate practice should be published to the Corporate Internet view.

Populating the Practice search ...

To activate the Practice search, perform the following:

1. The Practice search will need practices to search for, so you will need to define Internet Practice views in your intranet.

Displaying lists of published items ...

To display lists of published items (appearing on the right in Figure 4-10) perform the following:

1. Publish the PubItems that you want to display to the appropriate Internet Practices views.

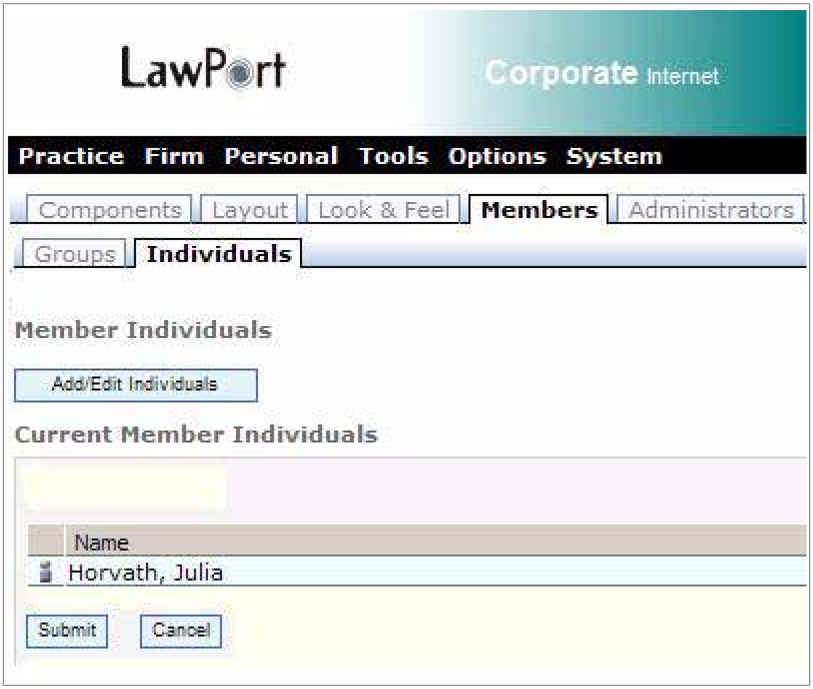
2. In the corresponding Practice page template's PubItemSearch web service, specify the PubItemType request parameter. In doing so, you can request PubItems of specific type (News, Research, Doc, etc.) to appear in your lists.

Displaying the practice primary section ...

To render the Practice Primary section, perform the following:

1. Add members to the appropriate Internet Practice view(s). Figure 4-15 shows the Corporate Internet view's Members screen with the member "Horvath, Julia" added. For more information on how to add members, refer to Chapter 1 of the LawPort Administrator Guide.

Figure 4-12 Adding members to the a Internet practices view



	Name
	Horvath, Julia

PRACTICE PAGE WEB SERVICES

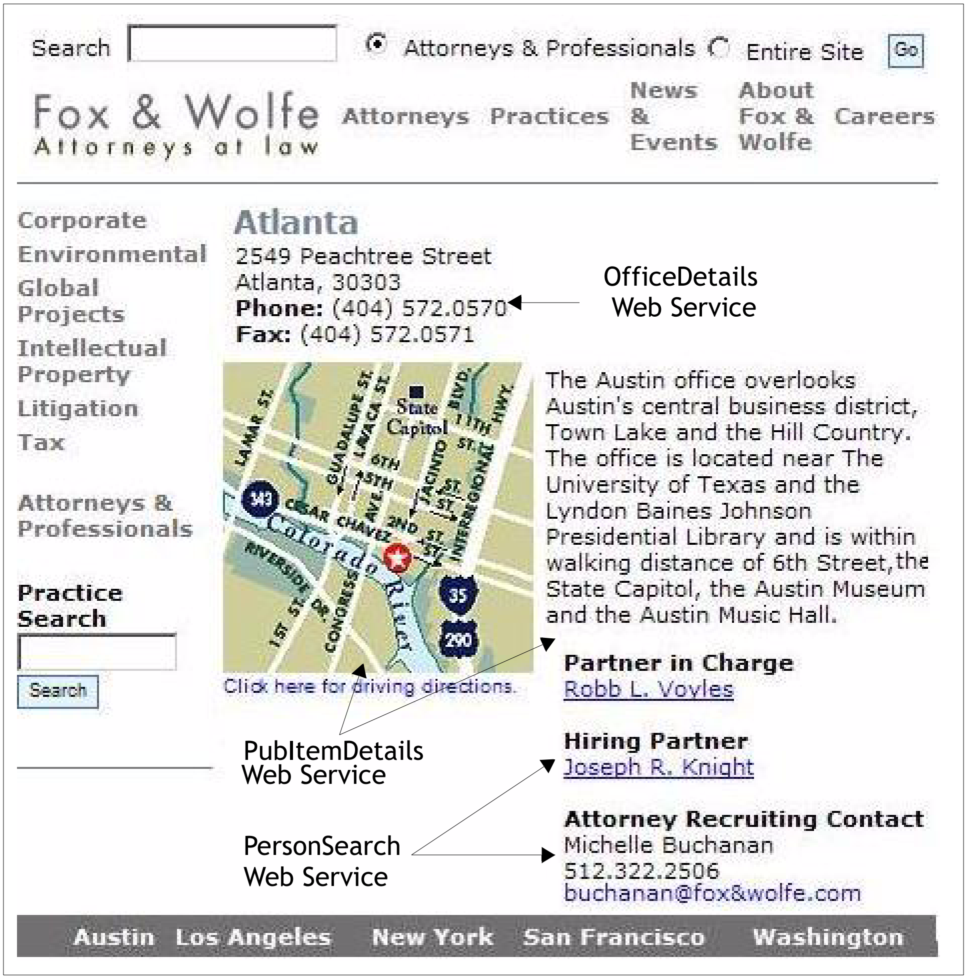
For additional information about the web services called by the Practice page template, refer to the following sections:

- "PersonSearch web service" on page 79
- "PubItemDetails web service" on page 86
- "PubItemSearch web service" on page 81
- "SiteSearch web service" on page 87

Office page template

You can display information about your firms individual offices using the Office page template, as with the Atlanta office in Figure 4-13 below. Refer to the following section to learn how to set up this page template.

Figure 4-13 Office Details page template



Search ● Attorneys & Professionals ○ Entire Site Go

Fox & Wolfe
Attorneys at law

Attorneys Practices News & Events About Fox & Wolfe Careers

Corporate
Environmental
Global Projects
Intellectual Property
Litigation
Tax

Attorneys & Professionals

Practice Search

Search

Atlanta

2549 Peachtree Street
Atlanta, 30303
Phone: (404) 572.0570
Fax: (404) 572.0571

Click here for driving directions.

The Austin office overlooks Austin's central business district, Town Lake and the Hill Country. The office is located near The University of Texas and the Lyndon Baines Johnson Presidential Library and is within walking distance of 6th Street, the State Capitol, the Austin Museum and the Austin Music Hall.

Partner in Charge
Robb L. Voyles

Hiring Partner
Joseph R. Knight

Attorney Recruiting Contact
Michelle Buchanan
512.322.2506
buchanan@fox&wolfe.com

Austin Los Angeles New York San Francisco Washington

Setting up the Office page

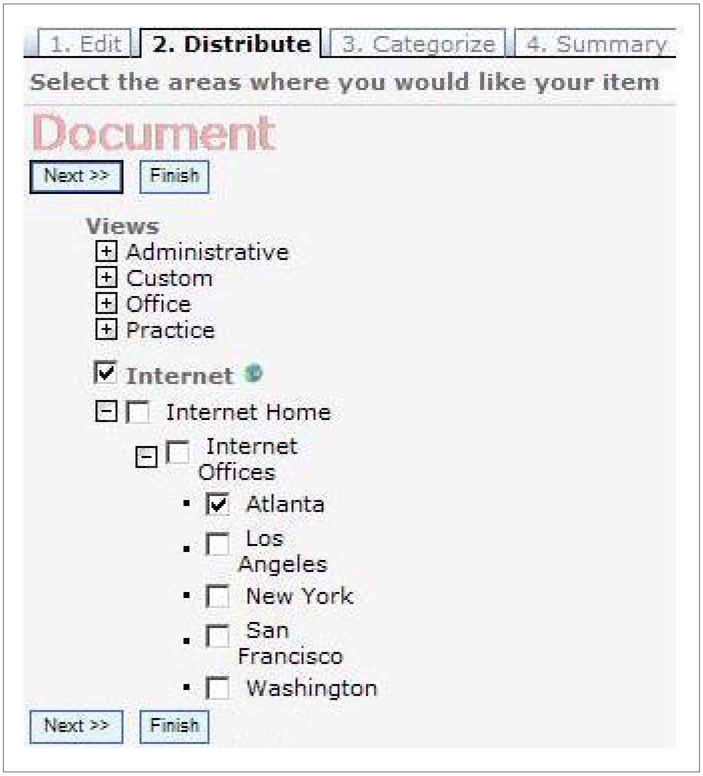
To set up your Office page using the Office Details template, perform the following:

Displaying an office's description ...

To display a specific office's description, perform the following:

1. Publish a PubItem containing a description to the Internet Office view whose description you want to display. Figure 4-14 shows the Publishing Wizard's Distribute step with the Internet Offices Atlanta view checked, specifying that the PubItem appear in the Internet Offices Atlanta view. For more details on how to publish content, refer to Chapter 4 of the LawPort User Guide.

Figure 4-14 Publishing office description to an Internet Office view



1. Edit | **2. Distribute** | 3. Categorize | 4. Summary

Select the areas where you would like your item

Document

Next >> Finish

Views
- ⊞ Administrative
- ⊞ Custom
- ⊞ Office
- ⊞ Practice

☑ Internet

⊟ □ Internet Home
- ⊟ □ Internet Offices
  - ☑ Atlanta
  - □ Los Angeles
  - □ New York
  - □ San Francisco
  - □ Washington

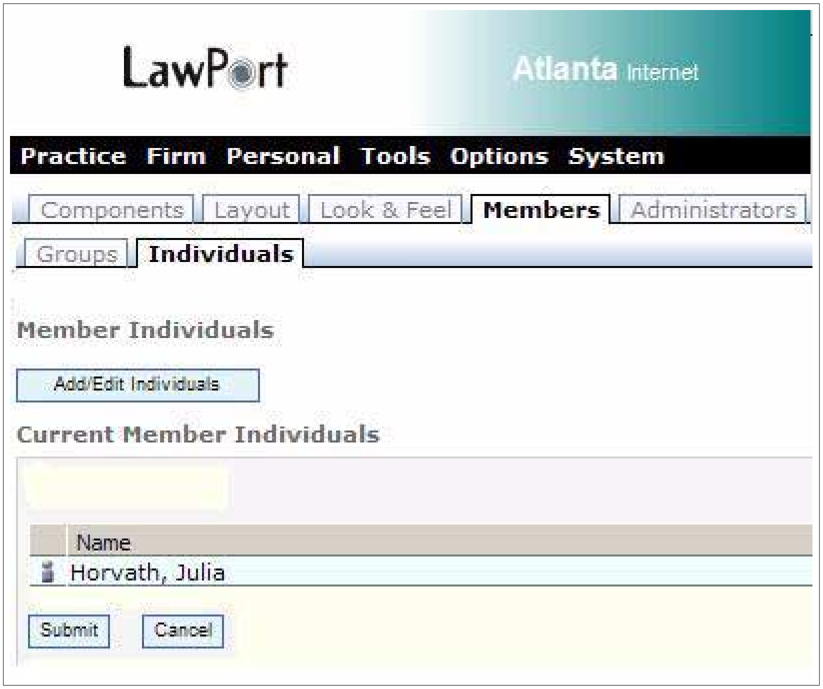
Next >> Finish

Displaying an office’s key contacts ...

To display a specific office’s key contacts, perform the following:

1. Add members whose contact information you want to display to the appropriate Internet Office view. Figure 4-15 shows the Atlanta Internet view’s Members screen with the member “Horvath, Julia” added. For more information on how to add members, refer to Chapter 1 of the LawPort Administrator Guide.

Figure 4-15 Adding members to an Internet Offices view



Name
Horvath, Julia

OFFICE PAGE WEB SERVICES

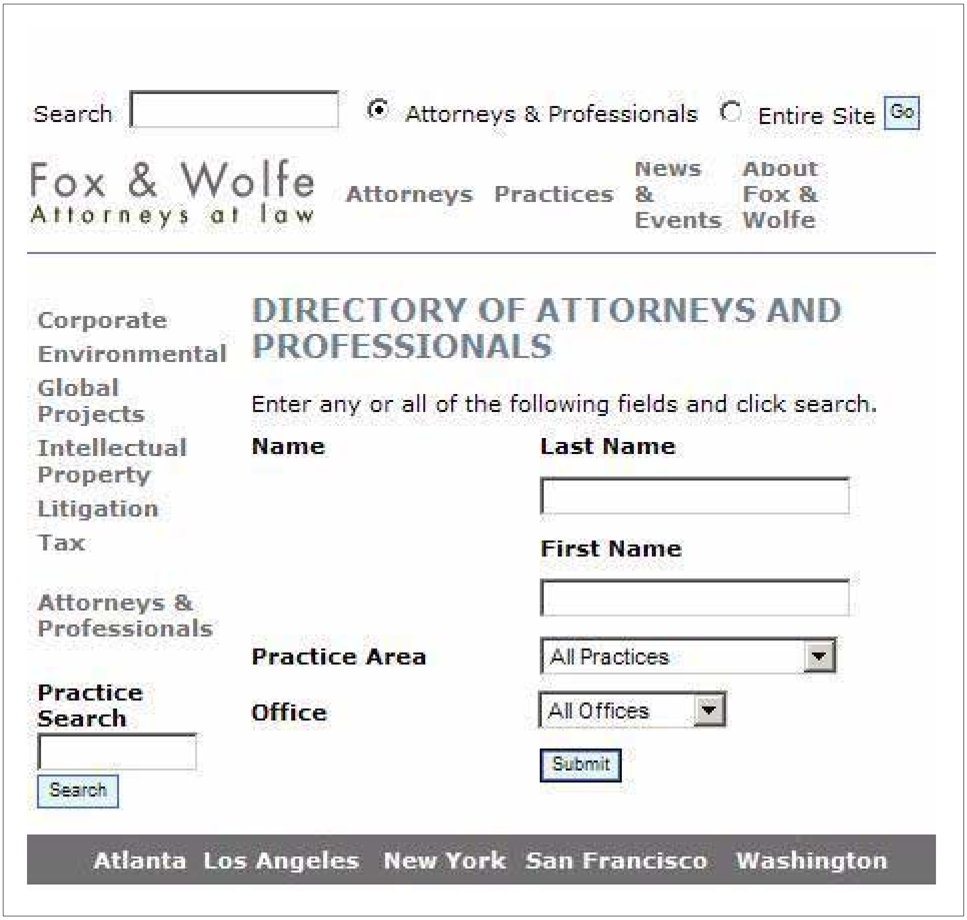
For additional information about the Office page web services, refer to the following sections:

- “SiteSearch web service” on page 87
- “OfficeDetails web service” on page 78

People Search page template

The People Search page template can be used to search for attorneys and professionals that have been associated with your Internet Practice and Internet Office views. Refer to the following section to learn how to set up this template.

Figure 4-16 People Search page template



Search ● Attorneys & Professionals ○ Entire Site

DIRECTORY OF ATTORNEYS AND PROFESSIONALS

Enter any or all of the following fields and click search.

Name

Last Name

First Name

Practice Area: All Practices

Office: All Offices

Setting up the People Search page template

To set up your People Search page, the following must be complete:

- Internet Practice views have been added to your intranet.
- Internet Office views have been added to your intranet.
- Members have been added to your Internet views.

People Search page web service

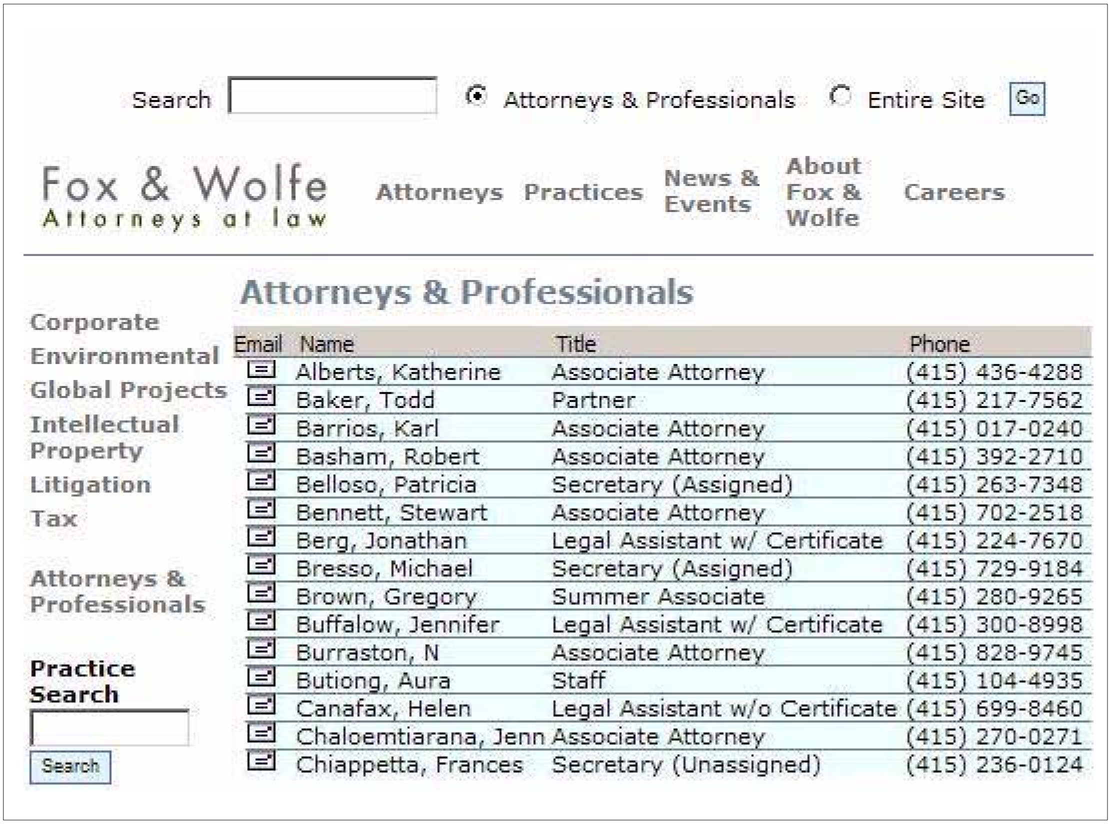
For additional information about the People Search page's web service, refer to the following sections:

- "PersonSearch web service" on page 79

People Search Results template

The People Search Results template corresponds with the People Search page template: it returns the matches from searches conducted in the People Search page. Refer to the next section to learn how to set up this template.

Figure 4-17 People Search Results template



Attorneys & Professionals

Email	Name	Title	Phone
✉	Alberts, Katherine	Associate Attorney	(415) 436-4288
✉	Baker, Todd	Partner	(415) 217-7562
✉	Barrios, Karl	Associate Attorney	(415) 017-0240
✉	Basham, Robert	Associate Attorney	(415) 392-2710
✉	Belloso, Patricia	Secretary (Assigned)	(415) 263-7348
✉	Bennett, Stewart	Associate Attorney	(415) 702-2518
✉	Berg, Jonathan	Legal Assistant w/ Certificate	(415) 224-7670
✉	Bresso, Michael	Secretary (Assigned)	(415) 729-9184
✉	Brown, Gregory	Summer Associate	(415) 280-9265
✉	Buffalow, Jennifer	Legal Assistant w/ Certificate	(415) 300-8998
✉	Burraston, N	Associate Attorney	(415) 828-9745
✉	Butiong, Aura	Staff	(415) 104-4935
✉	Canafax, Helen	Legal Assistant w/o Certificate	(415) 699-8460
✉	Chaloemtiarana, Jenn	Associate Attorney	(415) 270-0271
✉	Chiappetta, Frances	Secretary (Unassigned)	(415) 236-0124

Setting up the People Search Results template

To set up your People Search Results, the following must be complete:

- Internet Practice views have been added to your intranet.
- Internet Office views have been added to your intranet.
- Members have been added to your Internet views.

People Search Results web service

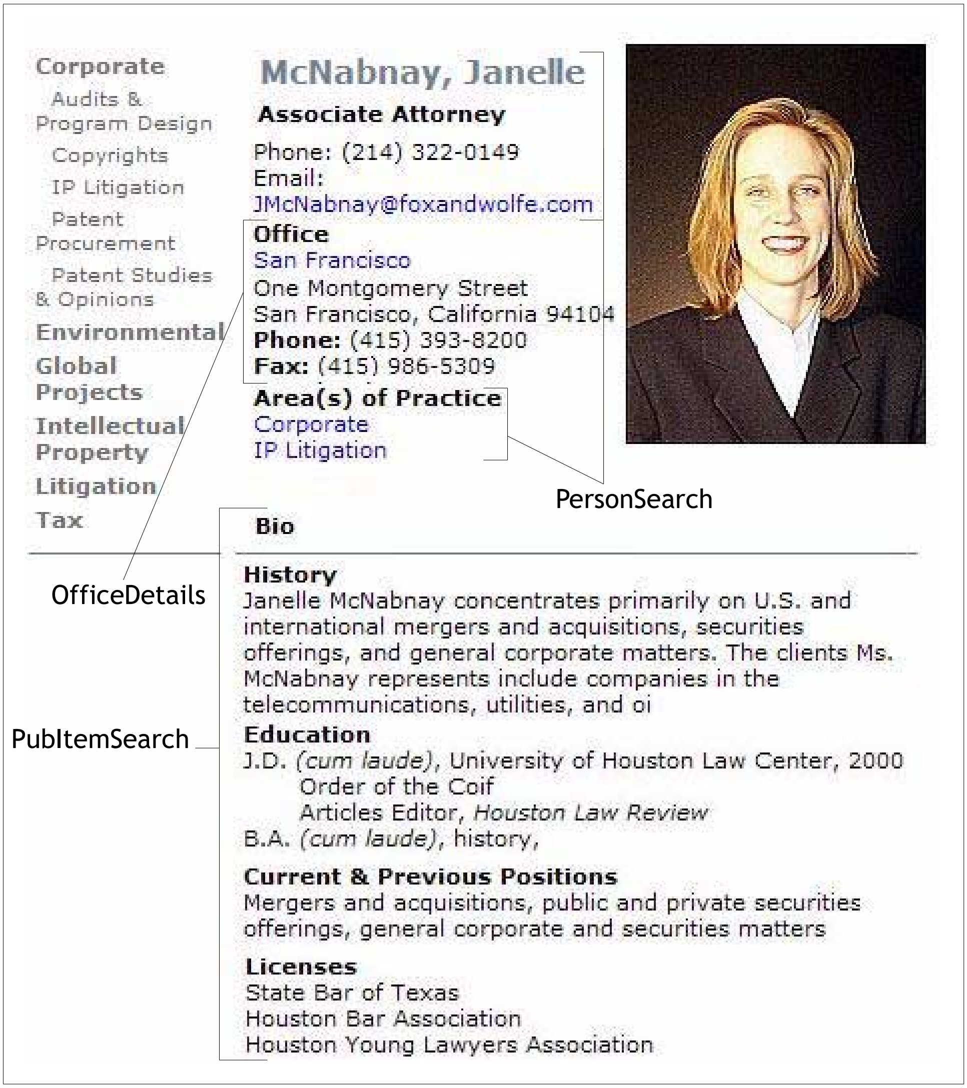
For additional information about the People Search Result's web service, refer to the following sections:

- "PersonSearch web service" on page 79

People Detail template

The People Detail template allows you to display an individual’s published biography (“Bio”). Refer to the next section to learn how to set up this template.

Figure 4-18 People Detail template



Corporate
Audits & Program Design
Copyrights
IP Litigation
Patent Procurement
Patent Studies & Opinions
Environmental
Global Projects
Intellectual Property
Litigation
Tax

McNabnay, Janelle

Associate Attorney

Phone: (214) 322-0149
Email:
JMcNabnay@foxandwolfe.com

Office
San Francisco
One Montgomery Street
San Francisco, California 94104
Phone: (415) 393-8200
Fax: (415) 986-5309

Area(s) of Practice
Corporate
IP Litigation

Bio

History
Janelle McNabnay concentrates primarily on U.S. and international mergers and acquisitions, securities offerings, and general corporate matters. The clients Ms. McNabnay represents include companies in the telecommunications, utilities, and oi

Education
J.D. *(cum laude)*, University of Houston Law Center, 2000
Order of the Coif
Articles Editor, *Houston Law Review*
B.A. *(cum laude)*, history,

Current & Previous Positions
Mergers and acquisitions, public and private securities offerings, general corporate and securities matters

Licenses
State Bar of Texas
Houston Bar Association
Houston Young Lawyers Association

Setting up the People Detail template

To set up your People Detail template, the following must be complete:

- Members who have published bios have been added to your Internet views.

People Detail web services

For additional information about the People Detail's web services, refer to the following sections:

- "PersonSearch web service" on page 79
- "PubItemDetails web service" on page 86
- "PubItemSearch web service" on page 81

Internet web services API

This section provides developers with general information about web services functionality and, in its last section, includes detailed information about each Internet web service.

This section includes the following:

Quick review of the web services standards

The following lists the standards to which the Web Services Internet discussed in this chapter adhere:

- **XML.** Extensible Markup Language is used to format the messages or requests passed between web services.
- **SOAP.** Simple Object Access Protocol is used to invoke the Internet web services' functionality.
- **HTTP.** Hyper Text Transfer Protocol is used to transmit the SOAP request across the web**.**
- **WSDL.** Web Services Description Language is used to describe the interfaces, request parameters and the return values that the Internet web services provide.

How do Internet web services work?

The Internet web services accessed from an Internet site you create pull their data from your firm's intranet application. The steps below briefly explain how data is exchanged when an Internet web service is called from an Internet site.

The Internet web services execute the following basic operations:

1. Internet site makes a request via SOAP to your firm's Internet web services, which are located on your intranet server.
2. The Internet web services then parse the XML contained in the SOAP message.
3. Once parsed, the web service executes functionality specified by the XML.
4. The web service extracts the results and then formats them in XML.
5. The reply containing the results is then transmitted via SOAP back to your Internet site.

Viewing WSDLs for LawPort web services

WSDL (Web Service Description Language) describes a web service. Many languages have tools to automatically generate client code from a WSDL file. To view a WSDL that is associated with a LawPort web service, perform the following:

1. Browse to: http://*yourfirmsname*/LawPort/Services/ and
2. Click on a LawPort web service in the list of web service that appears.

The WSDL file for the web service you selected will appear.

Internet web services

This section describes each web service's function, request parameters and return values.

This section includes the following:

"NavBarDetails web service" on page 77

"OfficeDetails web service" on page 78

"PersonSearch web service" on page 79

"PubItemSearch web service" on page 81

“PubItemFullTextSearch web service” on page 84

“PubItemDetails web service” on page 86

“SiteSearch web service” on page 87

NavBarDetails web service

Function: This service retrieves the navigation menu and links for a specific Internet view.

The sections listed below provide detailed information on the request parameters and the return values associated with the Navigation web service.

- “NavBarDetails request parameters” on page 77
- “NavBarDetails return values” on page 77

NavBarDetails request parameters

Table 4-19 lists and describes the request parameters associated with NavBarDetails.

Table 4-19 NavBarDetails request parameters

Request Parameter	Required/ Optional	Format	Parameter Description
ContainerID	Required	int	The ID of the navigation bar (NOT the component ID).

NavBarDetails return values

Table 4-20 lists return values that can be used in constructing a navigation bar as links.

Table 4-20 NavBarDetails request parameters

Return Values	Format	Parameter Description
ContainerID	String	A description of the link.
Description	int	The unique ID of the link.
ID	int	The unique Container ID of the parent container. Used to recreate multi-level menus.

Table 4-20 NavBarDetails request parameters

ParentContainerID	String	The text of the link.
Title	int	If this link also has a sub-menu, it must have a unique Container ID.
URL	String	A description of the link.

OfficeDetails web service

Function: This service retrieves office information, such as address, phone and fax numbers.

The sections listed below provide detailed information on the request parameters and the return values associated with the OfficeDetails web service.

- "OfficeDetails request parameters" on page 78
- "OfficeDetails return values" on page 79

OfficeDetails request parameters

Table 4-21 lists and describes the request parameters associated with OfficeDetails.

Table 4-21 OfficeDetails request parameters

Request Parameter	Required/ Optional	Format	Parameter Description
SiteID	Required	int	The ID of a site. The site must be associated with an office address.

OfficeDetails return values

Table 4-22 lists and describes the return values associated with OfficeDetails.

Table 4-22 OfficeDetails return values

Return Values	Format	Parameter Description
Address1	int	The office's street address.
Address2	int	The office's suite/floor.
Address3	int	Optional. (Firm can decide what address element is to be returned with this value.)
City	String	The city in which the office resides.
CountryName	String	The country in which the office resides.
Fax	int	The office's fax number. Includes number and extension.
Phone	int	The office's phone number. Includes number and extension.
postalCode	String	The office's postal code.
stateOrProvinceName	String	The office's state or province name.

PersonSearch web service

Function: This service can be called from an Internet site's People Search form to retrieve an Internet view's member list.

Sort Orders: First Name or Last Name are the available sort orders for this web service. LastName is the default sort order.

The sections listed below provide detailed information on the request parameters and the return values associated with the PersonSearch web service.

- "PersonSearch request parameters" on page 80
- "PersonSearch return values" on page 80

PersonSearch request parameters

Table 4-23 lists and describes the request parameters associated with the PersonSearch web service.

Table 4-23 PersonSearch request parameters

Request Parameter	Required/ Optional	Format	Parameter Description
ContactID	Required	int	The unique Contact ID of the person. Default of the Contact ID request parameter should be set to -1 if none is specified.
FirstName	Optional	String	Search for persons whose first names begin with this text.
LastName	Optional	String	Search for persons whose last names begin with this text.
SiteIdList	Optional	int Array	An array of Site Ids. Search for members of these sites.
MaxRecords	Optional	int	The maximum number of records to return.
SortOrder	Required	String	The sort order.
StartRecord	Optional	int	The first record to return.

PersonSearch return values

Table 4-24 lists and describes the return values associated with the PersonSearch web service.

Table 4-24 PersonSearch return values

Return Value	Format	Parameter Description
ContactID	int	The unique Contact ID of the person.
defaultEmail	Email object	The email of the person.
defaultPhone	Phone object	The phone number of the person. Includes number and extension.

Table 4-24 PersonSearch return values

Return Value	Format	Parameter Description
DisplayName	String	The default way to display the person's full name.
FirstName	String	The first name of the person.
InternetSiteIdList	int Array	An array of Internet Site ID's of which the person is a member.
JobTitle	String	The job title of the person.
LastName	String	The last name of the person.
Net ID	String	The unique Net ID of the person.

PubItemSearch web service

Function: This service can be called from an Internet site's content search form to retrieve PubItems that have been published to an Internet view. All the parameters indicated in first table below can be searchable, and the ViewID parameter can be used to retrieve content specific to an Internet view.

Sort Orders: The following list the sort orders available for the PubItem Search web service:

- Author/ModDate
- Author/Title
- ModDate/Title
- StartDate/Title
- Title/ModDate

The sections listed below provide detailed information on the request parameters and the return values associated with the PubItemSearch web service.

- "PubItemSearch request parameters" on page 82
- "PubItemSearch return values" on page 83

PubItemSearch request parameters

Table 4-25 lists and describes the request parameters associated with the PubItemSearch web service.

Table 4-25 PubItemSearch request parameters

Request Parameter	Required/ Optional	Format	Parameter Description
AuthorID	Optional	int	The Contact ID of the PubItem's author.
Description	Optional	String	Search for this text in the PubItem's description.
IncludeCustomFields	Optional	Boolean	Determines whether to return the custom fields in the result set, which can be slower.
MaxRecords	Optional	int	The maximum number of records to return.
PubItemType	Optional	String	The PubItem type (news, event, etc).
SiteIDList	Optional	int Array	The list of Internet Site ID's to which the PubItems were published.
SortOrder	Required	String	The sort order of the results.
StartRecord	Optional	int	The first record to return.
StartDate	Optional	Date	Search only for PubItems that are expected to be visible after this date.
Title	Optional	String	Search for this text in the PubItem's title.
primarySortOrder	Optional	String	The primary sort order of the results.
primarySortDirection	Optional	String	The primary sort direction of the results.
secondarySortOrder	Optional	String	The secondary sort order of the results.
secondarySortDirection	Optional	String	The secondary sort direction of the results.
includeExpired	Optional	Boolean	Determines if results will include expired PubItems. Set to "true" to include expired and set to "false" to not include expired.

PubItemSearch return values

Table 4-26 lists and describes the return values associated with the PubItemSearch web service.

Table 4-26 PubItem Search return values

Return Value	Format	Parameter Description
AuthorID	int	The author's unique Author ID.
CustomFields	int Array of Custom Field objects	The custom fields for the PubItem. CustomField returns values: Name (String) Value (String)
Description	String	The PubItem's description.
EndDate	Date	The date after which the PubItem will no longer be visible.
InternetSiteIdList	int Array	The list of Internet Site Ids to which the PubItem has been published.
ModificationDate	Date	The date the PubItem was last edited.
PubItemID	int	The unique PubItemID.
PubItemType	String	The type of PubItem (news, event, etc).
StartDate	Date	The date that the PubItem is first expected to be visible.
Title	text	The PubItem's title.

PubItemFullTextSearch web service

Function: This service is called from an Internet site's content search form, and only returns PubItems that have been published to an Internet view. Implemented with the Verity search functionality in LawPort, this web service call is a wrapper around the LawPort full-text search API.

The sections listed below provide detailed information on the request parameters and the return values associated with the PubItemFullTextSearch web service.

- "PubItemFullTextSearch web service" on page 84

- “PubItemFullTextSearch web service” on page 85

PubItemFullTextSearch request parameters

Table 4-27 lists and describes the request parameters associated with the PubItemFullTextSearch web service.

Table 4-27 PubItmeFullTextSearch request parameters

Request Parameter	Required/ Optional	Format	Parameter Description
DateLastModified-LowerLimit	Optional	Date	Search only for PubItems modified after this date.
DateLastModifiedUpperLimit	Optional	Date	Search only for PubItems modified before this date.
SortSpec	Required	String	The sort order of the results, either by PubItem type or by relevancy score. Valid values are (ITEM_TYPE, SCORE).
SearchStringType	Required	String	Determines whether the SearchString parameter should be interpreted as simple text or an advanced query using explicit Verity syntax. Valid values are (SIMPLE, ADVANCED).
SiteIdList	Optional	int Array	The list of Internet Site Ids to which the PubItem has been published.
SearchString	Required	String	The text to search for in the content of the PubItem.

PubItemFullTextSearch return values

Table 4-28 lists and describes the request parameters associated with the PubItemFullTextSearch web service.

Table 4-28 PubItemFullTextSearch return values

Return Value	Format	Parameter Description
CustomFields	int Array of Custom Field objects	The custom fields for the PubItem. CustomField returns values: Name (String) Value (String)
Description	String	The PubItem’s description.
EndDate	Date	The date after which the PubItem will no longer be visible.
InternetSiteIdList	int Array	The list of Internet Site Ids to which the PubItem has been published.
ModDate	Date	The date the PubItem was last edited.
PubItemID	int	The unique PubItemID.
PubItemType	String	The type of PubItem (news, event, etc).
StartDate	Date	The date that the PubItem is first expected to be visible.
Title	text	The PubItem’s title.

PubItemDetails web service

Function: This web service returns the content of a specific PubItem to the Internet site. Only content that has been published to a view of type Internet is returned.

The sections listed below provide detailed information on the request parameters and the return values associated with the PubItemDetails web service.

- “PubItemDetails request parameters” on page 86
- “PubItemDetails return values” on page 86

PubItemDetails request parameters

Table 4-29 lists and describes the request parameters associated with the PubItemDetails web service.

Table 4-29 PubItemDetails request parameters

Request Parameter	Required/ Optional	Format	Parameter Description
PubItemID	Required	integer	The unique Pubitem ID.

PubItemDetails return values

Table 4-30 lists and describes the return values associated with the PubItemDetails web service.

Table 4-30 PubItemDetails return values

Return Value	Format	Parameter Description
CustomFields	int Array of Custom Field objects	The custom fields for the PubItem.
Content	String	The content of the PubItem.
ContentFile	String	The path to the PubItem’s file, if applicable.
Description	String	The description of the PubItem.
EndDate	Date	The date after which the PubItem is no longer visible.
EditDate	Date	The date the PubItem was last modified.
PubItemID	int	The unique ID of the PubItem.
PubItemType	String	The type of PubItem (news, event, etc).
StartDate	Date	The date after which the PubItem should become visible.
SiteName	String	The name of the site.

Table 4-30 PubItemDetails return values

Return Value	Format	Parameter Description
Title	String	The title of the PubItem.
fileExtension	Boolean	Returns, when set to “true,” the file extension of the PubItem.

SiteSearch web service

Function: This service returns a list of Internet sites and all of its child (sub) sites. Only site of type Internet are returned.

The sections listed below provide detailed information on the request parameters and the return values associated with the SiteSearch web service.

- “SiteSearch request parameters” on page 87
- “SiteSearch return values” on page 88

SiteSearch request parameters

Table 4-31 lists and describes the request parameters associated with the SiteSearch request parameters.

Table 4-31 SiteSearch request parameters

Request Parameter	Required/ Optional	Format	Parameter Description
SiteID	Required	int	The ID of a site.

SiteSearch return values

Table 4-32 lists and describes the return values associated with the SiteSearch web service.

Table 4-32

Return Value	**Format**	**Parameter Description**
ParentSiteID	int	The unique Site ID of the site’s parent.
SiteID	int	The unique Site ID of the site.
SiteDescription	String	The description of the site.

5 Customizing LawPort: Developer Guidelines

Chapter Contents:

Overview of Customizing LawPort

This chapter describes the techniques that are supported by SV Technology to customize LawPort. These customizations will be automatically preserved during LawPort upgrades. Conversely, modifications to LawPort outside of the guidelines below cannot be expected to be preserved during upgrades.

Adding CSS styles

LawPort provides a stylesheet which is guaranteed not to be overwritten. The file is located at {lawport_root}/includes/CustomStyles.css. This stylesheet is included at the beginning of every view in LawPort, thus it is the place to add any custom styles. It can also be used to override standard LawPort styles to customize the appearance of LawPort views. However, while it is not likely, some of the dynamic HTML used in LawPort can be broken this way if you modify existing LawPort styles. For example, the "visibility" and "display" properties are typically used to handle the dynamic display of menus and collapsable tree structures. We recommend that you do not modify these style properties for existing elements.

LawPort Images

LawPort does not employ many of images, but there are a few that clients typically replace, including:

- /graphics/headerLogo.gif" is the graphic that is displayed in the upper-left corner of every intranet view.
- /graphics/extranetlogo.gif" is the graphic that is displayed in the upper-left corner of every extranet view.

Inserting custom HTML into a LawPort view

One of the most common customization techniques is to incorporate custom HTML to a view, such as a greeting or notice. LawPort supports this technique by providing a component that can display custom HTML. The component effectively is a custom UI block that can be placed anywhere on the view. The custom HTML can be entered directly into the component, or the component can take advantage of existing work and display the contents of an existing PubItem. For more details, refer to the Creating Components chapter in the LawPort Administrator Guide.

Incorporating custom applications via URL components

Custom web applications can be incorporated into LawPort. LawPort includes components that allow you to insert the HTML from external web sites into LawPort as components. To build a URL component, specify the component's URL and select the parameters to add to the URL. These parameters provide contextual information so your web applications can respond in a personalized manner. For example, your application could include information specific to the user or the client matter in LawPort. For more details, refer to the Creating Components chapter in the LawPort Administrator Guide.

JSP customizations

As a rule, changes to JSP files are not supported customizations. Nevertheless, most clients customize the JSPs in LawPort to some degree and take it upon themselves to maintain these customizations during software upgrades. The most commonly modified files are listed below:

- "/includes/header.jsp" is customized to modify the display that will be seen at the top of every view.

- "/includes/footer.jsp" is customized to modify the display that will be seen at the bottom of every view.

PubItem List formats

This is one of the few JSP customizations that is guaranteed to be preserved. SV supports the creation of new display formats for PubItem List components. This customization requires three steps:

1. Update lawport-config.xml with a new format name. These formats are located under the <pubitem-format-config> section, a sample of which is shown below. To add a new format, simply add a new line to this section. For example, the last line <myNewFormat>myNewFormat</myNewFormat> has been added below

```
<pubitem-format-config>
  <title>Title</title>
  <titleModificationDate>TitleModDate</titleModificationDate>
  <titleStartDate>TitleStartDate</titleStartDate>
  <titleEndDate>TitleEndDate</titleEndDate>
  <titleAuthor>TitleAuthor</titleAuthor>
  <myNewFormat>myNewFormat</myNewFormat>
</pubitem-format-config>
```

2. Modify the jsp page that contains the PubItem List formats. The file is {lawport_root}/jsp/component/pubitems/ListItem.jsp. The purpose of this page is to look for a format.

This code example below represents a sample PubItem format. The key point is that in the first line, the "value" attribute must be the same as the new line you entered into lawport-config.xml, in this example, value="myNewFormat".

```
<logic:equal name="displayFormat" value="myNewFormat">
  <tr><td>
    <a href="<%=sItemHREF%>"  style="font-size: larger;">
      <strong><bean:write name="pubItem" property="title" /></strong>
    </a>
    <%=sIcons%>
    <br/>
    <bean:write name="pubItem" property="description" />
    <br/>
  </td></tr>
</logic:equal>
```

3. The new format should become available in the PubItem List component administration screen. Both new components and existing components can be administered to use the new format.

Visual Treatment of Component Headers

The visual style of component headers can be altered by modifying CustomStyles.css stylesheet. The path to the stylesheet is: [LawPort_install_drive]\JRun4\servers\[server name]\lawport-ear\lawport-war\includes\CustomStyles.css. For general information on customizing LawPort styles, see the LawPort Style Sheet Guide.

To apply one of these themes, copy one of the samples below into CustomStyles.css and it will take precedence over the default style class in lawport.css. You are also able to customize the headers by changing the color schemes of any of the samples.

Sample themes

- WHITE - When applied on a white background, component headers will appear similar to the current look of component headers in LawPort.

- WISP - A very understated, subtle look. Fades from a pale gray on top to white on the bottom.
- BLUE FADE - Fades from a blue bottom to light blue on top. This is the style that you see in the screenshots of this document.
- SOLID BLUE - A solid blue header with rounded corners.

The default LawPort component header fades from a gray bottom to light gray on top. Except for the color, it has a similar look to the blue fade.

6 Customizing LawPort Style Classes

Chapter Contents

What you should know about customizations ...

There are a few points about including styles in the LawPort application, such as class precedence rules for cascading style sheets, that will be useful to know when you are customizing your own LawPort styles.

One of the precedence rules of cascading style sheets is that classes are prioritized by the order that they appear in a document. The class that appears later in the document takes precedence over those that appear first. Therefore, the last declaration class overwrites previous styles with the same class name.

LawPort uses linked style sheets that point to external.css. LawPort links to lawport.css first and follows with customStyles.css. As a result, due to the precedence rules of CSS, any style in customStyles.css with duplicate name of a class in lawport.css will overwrite the LawPort class.

When upgrading versions of LawPort, lawport.css is overwritten with the latest version, while customStyles.css is left as is.

For that reason, to make changes to existing LawPort styles you should create a duplicate class in customStyles.css to overwrite the LawPort class you wish to customize and set attributes accordingly. If you wish to create your own style classes for use in LawPort, these should also be created in customStyle.css. You should never edit or create classes directly in lawport.css as they will be overwritten.

Lawport.css and customStyles.css are both located in the *lawport-war/includes* directory.

For more information about style sheets, refer to “Adding LawPort Styles to your custom components” on page 8 of this guide.

Frequently Asked Questions

1. How do I place my own logo in the header?

The header logo is not controlled by lawport.css. To modify the path/filename of the image that appears in the header, edit it directly in lawport-war/includes/header.jsp. Search for headerLogo.gif.

2. How can I make my custom component look like a LawPort component?

When creating a custom URL component, use the style classes from lawport.css or customStyles.css in your component to make it look like a LawPort component. When creating a URL component that uses an iframe or a new browser window, you will need to add a link element to lawport.css and customStyles.css.

- <link rel="stylesheet "type="text/css" href="/includes/lawport.css">
- <link rel="stylesheet" type="text/css" href="/includes/CustomStyles.css">

Once these links are added to your custom component, you may access any of the style classes.

For more information on style sheets, refer to “Creating styles for your custom URL components” on page 8 of this guide.

3. How do I change header/navigation background colors?

Background and text colors for any view can be changed using the Look & Feel administration tool. You can access this area by choosing Options>Look & Feel from the top navigation menu.

4. How do I change navigation component highlight color?

After reading the "Customizing LawPort Style Classes" section of this document, you should create a class in customStyles.css named .containerOver. Set the background-color attribute of this class to the hex code of your choosing.

```
.containerOver{
        background-color: #33FF66;
}
```

5. How do I change fonts in LawPort?

The default fonts for LawPort are Verdana, Arial, Helvetica, sans-serif. To change the default font used throughout LawPort, you will need to override the font-family attribute of the body tag in LawPort by setting the body tag font-family attribute to the font of your choosing in customStyle.css.

```
body {
    font-family: Courier, Times New Roman;
}
```

6. How do I change LawPort styles or add my own styles?

See "Customizing LawPort Style Classes" on page 96 of this chapter for complete details on customizing and editing LawPort styles.

7. Where are lawport.css and customStyles.css located?

Both lawport.css and customStyles.css are located in the lawport-war/includes directory.

8. How do I change the appearance of links in LawPort?

Create your own a, a:active, a:hover, a:link and a:visited classes in customStyles.css and set attributes accordingly.

Customizing LawPort Style Classes

This section includes provides details for each of the following LawPort style classes:

Application Wide Styles

Class name: **a**
Class attributes:
text-decoration: none;
Description: controls style of all content between <a> tags.
Use in application: applies throughout entire application.
Sample: <a>content</a>

Class name: **a:hover**
Class attributes:
background-color: #CCCCFF;

Description: controls style of all content between <a> tags when hovered over by the mouse pointer.
Use in application: applies throughout entire application.
Sample: <a>content</a>

Class name: **body**
Class attributes:

```
	font-size: 9pt;
	font-family: Verdana, Arial, Helvetica, sans-serif;
	margin-left: 0px;
	margin-right: 0px;
	margin-bottom: 0px;
	margin-top: 0px;
```

Description: controls style of all content between <body> tags. These are basically the default settings for the entire document.
Use in application: applies to entire document
Sample: <body>content</body>

Class name: **button**
Class attributes:

```
	border-left: 1px solid #4169E1;
	border-top: 1px solid #4169E1;
	border-bottom: 1px solid #4169E1;
	border-right: 1px solid #4169E1;
	font-size: 7pt;
	color: #000000;
	background-color: #E4F3F8;
	font-family: sans-serif;
	cursor: hand;
```

Description: controls style of all content between <button> tags.
Use in application: applies to all button tags used throughout entire application.
Sample: <button>content</button>

Class name: **h1**
Class attributes:

font-size:24pt;
font-weight: bold;
color: #000033;
margin: 10px 0px 20px 0px;

Description: controls style of all content between <h1> tags.
Use in application: applies throughout entire application.
Sample: <h1>content</h1>

Class name: **h2**
Class attributes:

font-size:12pt;
font-weight: bold;
margin: 5px 0px 5px 0px;
color: #708090;

Description: controls style of all content between <h2> tags.
Use in application: applies throughout entire application.
Sample: <h2>content</h2>

Class name: **h3**
Class attributes:

font-size:10pt;
font-weight: bold;
Description: controls style of all content between <h3> tags.
Use in application: applies throughout entire application.
Sample: <h3>content</h3>

Class name: **h4**
Class attributes:

font-size: 8pt;
font-weight: 900;
color: #696969;
margin: 5px 0px 0px 0px;

Description: controls style of all content between <h4> tags.
Use in application: applies throughout entire application.
Sample: <h4>content</h4>

Class name: **hr**
Class attributes:
    height: 1px;
    color: #708090;
Description: controls style of output produced by a <hr> tag.
Use in application: applies throughout entire application.
Sample: <hr/>

Class name: **input**
Class attributes:
    font-size: 8pt;
Description: controls style of all content contained within an input control.
Use in application: applies to all input tags used throughout entire application.
Sample: <input/>

Class name: **p**
Class attributes:
    font-size: 8pt;
    margin:0in;
    margin-bottom:.0001pt;
Description: controls style of all content between <p> tags.
Use in application: applies throughout entire application.
Sample: <p>content</p>

Class name: **select**
Class attributes:
    font-size: 8pt;
    visibility: visible;
Description: controls style of all content contained within a select box input control.
Use in application: applies to all select tags used throughout entire application.
Sample: <select>
    .....
    </select>

Class name: **td**
Class attributes:
    font-size: 8pt;
Description: controls style of all content between <td> tags.
Use in application: applies to all <td> tags used throughout entire application.
Sample: <td>content</td>

Class name: **textarea**
Class attributes:
    font-size: 8pt;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    border: 1px inset;
    margin: 1px;
Description: controls style of all content contained within a textarea input control.
Use in application: applies to all textarea tags used throughout entire application.
Sample: <textarea></textarea>

General Style Classes

Class name: **.arrow**
Class attributes:
    font-family: Webdings;
    font-size: 10pt;
Description: controls style of all content between tags with the class name arrow.
Use in application: used in top navigation to display arrow symbol when navigation item has sub menu.
Sample: <div class="arrow">content</div>

Class name: **.button**
Class attributes:
    border-left: 1px solid #4169E1;
    border-top: 1px solid #4169E1;
    border-bottom: 1px solid #4169E1;
    border-right: 1px solid #4169E1;

font-size: 7pt;
color: #000000;
background-color: #E4F3F8;
font-family: sans-serif;
cursor: hand;

Description: controls style of all content between tags with the class name button.
Use in application: applied to button tags & input tags of type submit, button & reset throughout application.
Sample: <input class="button">content</input>

Class name: **.buttonClear**
Class attributes:

margin: 1px;
font-size: 7pt; color: black;
font-family: sans-serif;
cursor: hand;

Description: controls style of all content between tags with the class name buttonClear.
Use in application: used in the publishing tool on the pub item list type page when creating or editing.
Sample: <div class="buttonClear">content</div>

Class name: **.buttonHightlight**
Class attributes:

border-left: 1px solid #4169E1;
border-right: 1px solid #4169E1;
border-top: 1px solid #4169E1;
border-bottom: 1px solid #4169E1;
color: #000000;
background-color: #BDE0F2;

Description: controls style of all content between tags with the class name buttonHighlight.
Use in application: applied dynamically to input tags of type submit, button & reset when they are hovered over.
Sample: <div class="buttonHighlight">content</div>

Class name: **.button:hover**
Class attributes:
background-color: #BDE0F2;
Description: controls style of all content between tags with the class name of button when hovered over by the mouse pointer.
Use in application: applied to button tags & input tags of type submit, button & reset throughout application.
Sample: <div class="button">content</div>

Class name: .colorSelectContainer
Class attributes:
background-color: #FFFFFF;
display: ;
border: 1px solid silver;
position: absolute;
left: 20px;
top: 20px;
width: 230px;
height: 120px;
z-index: 100;
filter:progid:DXImageTransform.Microsoft.Shadow(color="#777777", Direction=135, Strength=3)alpha(Opacity=100);
-moz-opacity:0.9;
Description: controls style of all content between tags with the class name colorSelectContainer.
Use in application: applies style to color chooser pop-up in Look & Feel.
Sample: <div class="colorSelectContainer">content</div>

Class name: **.columnContent**
Class attributes:
background-color: #F5F5F5;
Description: controls style of all content between tags with the class name columnContent.
Use in application: applies style to client/matter details page label column and proposal status.
Sample: <div class="columnContent">content</div>

Class name: **.columnTitle**
Class attributes:
background-color: #CCCCFF;
Description: controls style of all content between tags with the class name columnTitle.
Use in application: applies style to window title bar for color chooser and calendar pop-ups. applies style to client/matter details page label column and proposal status.
Sample: <div class="columnTitle">content</div>

Class name: **.directoryLine**
Class attributes:
background-color: #999999;
Description: controls style of all content between tags with the class name directoryLine.
Use in application: applies style to divider between groups of data on Contact Detail page.
Sample: <div class="directoryLine">content</div>

Class name: **.divIndent**
Class attributes:
padding-left: 20px;
Description: controls style of all content between tags with the class name divIndent.
Use in application: used application wide to indent lines and blocks of text.
Sample: <div class="divIndent">content</div>

Class name: **.footer**
Class attributes:
font-size: 8pt;
padding: 1px;
width: 100%;
padding-top: 30px;
padding-left: 10px;
margin-top: 100px;
Description: controls style of all content between tags with the class name footer.
Use in application: applies style to the SV Technology copyright footer.
Sample: <div class="footer">content</div>

Class name: **.headerDefault**
Class attributes:
    background-color: SlateGray;
    color: white;
    font-weight: bold;
    border: none;
    font-size: 7pt;
Description: controls style of all content between tags with the class name headerDefault.
Use in application: applies style to system messages (ex. - error messages, successful operation).
Sample: <div class="headerDefault">content</div>

Class name: **.image a:hover**
Class attributes:
    background-color: white;
Description: controls style of all content between <a> tags when hovered over by the mouse pointer when embedded within tags with the class name of image.
Use in application: prevents application wide style for highlighting a link on mouseover (set above) from applying to images that are links.
Sample: <td class="image"><a>content</a></td>

Class name: **.imageOpacity**
Class attributes:
    filter: alpha(opacity=60);
Description: controls style of all content between tags with the class name imageOpacity.
Use in application: applied to pub item icons (ex. - secured, extranet. internet).
Sample: <div class="imageOpacity">content</div>

Class name: **.layoutSelects**
Class attributes:
    width: 175;
Description: controls style of all content between tags with the class name layoutSelects.
Use in application: applies style to multi-select boxes on layout options page.
Sample: <div class="layoutSelects">content</div>

Class name: **.lookFeelSample**
Class attributes:
border: solid;
border-width: 1px 1px 1px 1px;
border-color: #D3D3D3 #D3D3D3 #D3D3D3 #D3D3D3;
Description: controls style of all content between tags with the class name lookFeelSample.
Use in application: applies style to the sample page on the look & feel options page.
Sample: <div class="lookFeelSample">content</div>

Class name: **.subForm**
Class attributes:
border: solid;
border-width: 1px 1px 1px 1px;
border-color: #D3D3D3 #D3D3D3 #D3D3D3 #D3D3D3;
background: #F5F5F5;
padding: 5px 5px 5px 5px;
margin: 5px 0px 5px 0px;
Description: controls style of all content between tags with the class name subForm.
Use in application: used widely throughout application to box in and separate pertinent areas such as forms and search results.
Sample: <div class="subForm">content</div>

Class name: **.webdings**
Class attributes:
font-family: webdings;
font-size: 12pt;
text-align: center;
Description: controls style of all content between tags with the class name webdings.
Use in application: used to display webdings symbols throughout application.
Sample: <div class="webdings">content</div>

Component Styles

Class name: **.componentBigHeaderName**
Class attributes:

```
padding: 1px;
-moz-opacity: 0.9;
width: 100%;
padding-left: 0px;
margin-bottom: 0px;
font-size: 10pt;
font-weight: bold;
```

Description: controls style of all content between tags with the class name componentBigHeaderName.
Use in application: applies style to component name at top of components when on a single view of that component.
Sample: <div class="componentBigHeaderName">content</div>

Class name: **.componentContainer**
Class attributes:

```
padding-left:.5em;
```

Description: controls style of all content between tags with the class name componentContainer.
Use in application: style wrapper that adds spacing around containers.
Sample: <div class="componentContainer">content</div>

Class name: **.componentContainerSeparator**
Class attributes:

```
height: 1px;
width: 80%;
padding-top: 10px;
margin-top: 10px;
```

Description: controls style of all content between tags with the class name componentContainerSeperator.
Use in application: applies style to the component separator.
Sample: <div class="componentContainerSeperator">content</div>

Class name: **.componentContainerSeparatorLeft**
Class attributes:

filter:progid:DXImageTransform.Microsoft.Gradient(GradientType=1, StartColorStr=white, EndColorStr=gray);

Description: controls style of all content between tags with the class name componentContainerSeperatorLeft.
Use in application: applies style to the component separator.
Sample: <div class="componentContainerSeperatorLeft">content</div>

Class name: **.componentContainerSeparatorRight**
Class attributes:

filter:progid:DXImageTransform.Microsoft.Gradient(GradientType=1, StartColorStr=gray, EndColorStr=white);

Description: controls style of all content between tags with the class name componentContainerSeperatorRight.
Use in application: applies style to the component separator.
Sample: <div class="componentContainerSeperatorRight">content</div>

Class name: **.componentFull**
Class attributes:
Description: controls style of all content between tags with the class name componentFull.
Use in application: applied to component when display height is set to full.
Sample: <div class="componentFull">content</div>

Class name: **.componentFixed**
Class attributes:

height:300;

Description: controls style of all content between tags with the class name componentFixed.
Use in application: applied to component when display height is set to fixed.
Sample: <div class="componentFixed">content</div>

Class name: **.componentFixedScroll**
Class attributes:

height: 300;
overflow: scroll;

Description: controls style of all content between tags with the class name componentFixedScroll.
Use in application: applied to component when display height is set to scroll.
Sample: <div class="componentFixedScroll">content</div>

Class name: **.componentHeader**

Class attributes:
padding:1px;
-moz-opacity: 0.9;
width: 100%;
padding-left: 3px;
margin-bottom: 0px;
font-size: 9pt;
font-weight: bold;

Description: controls style of all content between tags with the class name componentHeader.
Use in application: applies style to text at top of components.
Sample: <div class="componentHeader">content</div>

Class name: **.componentHeaderName**
Class attributes:
padding:1px;
-moz-opacity: 0.9;
width: 100%;
padding-left: 0px;
margin-bottom: 0px;
font-size:10pt;
font-weight:bold;

Description: controls style of all content between tags with the class name componentHeaderName.
Use in application: applies style to component name at top of components.
Sample: <div class="componentHeaderName">content</div>

Class name: **.componentHeaderUnderline**
Class attributes:

filter:progid:DXImageTransform.Microsoft.Gradient(GradientType=1, StartColorStr=color: #191970, EndColorStr=#FFFACD);
height: 2px;

Description: controls style of all content between tags with the class name containerHeaderUnderline.
Use in application: applies style to divider between pub item header title, pub item details and content.
Sample: <div class="componentHeaderUnderline">content</div>

Class name: **.componentItemHeader**
Class attributes:

font-size: 7pt;

Description: controls style of all content between tags with the class name componentItemHeader.
Use in application: applies style to parent categories in category and pub items lists and to nodes in category and pub items trees.
Sample: <span class="componentItemHeader">content</span>

Class name: **.componentSmallHeader**
Class attributes:

font-size: 8pt;

Description: controls style of all content between tags with the class name componentSmallHeader.
Use in application: applies style to "new" links in Outlook and Notes mail components.
Sample: <a class="componentSmallHeader">content</a>

Container Styles

Class name: **.container**
Class attributes:
    position: relative;
    width: 140px;
Description: controls style of all content between tags with the class name container.
Use in application: applies style to navigation containers.
Sample: <div class="container">content</div>

Class name: **.container table**
Class attributes:
    width: 140px;
Description: controls style of all content between <table> tags when embedded within tags with the class name of container.
Use in application: applies style to navigation containers.
Sample: <div class="container"><table>content</table></div>

Class name: **.containerHeader table**
Class attributes:
    cursor: hand;
    color: #333333;
Description: controls style of all content between <table> tags when embedded within tags with the class name of containerHeader.
Use in application: sets styles to the inner <table> tags of the main navigation container.
Sample: <div class="containerHeader"><table>content</table></div>

Class name: **.containerHeader table td**
Class attributes:
    padding: 2px;
    border-bottom: 1px solid Silver;
Description: controls style of all content between <td> tags when embedded within tags with the class name of containerHeader and <table> tags, in that order.
Use in application: sets styles to the individual menu items of the main navigation container.
Sample: <div class="containerHeader"><table><td>content</td></table></div>

Class name: **.containerOutline**
Class attributes:
font-size: 1pt;
z-index: -1000;
width: 1px;
height: 1px;
Description: controls style of all content between tags with the class name containerOutline.
Use in application: applies style to the wrapper around a container.
Sample: <td class="containerOutline">content</td>

Class name: **.containerOver**
Class attributes:
background-color: blue;
Description: controls style of all content between tags with the class name containerOver.
Use in application: applied dynamically when mouse pointer hover over a container item.
Sample: <div class="containerOver">content</div>

Class name: **.containerSubmenu**
Class attributes:
width: 140px;
position: absolute;
background-color: #778899;
border: 1px solid black;
filter:progid:DXImageTransform.Microsoft.Shadow(color="#777777", Direction=135, Strength=3)alpha(Opacity=100);
-moz-opacity:0.9;
Description: controls style of all content between tags with the class name containerSubmenu.
Use in application: not currently used in application.
Sample: <div class="containerSubmenu">content</div>

Class name: **.containerSubmenu table**
Class attributes:
cursor: hand;
color: #333333;
Description: controls style of all content between <table> tags when embedded within tags with the class name of containerSubmenu.
Use in application: not currently used in application.
Sample: <div class="containerSubmenu"><table>content</table></div>

Class name: **.containerSubmenu table td**
Class attributes:
padding: 2px;
border-bottom: 1px solid Silver;
Description: controls style of all content between <td> tags when embedded within tags with the class name of containerSubmenu and <table> tags, in that order.
Use in application: not currently used in application.
Sample: <div class="containerSubmenu"><table><td>content</td></table></div>

DMS Folder Picker and Published Folder Explorer Styles

Class name: **.DMS_Component**
Class attributes:
background-color: #F5F5F5;
display: ;
border: 1px solid silver;
Description: controls style of all content between tags with the class name DMS_Component.
Use in application: applies style to DMS published folder explorer.
Sample: <table class="DMS_Component">content</table>

Class name: **.DMS_Files**
Class attributes:
background-color: #F0FFFF;
overflow-x: auto;
overflow-y: auto;

width: 500px;
height: 300px;

Description: controls style of all content between tags with the class name DMS_Files.
Use in application: applies style to DMS published folder explorer.
Sample: <div class="DMS_Files">content</div>

Class name: **.DMS_Folders**
Class attributes:

background-color: #FFFFFF;
overflow-x: auto;
overflow-y: auto;
border-right: 1px solid Silver;
width: 200px;

Description: controls style of all content between tags with the class name DMS_Folders.
Use in application: applies style to DMS published folder explorer.
Sample: <div class="DMS_Folders">content</div>

Class name: **.dropShadow**
Class attributes:

filter:progid:DXImageTransform.Microsoft.Shadow(color="#777777", Direction=135, Strength=3);

Description: controls style of all content between tags with the class name dropShadow.
Use in application: applies style to submit buttons for DMS check out/in unlock and navigation component admin.
Sample: <div class="dropShadow">content</div>

Class name: **.folderHighlight**
Class attributes:

background-color: Highlight;
color: #FFFFFF;

Description: controls style of all content between tags with the class name folderHighlight.
Use in application: applied dynamically when clicking on a folder in DMS published folder explorer.
Sample: <div class="folderHighlight">content</div>

Class name: **.folderSelect**

Class attributes:

```
background-color: #F5F5F5;
display: ;
border: 1px solid silver;
position: absolute;
left: 20px;
top: 20px;
width: 150px;
height: 160px;
z-index: 100;
filter:progid:DXImageTransform.Microsoft.Shadow(color="#777777", Direction=135,
Strength=3)alpha(Opacity=100);
-moz-opacity:0.9;
```

Description: controls style of all content between tags with the class name folderSelect.

Use in application: applies style to DMS folder picker.

Sample: <div class="folderSelect">content</div>

Class name: **.treeLineHorizon**

Class attributes:

```
background-image: url(../graphics/treeLineHorizon.gif);
```

Description: controls style of all content between tags with the class name treeLineHorizon.

Use in application: applies folder hierarchy images to DMS folders.

Sample: <div class="treeLineHorizon">content</div>

Class name: **.treeLineNoSub**

Class attributes:

```
background-image: url(../graphics/treeLineNoSub.gif);
```

Description: controls style of all content between tags with the class name treeLineNoSub.

Use in application: applies folder hierarchy images to DMS folders.

Sample: <div class="treeLineNoSub">content</div>

Class name: **.treeLineNoSubEnd**

Class attributes:

```
background-image: url(../graphics/treeLineNoSubEnd.gif);
```

Description: controls style of all content between tags with the class name treeLineNoSubEnd.
Use in application: applies folder hierarchy images to DMS folders.
Sample: <div class="treeLineNoSubEnd">content</div>

Class name: **.treeLineVert**
Class attributes:

    background-image: url(/graphics/treeLineVert.gif);

Description: controls style of all content between tags with the class name treeLineVert.
Use in application: applies folder hierarchy images to DMS folders.
Sample: <div class="treeLineVert">content</div>

Form Styles

Class name: **.formFieldHighlight**
Class attributes:

    background-color: #FFFFCC;
    filter:progid:DXImageTransform.Microsoft.Gradient(GradientType=1, StartColor-
    Str=#FFFFF0, EndColorStr=White);

Description: controls style of all content between tags with the class name formFieldHighlight.
Use in application: applied dynamically through javascript when an input field receives focus. currently disabled.
Sample: <div class="formFieldHighlight">content</div>

Class name: **input.buttonOver, button.buttonOver**
Class attributes:

    background-color: #BDE0F2;

Description: controls style of all content contained within an input control where the class name of the <input> tag is buttonOver. controls style of all content between <button> tags where the class name of the tag is buttonOver.
Use in application: applied dynamically to input tags of type submit, button & reset when they are hovered over.
Sample: <input class="buttonOver"/>

<button class="buttonOver">content</button>

Class name: **input.FormField**
Class attributes:
 border: 1px inset;
 margin: 1px;
Description: controls style of all content contained within an input control where the class name of the <input> tag is FormField.
Use in application: applied to input tags throughout application.
Sample: <input class="FormField"/>

Class name: **input.text**
Class attributes:
 border: 1px inset;
 margin: 1px;
Description: controls style of all content contained within an input control where the class name of the <input> tag is text.
Use in application: Used for input boxes in header quick search.
Sample: <input class="text"/>

Helptip Styles

(located in /lawport-war/includes/helptip.css)

Class name: **a.helpLink**
Class attributes:
 color: Green;
 text-decoration: none;
 border-bottom: 1px dashed Green;
 cursor: help;
Description: controls style of all content between <a> tags with the class name of helpLink.
Use in application: not currently used in application.
Sample: <a class="helpLink">content</a>

Class name: **a.helpLink:hover**
Class attributes:
  color: Red;
  text-decoration: none;
  border-bottom: 1px dashed Red;
Description: controls style of all content between <a> tags with the class name of helpLink when hovered over by the mouse pointer.
Use in application: not currently used in application.
Sample: <a class="helpLink">content</a>

Class name: **.help-tooltip**
Class attributes:
  position: absolute;
  width: 300;
  border: 1px Solid WindowFrame;
  background: Infobackground;
  color: InfoText;
  font: StatusBar;
  font: Status-Bar;
  padding: 3px;
  filter: progid:DXImageTransform.Microsoft.Shadow(color="#777777", Direction=135, Strength=3)Alpha(Opacity=85);
  z-index: 10000;
Description: controls style of all content between tags with the class name of help-tooltip.
Use in application: dynamically applied to plain .htm help files when help icon clicked.
Sample: <div class="help-tooltip">content</div>
Class name: **.help-tooltip a, .help-tooltip a:hover**
Class attributes:
  color: blue !important;
  background: none;
Description: controls style of all content between <a> tags when embedded between tags with the class name of help-tooltip.
Use in application: applies to links within help files.
Sample: <div class="help-tooltip"><a>content</a></div>

List and Tree Styles

Class name: **div#assocCatTree span**
Class attributes:
font-style: italic;
Description: controls style of all content between <span> tags nested within <div> tags with the id assocCatTree.
Use in application: adds style to category item of associated category list.
Sample: <div id="assocCatTree"><span>content</span></div>

Class name: **div#assocCatTree td**
Class attributes:
padding-left:15px;
vertical-align: top;
Description: controls style of all content between <td> tags nested within <div> tags with the id assocCatTree.
Use in application: adds spacing to items in associated category list.
Sample: <div id="assocCatTree"><td>content</td></div>

Class name: **div#assocCatTree_Title**
Class attributes:
font-weight: bold;
Description: controls style of all content between <div> tags with the id assocCatTree_Title.
Use in application: adds style to top node of associated category list.
Sample: <div id="assocCatTree_Title">content</div>
Class name: .elHighlight
Class attributes:
background-color: #CCCCFF;
Description: controls style of all content between tags with the class name elHighlight.
Use in application: highlights clicked element dynamically in associate category administration.
Sample: <div class="elHighlight">content</div>

Class name: **img.assocCatTree**
Class attributes:

padding-right: 5px;
margin-right: 5px;
cursor: pointer;

Description: controls style of <img> tags with the class name assocCatTree.
Use in application: adds style to expand/collapse image in associated category list.
Sample: <img class="assocCatTree"/>

Class name: **.itemDate**
Class attributes:

padding-left: 5px;
color: #666666;
font-size: smaller;
white-space: nowrap;

Description: controls style of all content between tags with the class name itemDate.
Use in application: applies style to pub item details (ex. - author, edit date).
Sample: <div class="itemDate">content</div>

Class name: **.itemDetail**
Class attributes:

padding-left: 5px;
width: 80%;

Description: controls style of all content between tags with the class name itemDetail.
Use in application: applies style to pub item details (ex. - author, edit date).
Sample: <div class="itemDetail">content</div>

Class name: **.simpleTreeCategory**
Class attributes:

font-style : italic;

Description: controls style of all content between tags with the class name simpleTreeCategory.
Use in application: adds style to category item of category list.
Sample: <div class="simpleTreeCategory">content</div>

Class name: **.simpleTreeDirectParent**
Class attributes:

font-style : italic;

Description: controls style of all content between tags with the class name simpleTreeDirectParent.

Use in application: adds style to category item that is a top level parent of category list.

Sample: <div class="simpleTreeDirectParent">content</div>

Class name: **.simpleTreeHeader**

Class attributes:

font-style: italic;

font-weight: bold;

Description: controls style of all content between tags with the class name simpleTreeHeader.

Use in application: adds style to header text on distribute tab of publishing wizard.

Sample: <div class="simpleTreeHeader">content</div>

Class name: **.simpleTreeTopNode**

Class attributes:

font-style: italic;

font-size: 7pt;

font-weight: bold;

Description: controls style of all content between tags with the class name simpleTreeTopNode.

Use in application: adds style to top node of category list.

Sample: <div class="simpleTreeTopNode">content</div>

Class name: **.textDescription**

Class attributes:

padding-left:10px;

Description: controls style of all content between tags with the class name textDescription.

Use in application: adds spacing around category names in category trees and lists.

Sample: <td class="textDescription">content</td>

Class name: **.textReverseIndent**

Class attributes:

text-indent: -10;

Description: controls style of all content between tags with the class name textReverseIndent.
Use in application: applies spacing style around category names in category trees and lists.
Sample: <div class="textReverseIndent">content</div>

Lotus Notes Styles

Class name: **.pimCalendarDay**
Class attributes:

    font-weight: bold;
    font-size: x-smaller;

Description: controls style of all content between tags with the class name pimCalendarDay.
Use in application: applies style to lotus notes calendar.
Sample: <div class="pimCalendarDay">content</div>

Class name: **.pimCalendarItem**
Class attributes:

    padding-left: 5px;
    color: #666666;
    font-size: 7pt;

Description: controls style of all content between tags with the class name pimCalendarItem.
Use in application: applies style to lotus notes calendar.
Sample: <div class="pimCalendarItem">content</div>

Class name: **.pimTable td**
Class attributes:

    padding: 1px 0px 0px 5px;
    margin: 0px 0px 0px 0px;
    border: 0px solid red;

Description: controls style of all content between <td> tags when embedded within tags with the class name of pimTable.
Use in application: applies style to lotus notes calendar.
Sample: <div class="pimTable"><td>content</td></div>

Quick Search Styles

(located in /lawport-war/includes/searchPanel.css)

Class name: **.tab**

Class attributes:

```
font-family: verdana;
font-size: 7pt;
line-height:140%;
position:absolute;
text-align:center;
border:0px;
border-color:#999999;
border-style:inset;
border-bottom-style:none;
width:80px;
margin:0px;
height: 13px;
cursor: hand;
```

Description: controls style of all content between tags with the class name tab.

Use in application: applies style to tabs in header quick search.

Sample: <div class="tab">content</div>

Class name: **.panel**

Class attributes:

```
font-family: verdana;
font-size: 7pt;
position: absolute;
border: 0px;
border-color :#999999;
border-style:inset;
width: 360px;
height: 30px;
left :0px;
top: 13px;
margin: 0px;
```

padding:0px;

Description: controls style of all content between tags with the class name panel.

Use in application: applies style to content area of each tab in header quick search.

Sample: <div class="panel">content</div>

Class name: **.panelText td**

Class attributes:

```
font-family: verdana;
font-size: 7pt;
line-height: 7px;
letter-spacing: 0px;
text-align: left;
padding: 0px 2px 0px 2px
```

Description: controls style of all content between <td> tags that are embedded between tags with the class name panelText.

Use in application: applies style to text contained within the content area of each tab in header quick search.

Sample: <table class="panel">content</table>

Class name: **.panelText td input**

Class attributes:

```
font-family: verdana;
font-size:7pt;
```

Description: controls style of all content in <input> tags that are embedded between <td> tags and tags with the class name panelText in that order.

Use in application: applies style to text entered into text boxes in the content area of each tab in header quick search.

Sample: <table class="panelText"><td><input/></td></table>

Class name: **.searchBottomGradient**

Class attributes:

```
filter: progid:DXImageTransform.Microsoft.Gradient(GradientType=0, StartColorStr=#eeeeee, EndColorStr=red);
width: 100%;
font-size: 2pt;
```

Description: controls style of all content between tags with the class name searchBottomGradient.
Use in application: applies style to fade bottom of content area of quick search into background color of header.
Sample: <td class="searchBottomGradient"></td>

Class name: **.searchRightGradient**
Class attributes:

```
filter: progid:DXImageTransform.Microsoft.Gradient(GradientType=0, StartColorStr=#eeeeee, EndColorStr=red);
height: 100%;
font-size: 3pt;
```

Description: controls style of all content between tags with the class name searchRightGradient.
Use in application: applies style to fade right side of content area of quick search into background color of header.
Sample: <td class="searchRightGradient"></td>

Resultset Styles

Class name: **table.tablesort thead td**
Class attributes:

```
font: menu;
padding: 0px 10px 0px 5px;
height: 16px;
vertical-align: bottom;
background: buttonface;
cursor: hand;
```

Description: controls style of all content between <td> tags when they are embedded within both <table> & <thead> tags, in that order, and where the <table> tag has the class name tablesort.
Use in application: used widely throughout application in the header for lists of data.
Sample: <table class="tablesort"><thead><td>content</td></thead></table>

Class name: **table.tablesort td**

Class attributes:

border-bottom-color: #708090;
border-bottom-style: solid;
border-bottom-width: 1px;
padding-top: 0px;
padding-bottom: 0px;
background-color: #F0FFFF;

Description: controls style of all content between <td> tags when they are embedded within <table> tags with the class name tablesort.

Use in application: used widely throughout application for lists of data.

Sample: <table class="tablesort"><td>content</td></table>

Class name: **table.tablesort thead .arrow**

Class attributes:

font-family: webdings;
color: black;
font-size: 6pt;
padding-left: 2px;

Description: controls style of all content between tags with the class name arrow when they are embedded within both <table> & <thead> tags, in that order, and where the <table> tag has the class name tablesort.

Use in application: applied dynamically when sorting data by clicking on header title. displays sort direction arrow.

Sample: <table class="tablesort"><thead><div class="arrow">content</div></thead></table>

Class name: **thead td**

Class attributes:

background: buttonface;
font: menu;
padding: 0px 10px 0px 5px;
height: 16px;
vertical-align: bottom;

Description: controls style of all content between <td> tags when they are embedded within <thead> tags.

Use in application: used widely throughout application in the header for lists of data.
Sample: <thead><td>content</td></thead>

Class name: **.tableScroll**
Class attributes:

```
overflow-y: auto;
background-color: #FFFFF0;
width: 100%;
height: 100px;
```

Description: controls style of all content between tags with the class name tableScroll.
Use in application: wraps lists of data, including header and dataset.
Sample: <div class="tableScroll">content</div>

Search Result Highlighting Styles

Class name: **.searchHighlightNav**
Class attributes:

```
background-color: #6699CC;
```

Description: controls style of all content between tags with the class name searchHighlightNav.
Use in application: determines border color of search result highlighting navigation bar.
Sample: <div class="searchHighlightNav">content</div>

Class name: **.searchHighlightNavLinks**
Class attributes:

```
background-color: #F5F5F5;
z-index: 100;
```

Description: controls style of all content between tags with the class name searchHighlightNavLinks.
Use in application: highlights search parameters in search result highlighting navigation bar.
Sample: <div class="searchHighlightNavLinks">content</div>

Tab Styles

Class name: **.activeTab**

Class attributes:

```
color: #000000;
border-bottom: 1px none Black;
border-left: 1px solid Black;
border-right: 1px solid Black;
border-top: 1px solid Black;
padding: 0px 0px 0px 5px;
font-size: 8pt;
font-weight: bold;
filter:progid:DXImageTransform.Microsoft.Gradient(GradientType=0, StartColorStr=#E4F3F8, EndColorStr=white);
```

Description: controls style of all content between tags with the class name activeTab.

Use in application: used to highlight currently selected tab/page in tabbed navigation items.

Sample: <div class="activeTab">content</div>

Class name: **.inactiveTab**

Class attributes:

```
color: Gray;
border-bottom: 1px solid Black;
border-left: 1px solid #778899;
border-right: 1px solid #778899;
border-top: 1px solid #778899;
padding: 0px 0px 0px 5px;
font-size: 8pt;
background-color: #F8F8FF;
```

Description: controls style of all content between tags with the class name inactiveTab.

Use in application: used to shade unselected items in tabbed navigation.

Sample: <div class="inactiveTab">content</div>

Class name: **.tabHightlight**

Class attributes:

border-left: 1px solid #4169E1;
border-right: 1px solid #4169E1;
border-top: 1px solid #4169E1;
color: black;
background-color: #E4F3F8;

Description: controls style of all content between tags with the class name tabHighlight.

Use in application: applied dynamically to shade unselected items in tabbed navigation.

Sample: <div class="tabHighlight">content</div>

Class name: **.tabHightlightActive**

Class attributes:

filter:progid:DXImageTransform.Microsoft.Gradient(GradientType=0, EndColorStr=white, StartColorStr=#C1E6F7);

Description: controls style of all content between tags with the class name tabHighlightActive.

Use in application: applied dynamically to highlight currently selected tab/page in tabbed navigation items.

Sample: <div class="tabHighlightActive">content</div>

Class name: **.tabStandard**

Class attributes:

border-bottom: 1px solid Black;
filter:progid:DXImageTransform.Microsoft.Gradient(GradientType=0, StartColorStr=white, EndColorStr=#B0C4DE);

Description: controls style of all content between tags with the class name tabStandard.

Use in application: provides appearance of shaded background for tabbed navigation.

Sample: <td class="tabStandard">content</td>

Text Styles

Class name: **.disabledText**
Class attributes:
font-size: 8pt;
color: gray;
Description: controls style of all content between tags with the class name disabledText.
Use in application: used application wide for messages when no results were returned for a search or a component list is empty.
Sample: <div class="disabledText">content</div>

Class name: **.disabledTextItalic**
Class attributes:
font-size: smaller;
color: gray;
font-style: italic;
Description: controls style of all content between tags with the class name disabledTextItalic.
Use in application: applies style to proposal header navigation.
Sample: <div class="disabledTextItalic">content</div>

Class name: **.headerSiteName**
Class attributes:
font-size: 12pt;
font-family: arial;
font-weight: bold;
Description: controls style of all content between tags with the class name headerSiteName.
Use in application: applies style to the site name text in the application header.
Sample: <div class="headerSiteName">content</div>

Class name: **.headerSiteType**
Class attributes:
font-size: 8pt;
font-weight: normal;
font-family: arial;

Description: controls style of all content between tags with the class name headerSiteType.
Use in application: applies style to the site type text in the application header.
Sample: <div class="headerSiteType">content</div>

Class name: **.standardText**
Class attributes:

```
font-size: 8pt;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-weight: normal;
color: #000000;
```

Description: controls style of all content between tags with the class name of standardText.
Use in application: applies to text throughout application.
Sample: <table class="standardText">content</table>

Class name: **.textNotAvailable**
Class attributes:

```
filter:progid:DXImageTransform.Microsoft.alpha(Opacity=40);
```

Description: controls style of all content between tags with the class name textNotAvailable.
Use in application: applies style to container items that are empty.
Sample: <div class="textNotAvailable">content</div>

Class name: **.warningText**
Class attributes:

```
color: red;
font-weight: bold;
```

Description: controls style of all content between tags with the class name warningText.
Use in application: applies style to no results and login failed messages.
Sample: <div class="warningText">content</div>

Top Navigation Menu Styles

(located in /lawport-war/includes/menu.css)

Class name: **.menu**
Class attributes:
    font-size: 8pt;
    position: relative;
    z-index: 99;
Description: controls style of all content between tags with the class name menu.
Use in application: this is a style wrapper around the top navigation.
Sample: <div class="menu">content</div>

Class name: **.menu .menuBar**
Class attributes:
    background-color: Menu;
    height: 18px;
Description: controls style of all content between tags with the class name menuBar that are embedded within tags with the class name menu.
Use in application: this is a style wrapper for the top level navigation bar.
Sample: <table class="menu"><tr class="menuBar">content</tr></table>

Class name: **.menu .headerItem**
Class attributes:
font-size: 8pt;
font-weight: bold;
text-decoration: none;
padding-left: 5px;
padding-right: 5px;
padding-top: 3px;
padding-bottom: 3px;
height: 18px;
Description: controls style of all content between tags with the class name headerItem that are embedded within tags with the class name menu.
Use in application: applies style to each of the top level navigation bar menu items.
Sample: <table class="menu"><td class="headItem">content</td></table>

Class name: **.menu .headerItemHover**

Class attributes:

padding-left: 4px;
padding-right: 4px;
padding-top: 2px;
padding-bottom: 2px;
border-bottom: 1px solid ButtonShadow;
border-right: 1px solid ButtonShadow;
border-left: 1px solid ButtonHighlight;
border-top: 1px solid ButtonHighlight;
cursor: hand;

Description: controls style of all content between tags with the class name headerItemHover that are embedded within tags with the class name menu.

Use in application: dynamically applies style to each of the top level navigation bar menu items when hovered over.

Sample: <table class="menu"><td class="headItem">content</td></table>

Class name: **.menu .headerItemDown**

Class attributes:

padding-left: 4px;
padding-right: 4px;
padding-top: 2px;
padding-bottom: 2px;
border-bottom: 1px solid ButtonHighlight;
border-right: 1px solid ButtonHighlight;
border-left: 1px solid ButtonShadow;
border-top: 1px solid ButtonShadow;
cursor: hand;

Description: controls style of all content between tags with the class name headerItem that are embedded within tags with the class name menu after being clicked on.

Use in application: dynamically applies style to each of the top level navigation bar menu items when it is clicked on.

Sample: <table class="menu"><td class="headItem">content</td></table>

Class name: **.subMenu**
Class attributes:
    position: absolute;
    visibility: hidden;
    z-index: 99;
Description: controls style of all content between tags with the class name subMenu.
Use in application: this is a style wrapper around sub menus of the top navigation bar.
Sample: <div class="subMenu">content</div>

Class name: **.subMenu table**
Class attributes:
    font-family: Arial, Helvetica, sans-serif;
    font-size: 10px;
    background-color: Menu;
    padding-left: 10px;
    padding-right: 10px;
    border-bottom: 2px solid ButtonShadow;
    border-right: 2px solid ButtonShadow;
    border-left: 2px solid ButtonHighlight;
    border-top: 2px solid ButtonHighlight;
Description: controls style of all content between <table> tags that are embedded between tags with the class name subMenu.
Use in application: this adds style to each sub menu.
Sample: <div class="subMenu"><table>content</table></div>

Class name: **.subMenu table td**
Class attributes:
    height: 18px;
    color: #000000;
Description: controls style of all content between <td> tags that are embedded between <table> tags and tags with the class name subMenu in that order.
Use in application: this adds style to each menu item contained within a sub menu.
Sample: <div class="subMenu"><table><td>content</td></table></div>

Class name: **.subMenu .arrow**
Class attributes:
font-family: webdings;
font-size: 7pt;
cursor: hand;
Description: controls style of all content between tags with the class name arrow that are embedded between tags with the class name subMenu.
Use in application: this adds the webdings style to submenus items that have child menus (for displaying pointer).
Sample: <div class="subMenu"><td class="arrow">content</td></div>

Class name: **.subMenu .moreArrowOver**
Class attributes:
font-family: webdings;
font-size: 7pt;
cursor: hand;
border-bottom: 1px solid ButtonShadow;
border-right: 1px solid ButtonShadow;
border-left: 1px solid ButtonHighlight;
border-top: 1px solid ButtonHighlight;
Description: controls style of all content between tags with the class name moreArrowOver that are embedded between tags with the class name subMenu.
Use in application: dynamically applied to submenus arrows when hovered over by the mouse pointer.
Sample: <div class="subMenu"><td class="arrow">content</td></div>

Class name: **.subMenu a**
Class attributes:
color: #000000;
text-decoration: none;
Description: controls style of all content between <a> tags that are embedded between tags with the class name subMenu.
Use in application: applies style to subMenu items that are links.
Sample: <div class="subMenu"><a>content</a></div>

Class name: **.menuDisableText**
Class attributes:
    color: gray;
Description: controls style of all content between tags with the class name menuDisableText.
Use in application: applies style to subMenu items that are not links.
Sample: <td class="menuDisableText">content</td>

Class name: **.menuDivider**
Class attributes:
    background-image: url(/graphics/menuSeperator.gif);
Description: inserts background image behind all content contained with tags with the class name menuDivider.
Use in application: used to separate groups of sub-menu navigation bar menu items.
Sample: <td class="menuDivider">content</td>

Class name: **.subItem**
Class attributes:
    color: #000000;
Description: controls style of all content between tags with the class name subItem.
Use in application: applies style to each of the sub-menu navigation bar menu items.
Sample: <td class="subItem">content</td>

Class name: **.subItemHover**
Class attributes:
    background-color: Highlight;
    color: #FFFFFF;
    cursor: hand;
Description: controls style of all content between tags with the class name subItemHover.
Use in application: dynamically applies style to each of the sub-menu navigation bar menu items when hovered over.
Sample: <td class="subItemHover">content</td>

Tree View Styles

(located in /lawport-war/includes/treeView.css)

Class name: **.tree-container**
Class attributes:

```
margin: 0px;
padding: 0px;
white-space: nowrap;
```

Description: controls style of all content between tags with the class name of tree-container.
Use in application: style wrapper around navigation component administration tree tool.
Sample: <div class="tree-container">content</div>

Class name: **.tree-item**
Class attributes:

```
padding: 0px 1px 0px 1px;
margin: 0px 1px 0px 1px;
white-space: nowrap;
vertical-align: middle;
```

Description: controls style of all content between tags with the class name of tree-item.
Use in application: applies style to navigation items within a navigation component on the administration tree tool.
Sample: <td class="tree-item">content</td>

Class name: **.tree-item a, .tree-item a:active, .tree-item a:hover**
Class attributes:

```
margin-left: 3px;
padding: 1px 2px 1px 2px;
```

Description: controls style of all content between <a> tags when embedded within tags with the class name of tree-item.
Use in application: applies style to linked navigation items within a navigation component on the administration tree tool.
Sample: <td class="tree-item"><a>content</a></td>

Class name: **.tree-item a**

Class attributes:

color: black;

text-decoration: none;

white-space: normal;

Description: controls style of all content between <a> tags when embedded within tags with the class name of tree-item.

Use in application: applies style to linked navigation items within a navigation component on the administration tree tool.

Sample: <td class="tree-item"><a>content</a></td>

Class name: **.tree-item a:hover**

Class attributes:

color: blue;

text-decoration: underline;

Description: controls style of all content between <a> tags when embedded within tags with the class name of tree-item when hovered over by mouse pointer.

Use in application: dynamically applies style to linked navigation items within a navigation component on the administration tree tool when hovered over by mouse pointer.

Sample: <td class="tree-item"><a>content</a></td>

Class name: **.tree-item a:active**

Class attributes:

background: highlight;

color: highlighttext;

text-decoration: none;

Description: controls style of all content between <a> tags when embedded within tags with the class name of tree-item when it is the last item clicked on by the user.

Use in application: dynamically applies style to linked navigation items within a navigation component on the administration tree tool when clicked on by the user.

Sample: <td class="tree-item"><a>content</a></td>

Class name: **.tree-item img**
Class attributes:
vertical-align: middle;
border: 0px;
Description: controls style of all content between <img> tags when embedded within tags with the class name of tree-item.
Use in application: applies style to image icons for each navigation item within a navigation component on the administration tree tool.
Sample: <td class="tree-item"><img>content</img></td>

Class name: **.tree-icon**
Class attributes:
width: 16px;
height: 16px;
Description: controls style of all content between tags with the class name of tree-icon.
Use in application: not currently used in application.
Sample: <td class="tree-icon">content</td>

Class name: **.tree-selected**
Class attributes:
border-left: 1px solid #4169E1;
border-top: 1px solid #4169E1;
border-bottom: 1px solid #4169E1;
border-right: 1px solid #4169E1;
color: #000000;
background-color: #E4F3F8;
Description: controls style of all content between tags with the class name of tree-selected.
Use in application: dynamically applies style to navigation items within a navigation component on the administration tree tool when clicked on by the user.
Sample: <td class="tree-selected">content</td>

Class name: **.containerSelect**

Class attributes:

```
background-color: #F5F5F5;
border: 1px ridge Silver;
z-index: 100;
position: relative;
filter:progid:DXImageTransform.Microsoft.Shadow(color="#777777", Direction=135,
Strength=3)alpha(Opacity=95);
-moz-opacity:0.9;;
```

Description: controls style of all content between tags with the class name of containerSelect.

Use in application: style wrapper around folder picker window when moving navigation items within a navigation component on the navigation component administration screen.

Sample: <td class="containerSelect">content</td>

Index