Introduction

Welcome to the Alagad WysiPad 2 Browser Based WYSIWIYG HTML Editor ActiveX control developer documentation. This documentation is indented to help you, a web application developer, integrate Alagad WysiPad 2 WYSIWIYG HTML Editor into your web application.

WysiPad 2 offers and easy, customizable, browser based WYSIWYG HTML editor as an alternative to clumsy text boxes.

This documentation assumes that you have some experience with HTML and another server side technology such as ASP, ColdFusion, CGI, JSP, PHP or any other such tool.

Conventions Used In This Document:

> Words surrounded in angle brackets indicate HTML tags. For example, <object> indicates the HTML Object tag.
>
> Italicized words indicate HTML tag attributes. For example *bgcolor* indicates a Bgcolor attribute.
>
> Words highlighted in blue indicate WysiPad 2 parameters. For example pathToImageList indicates the pathToImageList parameter.
>
> Words highlighted in blue followed by parenthesis indicate WysiPad 2 methods. For example getContent() indicates the getContent method.
>
> Paths to files or directories are surrounded in quotation marks. For example "/WysiPad2".
>
> HTML code is highlighted in brown. For example: `<p>Example</p>`
>
> The "¬" character at the end of a line of example code indicates a line which should be continued on the same line, even though it's shown as wrapping.

About WysiPad 2

Alagad WysiPad 2 HTML editor is the only tool you need for browser-based WYSIWYG HTML authoring and editing. Now there is an alternative to text area fields! Simply embed WysiPad 2 wherever you need to provide access to HTML formatted content. The documentation and sample files make it easy to get up and running in no time. WysiPad HTML Editor works with virtually any servside technology!

WysiPad 2 is easy to install. – Simply point your browser to a page using the editor and follow the simple instructions.

WysiPad 2 is customizable. – With over 100 customization parameters, WysiPad 2 gives Developers the ability to programmatically enable or disable almost every feature. You get to choose if your users can add links, images or tables. Restrict your users from changing font colors, or enable it as you choose. You have the choice to give your users access to edit the generated HTML source, or you prevent them from ever seeing it.

WysiPad 2 uses your Cascading Style Sheet – Many sites use CSS to determine how the site will appear. Point WysiPad 2 to your style sheet and force your users to adhere to site standards.

WysiPad 2 Features

- WYSIWYG HTML editing
- Open and edit HTML files from your computer
- Save content directly to the server
- Standard text formatting features
    - Bold, italic, underline
    - Fonts, font sizes, block formatting
    - Cut, Copy, Paste
    - Etc.
- Spell Checking
- Human readable HTML
- Visual aids for easier editing
- The ability to insert images into the document
- The ability to upload images to your server
- The ability to delete images from your server
- Cell-level table editing
- Special characters
- Keeps sessions alive, even when users type for durations longer than the site's session timeout configuration
- Ability to insert Internet or Email links manually or exactly as provided by the server
- Supports Cascading Style Sheets
- Apply CSS classes to objects in your content
- Ability to link to external help pages from the WysiPad 2 user interface
- Developer-customizable user interface
    - Enable or remove almost every interface element
- A free 30 day trial on any domain, for every user.
    - Download and use live for 30 days free.
- Much, much, more!

WysiPad 2 is an HTML body editor. WysiPad 2 creates content for placement between <body> tags. WysiPad 2 does not return any HTML which would appear outside of the <body> tag such as <html>, <head>, or <title> tags.

How to Contact Alagad Support

The preferred method to contact Alagad Technical Support is to us the contact us form located on http://www.alagad.com. We will make every effort to respond to messages for support as quickly as possible.

Files Distributed with WysiPad 2

WysiPad 2 is distributed in a Zip file format. The following files are distributed in the package:

The following 8 files are the files required when using WysiPad 2 on your website. All of these files must be in the same directory. The path to the WysiPad2.cab file will be referenced by your <object> tag. For more information see the "How to Setup WysiPad" section.

ComDlg32.cab - WysiPad 2 Support Cab File
MSComCtl.cab - WysiPad 2 Support Cab File
MSInet.cab - WysiPad 2 Support Cab File
MsStkPrp.cab - WysiPad 2 Support Cab File
MSWinSck.cab - WysiPad 2 Support Cab File
RichTx32.cab - WysiPad 2 Support Cab File
VBRun60.cab - WysiPad 2 Support Cab File
WysiPad2.cab - WysiPad 2 Cab File

The version.txt file contains the exact version of WysiPad which is being distributed. This must be referenced correctly in your <object> tag. For more information see the "How to Setup WysiPad" section.

version.txt - WysiPad 2 Version Information

The documentation directory contains this documentation file. Additionally, there are three example applications written in ASP, ColdFusion and PHP. For more information on these example applications see the readme.txt file in each directory.

\ Documentation\WP2_ASP_Example - An example ASP application
\ Documentation\WP2_CFM_Example - An example ColdFusion application
\ Documentation\WP2_PHP_Example - An example PHP application

WysiPad Licensing

WysiPad 2 is licensed on a per-domain basis. WysiPad 2 *must* be licensed for *each* domain WysiPad is accessed on. This license is enforced by a license key which is specific to the domain you are accessing WysiPad on.

There are three types of licenses avaliable:

Domain License – A Domain license gives you the right to provide access to WysiPad 2 to your users on a domain which you have licensed or any sub-domain of that domain. For example, if you licensed "example.com" you would be free to use WysiPad on "example.com", "www.example.com", "another.example.com", and any other sub-domain of "example.com". You would not be licensed to use WysiPad 2 on any other domain name, such as "anotherdomain.com" without purchasing an additional license. This is enforced by matching the domain parameter within your <object> tag against the licensed domain specified in the key parameter. For more information see the "How to Setup WysiPad" section in this document. You must purchase a Domain License for each domain you wish to use or display WysiPad on.

Enterprise License – An Enterprise license gives you the right to provide access to WysiPad 2 to your users via any domain name. The only restriction is that you must obtain a separate license key for each domain. Sub-domains are covered by a license key for their domain. For instance, if you were to obtain a license key for "example.com" it would also work for "www.example.com" but would not work for "anotherdomain.com." To use WysiPad on "anotherdomain.com" you would have to obtain a new license key. This is enforced by matching the domain parameter within your <object> tag against the licensed domain specified in the key parameter. For more information see the "How to Setup WysiPad" section in this document. *There is no additional cost for each license key once you have purchased an Enterprise license*

Unrestricted Enterprise License – An Unrestricted Enterprise license gives you the right to provide access to WysiPad to your users via any domain name with one license key. There is no need to obtain a new key for each domain.

WysiPad 30 Day Trial

WysiPad 2 allows for a free 30 day trial period for each client web browser on any domain. This means that from the point which a new user first runs WysiPad 2 on your domain that they can continue to use WysiPad 2 on your domain for free for a period of 30 days. As a developer you are also restricted to the 30 day trial period.

As an example:

You elect to test WysiPad 2 on your website which is hosted on "example.com". You have downloaded WysiPad and configuring it to run on your website. The first time you open the page which makes use of WysiPad will start your free 30 day trial period. After working with WysiPad for 5 days you invite your client to look at the page which uses WysiPad. When they first open this page they will start their 30 day trial period which will expire 5 days after yours.

During the trial period there are no popup messages or warnings. You can look under the Help > About… menu to determine how many days remain in your trial period.

After working with WysiPad for a while you elect to purchase a license key for "example.com". To complete the licensing process you simply add a key parameter with its value set to your new license key's value. There will be no obvious transition for you or your users. The Help > About… window will indicate that you are licensed to use WysiPad 2 on "example.com".

If you do not purchase a license key before the end of your thirty day trial you will received a message indicating that the trial period has expired. Your client, who first accessed WysiPad five days after you will continue to have access until their 30 days have expired at which point they will receive the same notice.

How to Setup WysiPad

To set up WysiPad you must first extract all of the .cab files from the distribution .zip file into a directory under your website, for example "/WysiPad2". For more information on WysiPad 2 files see the "Files Distributed with WysiPad 2" section.

To display and use WysiPad 2, your webpage must contain specific code which references the WysiPad 2 .cab file on your web server. The <object> tag is used by Internet Explorer on Windows to instruct the browser to load the WysiPad 2 ActiveX control. Multiple <param> tags are nested within your <object> tag to provide or restrict WysiPad 2 functionality.

If you extracted the WysiPad .cab files into a directory "/WysiPad2" under your website's root folder, a minimal <object> and <param> tag configuration would look like this:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">
</object>
```

The <object> tag takes the following attributes:

Attribute Name	Description
ID	The *ID* attribute is used to uniquely identify this particular usage of WysiPad. Each <object> tag must have a unique value for its *ID* attribute. The *ID* attribute is used when updating parameter values or reading content from WysiPad.
Classid	The *Classid* attribute must be provided and must be "CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184". This uniquely identifies the WysiPad ActiveX control to Windows. WysiPad will not work if this is not correct.
Codebase	The *Codebase* attribute is made up of two parts, the path to the WysiPad .cab files on your web server and the version number you are using. For more information see "The Codebase Attribute Explained" section.
Width	The *Width* attribute sets the width of WysiPad in pixels.
Height	The *Height* attribute sets the height of WysiPad in pixels.

The <param> tag takes the following attributes:

Attribute Name	Description
Name	The *name* attribute specifies the name of the parameter you are providing. For instance, if you want to set the domain parameter then the *name* attribute value would be “domain”.
Value	The *value* attribute specifies the value of the parameter identified by the *name* attribute. For instance, if your *name* attribute was set to “domain” and your domain name is “www.example.com” then your *value* attribute would be set to “www.example.com”.

WysiPad 2 requires the domain and port parameters and will not function without them. These must be set to the domain name and port you are using WysiPad on. (When in doubt, the port is probably 80.)

After licensing WysiPad 2 you must also provide the key parameter to with its *value* attribute set to the your license key string. For all licenses other than the Unrestricted Enterprise License the domain name referenced in the domain parameter must match or be a sub-domain of the licensed domain.

The licenses key is provided as a long string of letters and numbers. This string does not have any spaces and should not have any line breaks. Simply place the entire string into the key parameter’s *value* attribute. For more information about WysiPad 2 licenses see the “WysiPad Licensing” section.

The Codebase Attribute Explained

The *codebase* attribute is made up of two particularly important parts. The first part is the path to the WysiPad2.cab file. The second part indicates the exact version of WysiPad 2 that you are using. These are separated by a pound sign.

Example:

/wysipad2/WysiPad2.CAB#version=2,1,0,252

In this example the path to WysiPad is /wysipad2/WysiPad2.CAB and the version in use is 2,1,0,252. You must set the version number correctly or WysiPad may not install at all or will download and install on every visit to your page. You can determine the version number you are using by opening the version.txt file that was distributed with WysiPad.

When Internet Explorer finds an <object> tag in your document it will first look at the *classid* attribute to determine if this object is already installed. In not, it will download and install the object specified in the *codebase* attribute regardless of the provided version number.

If you already have WysiPad 2 installed, Internet Explorer will compare the version number of the installed WysiPad 2 against the version number in the codebase attribute. If the version number in the *codebase* attribute is greater than the version number of the version number of the installed WysiPad 2, then Internet Explorer will download and install the file specified in the *codebase* attribute. If the version number in the *codebase* attribute is less than or equal to the installed WysiPad 2 version number, then Internet Explorer will not download the specified file and will execute the installed version of the WysiPad 2.

Having the *codebase* attribute version number set incorrectly can cause problems. For instance, assume you have configured your website to work with WysiPad 2, version 2.1.0.252. A month later Alagad

Inc. releases an updated version 2.1.0.253 which you download and extract correctly. If you do not update the version number in your *codebase* attribute then users with 2.1.0.252 will not install the new version.

As another example, assume you are actually using version 2.1.0.252 on your website, but have your version number set as 2.1.0.253. Users who do not have 2.1.0.253 or later installed will download and install WysiPad 2 on every page request. This occurs because Internet Explorer knows that it has version 2.1.0.252 installed, and that you are requesting version 2.1.0.253. Because IE thinks the version on your server is newer it will download and install it. However, because IE is actually installing 2.1.0.252 the version number of the installed WysiPad 2 does not change. This process will repeat for each page request until the version number in your *codebase* attribute is set to the correct version number.

You can tell what version number of WysiPad 2 is installed by clicking Help > About… within the WysiPad 2 interface.

Using WysiPad 2 Parameters

WysiPad 2 provides over two hundred parameters for your use. These parameters can be used to provide or remove functionality from the WysiPad 2 interface as well as setting the content to display and retrieving the edited content. For instance, if you wanted to restrict your users so that they could not edit the HTML generated by WysiPad, then you could provide the disableHtmlView parameter with a value of 1 to indicate true.

To disable the HTML view in the document you could add the disableHtmlView parameter like this:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">

      <!-- This removes the HTML view tab -->
      <param name="disableHtmlView" value="1">
</object>
```

Setting the Content to Edit

To set the content to display and edit you can either use the content or the pathToContent parameters.

Setting the Content with the Content Parameter

The content parameter’s value indicates the actual content you wish to edit. For example:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">

      <!-- This is the content I'm editing -->
      <param name="content"
            value="<p>This is my content</p>">
</object>
```

When providing content with the content parameter you must always replace every instance of the "&" character with the HTML equivalent "&amp;". This is true even when the "&" character is used within another HTML equivalent.

For example, assume your content is the following string:

```
      <p>This &lt;Body&gt; tag holds contains body of your document.</p>
```

This string would be expressed correctly as:

```
      <p>This &amp;lt;Body&amp;gt; tag holds contains body of your document.</p>
```

You must *also* replace every use of quotation marks in content with the HTML equivalent, "&quot;" or "&#34;". **This must be done after replacing the "&" characters or your document will not display correctly.**

For example, assume your content is the following string:

```
      <p>Copyright &copy; 2001
      <a href="http://www.alagad.com">Alagad Inc.</a></p>
```

This string would be expressed correctly as:

```
      <p>Copyright &amp;copy; 2001
      <a href=&quot;http://www.alagad.com&quot;>Alagad Inc.</a></p>
```

Notice that the "&" character in the "&quot;" was *not* replaced with "&amp;". This is because the quotation mark characters were replaced *after* the ampersand characters.

WysiPad 2 can only be used to edit the body of an HTML document. Any portions of a document provided outside of a <body> tag will be removed from the content and ignored. If the content does not have a <body> tag WysiPad 2 will assume the entire document is <body> content.

Setting the Content with the PathToContent Parameter

The pathToContent parameter's value indicates a document on your server which when called will return the content you wish to edit. For example:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">

      <!-- This is the content I'm editing -->
      <param name="pathToContent"
            value="/content.asp">
</object>
```

This file referenced by the pathToContent parameter must return raw HTML to WysiPad 2. The example above uses an ASP file as an example but there are no restrictions on the language or technology used to retrieve the content.

You can also pass URL variables within the *value* attribute. For example:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">

      <!-- This is the content I'm editing -->
      <param name="pathToContent"
            value="/content.asp?id=75">
</object>
```

You can also point the pathToContent parameter directly to specific static HTML documents.

Unlike setting the content parameter, you do not need to replace any characters in the returned HTML.

WysiPad 2 can only be used to edit the body of an HTML document. Any portions of a document provided outside of a <body> tag will be removed from the content and ignored. If the content does not have a <body> tag WysiPad 2 will assume the entire document is <body> content.

For more information on pathTo parameters see the "Using WysiPad 2 PathTo Parameters" section.

Retrieving the Edited Content

You can use client side scripting to access any parameter or method of WysiPad 2. All examples in this document use JavaScript.

You can access an instance of WysiPad in your document by its *ID* attribute as specified in its <object> tag. If your object's *ID* is "ctlWysiPad2" you could use JavaScript to access WysiPad 2 like this:

```
      document.ctlWysiPad2
```

WysiPad does not automatically submit its content when you submit forms in your HTML document. To retrieve content from WysiPad 2 you must use client side scripting to either read the value of the content parameter or execute the getContent() method. The resulting data can then be placed in a form field and submitted with your form. The following subsections cover two ways to do this.

Retrieving Content with the Content Parameter

You can read the value of the content parameter from JavaScript like this:

```
      document.ctlWysiPad2.content
```

The following example shows how to read content into a hidden form field when submitting a form.

```
<form name="editContent" action="/action.cfm" method="post">

<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">

      <!-- This is the content I'm editing -->
      <param name="content"
            value="<p>This is my content.</p>">
</object>

<!-- This hidden form field is used to post the contents from WysiPad to the form
action -->
<input type="hidden" name="content" value="">

<p>
<input type="Submit" name="Submit" value="Submit"
      onClick="document.editContent.content.value = document.ctlWysiPad2.content;">
</p>

</form>
```

Retrieving Content with the GetContent() Method

You can also read the content from WysiPad using the getContent() method from JavaScript. This method takes an attribute which indicates what type of content you would like to receive.

To return the HTML content:

```
      document.ctlWysiPad2.getContent(0)
```

To return the textual content of the document with all HTML removed:

```
      document.ctlWysiPad2.getContent(1)
```

The following example shows how to retrieve both the HTML and text content from WsyiPad when submitting a form:

```
<script language="JavaScript">
      function jsGetContent(){
            document.editContent.htmlContent.value = ¬
                  document.ctlWysiPad2.getContent(0);
            document.editContent.textContent.value = ¬
                  document.ctlWysiPad2.getContent(1);
      }
</script>

<form name="editContent" action="/action.cfm" method="post">

<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">

      <!-- This is the content I'm editing -->
      <param name="content"
            value="<p>This is my content.</p>">
</object>

<!-- This hidden form field is used to post the HTML content
from WysiPad to the form action -->
<input type="hidden" name="htmlContent" value="">

<!-- This hidden form field is used to post the text content
from WysiPad to the form action -->
<input type="hidden" name="textContent" value="">

<p>
<input type="Submit" name="Submit" value="Submit"
onClick="jsGetContent();">
</p>

</form>
```

Disabling and Enabling Interface Elements

WysiPad 2 provides the ability to turn on or off almost every interface element. For instance, if you wanted to remove the File menu from the menu bar you could add the disableFileMenu parameter with its value set to 1.

All of the parameters which are used to enable or disable interface elements begin with "disable" or "hide". Parameters which begin with "disable" actually remove the functionality completely from the interface. Parameters which begin "hide" remove the functionality but not associated shortcuts. For instance, the hideCopy parameter removes the Edit > Copy menu item and the Copy icon in the toolbar but does not remove the Ctrl-C shortcut.

Disabling or hiding a feature does not guarantee that users will not be able to implement the feature within the HTML view. For example, if you use the disableTables parameter to remove the ability to insert tables into the document, but do not disable access to the HTML view then users could manually add their own tables in the HTML view.

Each of the "disable" or "hide" parameters takes 1 for true or 0 for false.

Some functionality such as image support and saving directly to the server will be disabled if you do not provide the necessary support files in with pathTo parameters. This is true even if you explicitly enable the feature. For instance, if you do not provide a pathToImageList parameter the Images icon in the tool bar and the Insert > Images menu items will not appear, even if you have specifically set the disableImages parameter to false.

For a complete list of "disable" and "hide" parameters see the "Interface Parameters" section.

Using WysiPad 2 PathTo Parameters

Some WysiPad 2 parameters specify files created by you which are hosted on your server and perform a specific function. Each of these parameter names start with "pathTo" and are referred to as pathTo parameters. For instance, the pathToStyleSheet parameter provides the path to a style sheet on your server which will be used when displaying WysiPad 2. The pathToLinkList parameter indicates the path to a file on your server which provides a list of link names and addresses.

WysiPad 2 makes a standard HTTP request to access all of the files referenced by pathTo parameters. The HTTP request passes cookies, URL variables and Form variables to the document specified in the pathTo attribute. Because of this, you can write your pathTo files in any programming language supported by your web server. The following <object> tag uses the pathToStyleSheet attribute:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">
```

```
	<!-- the domain and port parameters are required -->
	<param name="domain" value="www.example.com">
	<param name="port" value="80">

	<!-- This is the content I'm editing -->
	<param name="content"
		value="<p>This is my content.</p>">

	<!-- This sets the Style Sheet to use -->
	<param name="pathToStyleSheet"
		value="/styles/styles.css">
</object>
```

The URL for the style sheet referenced above could be expressed as:

http://www.example.com:80/styles/styles.css.

If all of the images and pathTo files are housed under specific root directory on your website, you can use the documentRoot parameter to indicate the root directory for WysiPad to look under. For example:

```
<object id="ctlWysiPad2"
	classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
	codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
	width="600" height="400">

	<!-- the domain and port parameters are required -->
	<param name="domain" value="www.example.com">
	<param name="port" value="80">

	<!-- This is the content I'm editing -->
	<param name="content"
		value="<p>This is my content.</p>">

	<!-- This sets the Style Sheet to use -->
	<param name="pathToStyleSheet"
		value="/styles/styles.css">

	<!-- This sets the document root -->
	<param name="documentRoot"
		value="/pathToFiles">
</object>
```

In this example the URL for the style sheet referenced below could be expressed as:

http://www.example.com:80/pathToFiles/styles/styles.css.

A set of example applications are provided with the WysiPad .zip file. The examples are provided for ASP, ColdFusion and PHP. These are intended to give you a working example of how to implement WysiPad with the various pathTo files.

ColdFusion users in particular should be aware that the ColdFusion server's debug output may break some pathTo pages and should either be disabled in the ColdFusion administrator or suppressed within pathTo documents buy using the ColdFusion code:

```
<cfsetting showDebugOutput="No">
```

When developing pathTo documents it is suggested that you turn WysiPad's debugging mode on. For more information see the "Debugging PathTo Documents" section.

For more information on the various pathTo parameters see the "Full PathTo Parameters Details" section.

Debugging PathTo Documents

Custom programming always requires debugging. By default, WysiPad does not provide any feedback on bugs or errors in your pathTo documents. *This is usability feature* which prevents your users from receiving error messages which will not be useful to them. When debugging mode is turned on WysiPad will return the results of every HTTP request except those for images displayed within the content.

WysiPad provides the ability to turn on debugging messages by using the debugMode parameter with a value of 1. Here's an example:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">

      <!-- This is the content I'm editing -->
      <param name="content"
            value="<p>This is my content.</p>">

      <!-- This sets the location of the image list -->
      <param name="pathToImageList"
            value="/imageList.php">

      <!-- Turn on debug mode -->
      <param name="debugMode" value="1">
</object>
```

When debugMode is set to 1 the results of HTTP requests will be displayed. This allows you to see exactly what is being returned by your documents and will help you to find and fix problems. With debugMode set to 1 you will also receive error messages if WysiPad is unable to parse your document or if there were other errors.

Cookie and URL Parameters

It is assumed that most developers will integrate WysiPad into dynamic websites. These dynamic websites may require users to login before they are given access to the application. Additionally, they may be modular or dynamic to the point where various URL parameters will modify the exact behavior of a given document.

Most dynamic websites use some amount of cookies and/or server-side session variables to maintain state in the application. Frequently the server side variables are associated with a particular user by a unique cookie placed on their computer when their session began. ASP, ColdFusion, PHP and almost all others use at least one client side cookie for this purpose.

When a user accesses your website, Internet Explorer will receive cookies names and values and save them locally in RAM or to disk. These cookies are not automatically handed to WysiPad. When WysiPad makes HTTP requests to pathTo documents these cookies will not be sent back to the server unless you tell WysiPad to send them.

To do so, you can use a set of up to ten cookieName and cookieValue parameters to set name and value of your cookies. The following example uses ColdFusion’s CFid and CFtoken cookies:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">

      <param name="pathToContent" value="/getContent.cfm”>

      <!-- This is the content I'm editing -->
      <param name="content"
            value="<p>This is my content.</p>">

      <!-- set the CFid cookie -->
      <param name="cookieName1"
            value="CFID">
      <param name="cookieValue1"
            value="<CFoutput>#cookie.cfid#</CFoutput>">

      <!-- set the CFtoken cookie -->
      <param name="cookieName2"
            value="CFTOKEN">
      <param name="cookieValue2"
            value="<CFoutput>#cookie.cftoken#</CFoutput>">
</object>
```

You can set WysiPad to send up to ten cookies with each request. For more information on the cookie parameters see the “Complete List of Cookie Parameters” section.

The example applications provided in the WysiPad 2 .zip file show how to access the session cookies for ASP, ColdFusion and PHP.

You may also want to send URL variables when making pathTo documents. This can be accomplished by simply appending the URL variable to the pathTo parameter's *value* attribute. For example:

```
	<param name="pathToContent" value="/getContent.cfm?id=75">
```

However, if you want to send the same URL variables with every HTTP request then you can use up to ten urlName and urlValue parameters. These follow the same pattern as the cookieName and cookieValue parameters. For example:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">

      <param name="pathToContent" value="/getContent.cfm”>

      <!-- This is the content I'm editing -->
      <param name="content"
            value="<p>This is my content.</p>">

      <!-- set a URL variable -->
      <param name="urlName1"
            value="ID">
      <param name="urlValue1"
            value="75">

      <!-- set another URL variable -->
      <param name="section"
            value="CFTOKEN">
      <param name="urlValue2"
            value="admin">
</object>
```

The URL for pathToContent in the example above could be expressed as:

http://www.example.com:80/getContent.cfm?id=75&section=admin

Complete List of Cookie Parameters

The following is a complete list of cookie parameters:

Cookie Name Parameter	Options	Cookie Value Parameter	Options
cookieName1	Valid cookie name	**cookieValue1**	String

cookieName2	Valid cookie name	**cookieValue2**	String
cookieName3	Valid cookie name	**cookieValue3**	String
cookieName4	Valid cookie name	**cookieValue4**	String
cookieName5	Valid cookie name	**cookieValue5**	String
cookieName6	Valid cookie name	**cookieValue6**	String
cookieName7	Valid cookie name	**cookieValue7**	String
cookieName8	Valid cookie name	**cookieValue8**	String
cookieName9	Valid cookie name	**cookieValue9**	String
cookieName10	Valid cookie name	**cookieValue10**	String

Complete List of URL Parameters

The following is a complete list of URL parameters:

URL Name Parameter	Options	URL Value Parameter	Options
URLName1	Valid URL variable name	**URLValue1**	String
URLName2	Valid URL variable name	**URLValue2**	String
URLName3	Valid URL variable name	**URLValue3**	String
URLName4	Valid URL variable name	**URLValue4**	String
URLName5	Valid URL variable name	**URLValue5**	String
URLName6	Valid URL variable name	**URLValue6**	String
URLName7	Valid URL variable name	**URLValue7**	String
URLName8	Valid URL variable name	**URLValue8**	String
URLName9	Valid URL variable name	**URLValue9**	String
URLName10	Valid URL variable name	**URLValue10**	String

Using Style Sheets

By using the pathToStyleSheet parameter you can use a Cascading Style Sheet to control the appearance of your content. CSS classes within the Style Sheet show up under the Style > CSS Classes menu. If there are no classes provided in the Style Sheet then CSS Classes submenu does not show up. To include a Style Sheet provide the pathToStyleSheet parameter.

CSS Classes can be applied to your content by making a selection and clicking on a class name under the Styles > CSS Classes menu.

Example usage of the pathToStyleSheet parameter:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">
      <!-- This is the content I'm editing -->
      <param name="content"
            value="<p>This is my content.</p>">

      <!-- This sets the Style Sheet to use -->
      <param name="pathToStyleSheet"
            value="/styles/styles.css">
</object>
```

Be aware that the Style Sheet is not actually embedded or linked to the document except for display purposes within WysiPad. When retrieving content from WysiPad you will not receive any <link> or <style> tags which would have been nested in the <head> tag. Items in the document which have had CSS class applied to them will have their *class* attribute set. When displaying this content you will want to be sure to include a reference to the Style Sheet.

Using Images in WysiPad

You can enable the ability to insert images from your server into your content. You can also give your users the ability to upload to and delete images from your server. There are several image related parameters available.

Image Related Parameters:

Parameter	Options	Description
disableDeleteImage	0 = True 1 = False	Disables the image delete feature. This functions even if the pathToImageDelete parameter is provided.
disableEditImageWidthHeight	0 = True 1 = False	Disables the Width and Height fields in the Images Window and the Image Properties popup and resets the width and height when images are resized. Does not disable the resize handles on images. Does not automatically remove width and height on all images in the content.
disableUploadImage	0 = True 1 = False	Disables the image upload feature. This functions even if the pathToImageUpload parameter is provided.
maxUploadSize	Positive Integer	Sets the maximum size allowed for uploaded files in bytes. Users who attempt to upload files larger than this size will be presented with an informative message stating that their file is too large to upload.
pathToImageDelete	Path To File	Indicates a file used to delete images off the server. For more information see “Deleting Images” below. When provided the Delete tab is shown in the Images Window.
pathToImageList	Path To File	Indicates a file used to provide a list of images on the server to WysiPad. For more information see “Enabling Images in WysiPad” below. When provided the Upload tab is shown in the Images Window.
pathToImageUpload	Path To File	Indicates a file used for uploading images to the server. For more information see “Uploading Images” below. When provided the Insert > Image menu item is shown as is the Images icon on the toolbar.

Enabling Images in WysiPad

To enable the ability to insert images into your content you must provide the pathToImageList parameter. The pathToImageList parameter indicates a document on your server which must return a document containing specific information in a specific format.

The pathToImageList document uses a simple WysiPad specific markup similar to HTML and XML to create the list. The following is an example image list:

```
<wpImageList>
      <wpImage>
            <wpImageName>Alagad Homepage</wpImageName>
            <wpImageSize>773</wpImageSize>
            <wpImageWidth>151</wpImageWidth>
            <wpImageHeight>101</wpImageHeight>
            <wpImagePath>/images/hp_screenshot1.jpeg</wpImagePath>
      </wpImage>
      <wpImage>
            <wpImageName>Couple Walking on Beach</wpImageName>
            <wpImageSize>48</wpImageSize>
            <wpImageWidth>9</wpImageWidth>
            <wpImageHeight>4</wpImageHeight>
            <wpImagePath>/stockart/img1234.jpeg</wpImagePath>
      </wpImage>
      <wpImage>
            <wpImageName>Logo</wpImageName>
            <wpImageSize>15323</wpImageSize>
            <wpImageWidth>330</wpImageWidth>
            <wpImageHeight>86</wpImageHeight>
            <wpImagePath>/images/Logo.gif</wpImagePath>
      </wpImage>
</wpImageList>
```

This list denotes three images. Each image has an associated name, file size, width, height and path to the image.

The image list has the following hierarchy of tags:

- wpImageList
  - wpImage
    - wpImageName
    - wpImageSize
    - wpImageWidth
    - wpImageHeight
    - wpImagePath

Tag Name	Description
wpImageList	The wpImageList tag surrounds the entire list of images.
wpImage	Each image is denoted with a wpImage tag. This tag surrounds five tags which describe the image to WysiPad.
wpImageName	Indicates the name of the image as it will be displayed in the Images window and the default alt tag value.
wpImageSize	Indicates the size of the image in bytes. The image size is displayed over the image preview.
wpImageWidth	The width of the image in pixels. This is used to populate the width field in the image window.

wpImageHeight	The height of the image in pixels. This is used to populate the height field in the image window.
wpImagePath	The path to the image file. This is the path used in <img> tags within your content.

You should not provide any additionally content inside or outside of these tags as this may cause WysiPad to malfunction.

For specifics on the pathToImageUpload parameter see the “Full PathTo Parameters Details” section.

Uploading Images

The pathToImageUpload parameter specifies a file which is used to upload images from WysiPad to your server. The purpose of this file is to receive the posted Image file and a small set of data on the file.

This document receives the following posted variables:

Form Field Name	Description
wpImageName	This is the name of the image as entered in the Image Name field in the Upload tab on the Image Window.
wpImageSize	This is the size in bytes of the file.
wpImageWidth	The width of the image.
wpImageHeight	The height of the image
wpImageFile	The data of the image file being uploaded. This is in the same format as any file upload from a form with an enctype of multipart/form-data.

You can use these form fields to upload the image to your server. By storing this information in a database you can easily use it to supply data back to pathToImageList document.

By supplying the maxUploadSize parameter your users will be restricted to uploading files smaller than the specified size.

This file does not return information to WysiPad so no output is required. You can output anything you wish without impacting WysiPad.

For more information on the pathToImageUpload parameter see the “Full PathTo Parameters Details” section.

Deleting Images

The pathToImageDelete parameter specifies a file which is used to delete images from your server. This file receives a set of form variables posted from WysiPad. You can use these to identify your image and delete it.

Form fields received from WysiPad:

Form Field Name	Description

wpImageName	This is the name of the image as provided in the pathToImageList document.
wpImageSize	This is the size in bytes of the file as provided in the pathToImageList document.
wpImageWidth	The width of the image as provided in the pathToImageList document.
wpImageHeight	The height of the image as provided in the pathToImageList document.
wpImagePath	The path to the image on the server as provided in the pathToImageList document.

You should use this information to delete the file off you server or in some way prevent the pathToImageList file from continuing to return this image.

This file does not return information to WysiPad so no output is required. You can output anything you wish without impacting WysiPad.

Using Links in WysiPad

By default WysiPad 2 allows users to create hyperlinks or email links. In addition to this you can provide a list of hyperlinks and email links by using the pathToLinkList and pathToEmailLinkList parameters respectively.

The following is a complete list if link related parameters:

Parameter	Options	Description
disableCustomEmailLinks	0 = True 1 = False	This disables user's ability to type email addresses into the Mail To field in the Email Links tab in the Links window. Links provided from pathToEmailLinkList are still available.
disableCustomLinks	0 = True 1 = False	This disables user's ability to type hyperlink addresses into the Link field in the Hyperlinks tab in the Links window. Links provided from pathToLinkList are still available.
disableEmailLinks	0 = True 1 = False	This removes the user's ability to create Email Links. The Email Links tab is removed from the Links window. If this and disableLinks are both set to 1 then the Insert > Link menu item and the Link icon in the toolbar are removed.
disableLinks	0 = True 1 = False	This removes the user's ability to create Hyperlinks. The Hyperlinks tab is removed from the Links window. If this and disableEmailLinks are both set to 1 then the Insert > Link menu item and the Link icon in the toolbar are removed.
forceLinkNameAsLinkText	0 = True 1 = False	This forces the name of the link shown in the Hyperlinks and Email Links tab to be used as the text for the link. Will replace any selection with the link's name.
pathToLinkList	Path To File	Indicates a file used to provide a list of links to WysiPad. For more information see "Hyperlinks" below. When provided, the Link dropdown in the Links window is populated with a list of links.
pathToEmailLinkList	Path To File	Indicates a file used to provide a list of email links to WysiPad. For more information see "Email Links" below. When provided, the Mail To dropdown in the Links window is populated with a list of email links.

Hyperlinks

You can provide a list of link names and address to WysiPad by using a file referenced by the pathToLinkList parameter. The link list is used to populate the Link dropdown in the Links window.

The pathToLinkList document uses a simple WysiPad specific markup similar to HTML and XML to create the list. The following is an example link list:

```
<wpLinkList>
      <wpLink>
            <wpLinkName>Example Local Link</wpLinkName>
            <wpLinkAddress>/example.php?id=123</wpLinkAddress>
      </wpLink>
      <wpLink>
            <wpLinkName>Example External Link</wpLinkName>
            <wpLinkAddress>http://www.example.com</wpLinkAddress>
      </wpLink>
      <wpLink>
            <wpLinkName>CNN</wpLinkName>
            <wpLinkAddress>http://www.cnn.com</wpLinkAddress>
      </wpLink>
</wpLinkList>
```

This list denotes three links. Each link has an associated name, and address.

The link list has the following hierarchy of tags:

- wpLinkList
  - wpLink
    - wpLinkName
    - wpLinkAddress

Tag Name	Description
wpLinkList	The wpLinkList tag surrounds the entire list of links.
wpLink	Each link is denoted with a wpLink tag. This tag surrounds two tags which describe the link to WysiPad.
wpLinkName	Indicates the name of the link as it will be displayed in the Link field in the Links window.
wpLinkAddress	Indicates the address which will be used in <a> tags within your content.

You should not provide any additionally content inside or outside of these tags as this may cause WysiPad to malfunction.

For specifics on the pathToLinkList parameter see the "Full PathTo Parameters Details" section.

Email Links

Email links function almost identically to hyperlinks. You can provide a list of names and email addresses to WysiPad by using a file referenced by the pathToEmailLinkList parameter. The email link list is used to populate the Mail To dropdown in the Links window.

The pathToEmailLinkList document uses a simple WysiPad specific markup similar to HTML and XML to create the list. The following is an example email link list:

```
<wpEmailLinkList>
```

```
      <wpEmailLink>
            <wpEmailLinkName>Alagad Inc.</wpLinkName>
            <wpEmailLinkAddress>alagad@alagad.com</wpLinkAddress>
      </wpLink>
      <wpEmailLink>
            <wpEmailLinkName>John Smith, CEO</wpLinkName>
            <wpEmailLinkAddress>jsmith@bigcorp.com</wpLinkAddress>
      </wpLink>
      <wpEmailLink>
            <wpEmailLinkName>Your Senator</wpLinkName>
            <wpEmailLinkAddress>yoursenator@us.gov</wpLinkAddress>
      </wpLink>
</wpLinkList>
```

This list denotes three email addresses. Each address has an associated name, and email address.

The email link list has the following hierarchy of tags:

- wpEmailLinkList
  - wpEmailLink
    - wpEmailLinkName
    - wpEmailLinkAddress

Tag Name	Description
wpEmailLinkList	The wpEmailLinkList tag surrounds the entire list of email links.
wpEmailLink	Each email link is denoted with a wpEmailLink tag. This tag surrounds two tags which describe the email link to WysiPad.
wpEmailLinkName	Indicates the name of the email link as it will be displayed in the Mail To field in the Links window.
wpEmailLinkAddress	Indicates the email address which will be used in <a> tags within your content.

You should not provide any additionally content inside or outside of these tags as this may cause WysiPad to malfunction.

For specifics on the pathToEmailLinkList parameter see the "Full PathTo Parameters Details" section.

Saving to the Server

WysiPad supports the ability to save content directly to your server. This is accomplished by using the pathToSaveToServer parameter. This parameter points to a file which will receive content and save it. If you have provided the contentID parameter it will be posted along with the content. The pathToSaveToServer document should additionally return the contentID per the instructions below. The returned contentID will be used to set or reset the contentID parameter.

If you have provided the pathToSaveToServer and the disableSaveToServer parameter is not set to 1, then the File > Save To Server menu item and the Save To Server icon in the toolbar will be displayed.

The following is a complete list if link related parameters:

Parameter	Options	Description
contentID	String	This indicates an ID used by your application for this content. This is posted with the content to the pathToSaveToServer page. The pathToSaveToServer page also sets or updates this value.
disableSaveToServer	0 = True 1 = False	This disables the Save To Server feature and removes File > Save To Server menu item and the Save To Server icon in the toolbar. This is true even when the pathToSaveToServer parameter is provided.
pathToSaveToServer	Path To File	This indicates a file on your server which is used to save content from WysiPad 2 to your server. When provided, The File > Save To Server menu item and the Save To Server icon in the toolbar are displayed.
saveContentMethod	0 = HTML 1 = Text	This sets the type of content sent when saving to the server. HTML is the full HTML content. Text is the text only version of the content.

The pathToSaveToServer document receives the following form field:

Form Field Name	Description
wpContent	This is the current WysiPad content. This may be Text or HTML depending on the value of the saveContentMethod parameter.

The pathToSaveToServer document receives the following URL variable:

Form Field Name	Description
wpContentID	This is a unique ID for the content as provided in the contentID parameter. If the contentID parameter is not provided then this will be an empty string.

When the pathToSaveToServer document is called it should take the appropriate actions to save the content. To see a working example of a pathToSaveToServer document please refer to the example applications which were provided with WysiPad.

The pathToSaveToServer document returns the contentID. This is done with a simple WysiPad specific markup similar to HTML and XML. The following is an example returned value:

```
<wpContentID>41</wpContentID>
```

This list denotes the contentID.

The pathToSaveToServer returns only this tag:

- wpContentID

Tag Name	Description
wpContentID	The wpContentID tag surrounds the contentID.

You should not provide any additionally content inside or outside of these tags as this may cause WysiPad to malfunction.

For specifics on the pathToSaveToServer parameter see the "Full PathTo Parameters Details" section.

Keeping Server Sessions Alive

You may find that users will type directly into WysiPad 2 for lengths of time longer than the configured session expiration timeout on your website. In these circumstances when your users submit their form, or save their document they may be presented with a login prompt or an error message. When the user clicks their back button their content will probably be lost.

The solution to this problem is to use the pathToKeepAlive property. This property indicates a file on your server which when accessed will prevent the user's session from timing out.

The following is a complete list if link related parameters:

Parameter	Options	Description
keepAliveTimeout	Positive Integer	This is the amount of time in seconds between requests to the pathToKeepAlive document. The request to pathToKeepAlive repeats every time this timeout expires.
pathToKeepAlive	Path To File	This is the path to the file which will be called when the keepAliveTimeout expires.

Typically sessions on web based applications are tied into Cookies. You must pass these cookies through WysiPad or calls to this document will have no effect. The pathToKeepAlive document does not need to return any content to WysiPad 2.

For more information on sending cookies with WysiPad HTTP requests see the "Cookie and URL Parameters" section. Examples of working pathToKeepAlive documents written in ASP, ColdFusion and PHP are provided with the WysiPad .zip file.

Linking to Help Text

To aid you in providing support to your users you can link to any internet address from the Help > Contents… menu. Simply place the full URL to your help address in the helpContentsURL parameter. This is not a pathTo document and expects a fully qualified address.

Here's an example:

```
<object id="ctlWysiPad2"
      classid="CLSID:3FF35119-0152-4CDA-9C5F-6A9B4D35A184"
      codebase="/WysiPad2/WysiPad2.CAB#version=2,1,0,252"
      width="600" height="400">

      <!-- the domain and port parameters are required -->
      <param name="domain" value="www.example.com">
      <param name="port" value="80">

      <!-- This is the content I'm editing -->
      <param name="content"
            value="<p>This is my content.</p>">
```

```
    <!-- Turn on debug mode -->
    <param name="helpContentsURL"
          value="http://www.example.com/help.html">
</object>
```

Interface Parameters

WysiPad 2 provides many parameters for controlling which interface elements are available to users. The use of these parameters is outlined in the "Disabling and Enabling Interface Elements" section.

Parameter Name	Options	Menu	Toolbars	Windows	Desctiption
disableAbout	0 = HTML 1 = Text	Help > About WysiPad	None	None	Removes the About WysiPad feature. Alagad would appreciate if you did not remove this from the interface. Thanks!
disableAlignCenter	0 = HTML 1 = Text	Format > Alignment > Center	Center	None	Removes the Align Center feature.
disableAlignLeft	0 = HTML 1 = Text	Format > Alignment > Left	Left	None	Removes the Align Left feature.
disableAlignRight	0 = HTML 1 = Text	Format > Alignment > Right	Right	None	Removes the Align Right feature.
disableBlockformat	0 = HTML 1 = Text	Styles > Normal Styles > Heading 1 Styles > Heading 2 Styles > Heading 3 Styles > Heading 4 Styles > Heading 5 Styles > Heading 6 Styles > Address Styles > Formatted	Block Format Dropdown	None	Removes Block Formatting features.
disableBulletList	0 = HTML 1 = Text	Format > List > Bullet List	Bullet List	None	Removes the Bullet List feature.
disableColor	0 = HTML 1 = Text	Format > Color	Color	None	Removes the Set Font Color feature.
disableContents	0 = HTML 1 = Text	Help > Contents…	None	None	Removes the Help Contents feature. This is disabled automatically if the helpContentsURL parameter is not provided.
disableCopyHtml	0 = HTML 1 = Text	Edit > Copy HTML	None	None	Removes the Copy HTML feature.
disableCssStyles	0 = HTML 1 = Text	Styles > CSS Classes Styles > Clear Style	None	None	Removes Cascading Style Sheet related features. This is disabled automatically if the pathToStyleSheet parameter is not provided.
DisableCustomEmailLinks	0 = HTML 1 = Text	None	None	Links	Disables the ability to type email addresses into the Email Links tab in the Links window.
DisableCustomLinks	0 = HTML 1 = Text	None	None	Links	Disables the ability to type links into the Hyperlinks tab in the Links window.
disableDeleteImage	0 = HTML 1 = Text	None	None	Images	Removes the Delete Image feature. Specifically, this removes the Delete tab in the

					Images window. This is disabled automatically if the pathToImageDelete parameter is not provided.
disableEditImageWidthHeight	0 = HTML 1 = Text	None	None	Images Image Properties	Disables all Width and Height fields in the Image window and Image properties.
disableEditMenu	0 = HTML 1 = Text	Edit	None	None	Removes the Edit menu
disableEmailLinks	0 = HTML 1 = Text	Insert > Link	Link	Links	Removes the Insert Email Link feature. Specifically, this removes the Email Links tab in the Links window. If disableLinks is also set to 1 then all Link related features are removed.
disableFileMenu	0 = HTML 1 = Text	File	None	None	Removes the File menu.
disableFont	0 = HTML 1 = Text	Format > Font	Font Dropdown	None	The disableFont property indicates if the Font Menu will appear in the Format menu of if the Font menu appears in the Font Format toolbar.
disableFontSize	0 = HTML 1 = Text	Format > Font Size	Font Size Dropdown	None	The disableFontSize property indicates if the Font Size Menu will appear in the Format menu of if the Font Size menu appears in the Font Format toolbar.
disableFontToolbar	0 = HTML 1 = Text	None	Block Format Dropdown Font Dropdown Font Size Dropdown	None	Removes the Font toolbar items.
disableFormatingToolbar	0 = HTML 1 = Text	None	Bold Italic Underline Left Center Right Bullet List Number List Indent Outdent Color	None	Removes the Formatting toolbar items.
disableFormatMenu	0 = HTML 1 = Text	Format	None	None	Removes the Format Menu
disableHelpMenu	0 = HTML 1 = Text	Help	None	None	Removes the Help Menu
disableHorizontalRule	0 = HTML 1 = Text	Insert > Horizontal Rule	Horizontal Rule	None	Removes the Insert Horizontal Rule feature.
disableHtmlView	0 = HTML 1 = Text	View > WYSIWYG View > HTML	None	None	Removes the HTML view feature.
disableImages	0 = HTML 1 = Text	Insert > Image	Image	None	Removes the Insert Image feature. This is disabled automatically if the pathToImageList parameter is not provided.
disableInsertMenu	0 = HTML 1 = Text	Insert	None	None	Removes the Insert menu

disableLinks	0 = HTML 1 = Text	Insert > Link	Link	Links	Removes the Insert Link feature. Specifically, this removes the Hyperlinks tab in the Links window. If disableEmailLinks is also set to 1 then all Link related features are removed.
disableMenuBar	0 = HTML 1 = Text	All	None	None	Removes the entire Menu Bar.
disableNumberList	0 = HTML 1 = Text	Format > List > Number List	Number List	None	Removes the Number List feature.
disableObjectsToolbar	0 = HTML 1 = Text	None	Image Table Horizontal Rule Link	None	Removes the Objects toolbar items.
disableOpenLocal	0 = HTML 1 = Text	File > Open Local…	Open Local…	None	Removes the Open Local feature
disablePasteHtml	0 = HTML 1 = Text	Edit > Paste HTML	None	None	Removes the Paste HTML feature.
disablePasteUnformatted	0 = HTML 1 = Text	Edit > Paste Unformatted	None	None	Removes the Paste Unformatted feature.
disableRecentFiles	0 = HTML 1 = Text	File > [All Recent Files]	None	None	Removes the Recent Files feature.
disableRemoveFormat	0 = HTML 1 = Text	Styles > Remove Formatting	None	None	Removes the Remove Formatting feature.
disableSaveLocal	0 = HTML 1 = Text	File > Save Local File > Save Local As…	None	None	Removes the Save Local feature.
disableSaveToServer	0 = HTML 1 = Text	File > Save To Server	Save To Server	None	Removes the Save To Server feature. This is disabled automatically if the pathToSaveToServer parameter is not provided.
disableSpecialCharacters	0 = HTML 1 = Text	Insert > Special Characters	None	None	Removes Special Characters features.
disableSpellCheck	0 = HTML 1 = Text	Edit > Check Spelling	Check Spelling	None	Removes the Spell Check feature.
disableStandardToolbar	0 = HTML 1 = Text	None	Open Local… Save To Server Check Spelling Find Cut Copy Paste Undo Redo	None	Removes the Standard toolbar items.
disableStyleMenu	0 = HTML 1 = Text	Style	None	None	Removes the Style menu
disableTables	0 = HTML 1 = Text	Insert > Table	Table	None	Removes the Insert Table feature.
disableUploadImage	0 = HTML 1 = Text	None	None	Images	Removes the Upload Image feature. Specifically, this removes the Upload tab in the Images window. This is disabled automatically if the pathToImageUpload parameter is not provided.
disableViewMenu	0 = HTML 1 = Text	View	None	None	Removes the View menu
disableVisualAids	0 = HTML 1 = Text	View > Visual Aids	None	None	Removes the Visual Aids feature.

hideBold	0 = HTML 1 = Text	Styles > Bold	Bold	None	Hides the Bold feature. Does not disable the shortcut.
hideCopy	0 = HTML 1 = Text	Edit > Copy	Copy	None	Hides the Copy feature. Does not disable the shortcut.
hideCut	0 = HTML 1 = Text	Edit > Cut	Cut	None	Hides the Cut feature. Does not disable the shortcut.
hideFind	0 = HTML 1 = Text	Edit > Find	Find	None	Hides the Find feature. Does not disable the shortcut.
hideIndent	0 = HTML 1 = Text	Format > Indent	Indent	None	Hides the Indent feature. Does not disable the shortcut.
hideItalic	0 = HTML 1 = Text	Styles > Italic	Italic	None	Hides the Italic feature. Does not disable the shortcut.
hideOutdent	0 = HTML 1 = Text	Format > Outdent	Outdent	None	Hides the Outdent feature. Does not disable the shortcut.
hidePaste	0 = HTML 1 = Text	Edit > Paste	Paste	None	Hides the Paste feature. Does not disable the shortcut.
hideRedo	0 = HTML 1 = Text	Edit > Redo	Redo	None	Hides the Redo feature. Does not disable the shortcut.
hideSelectAll	0 = HTML 1 = Text	Edit > Select All	None	None	Hides the Select All feature. Does not disable the shortcut.
hideUnderline	0 = HTML 1 = Text	Styles > Underline	Underline	None	Hides the Underline feature. Does not disable the shortcut.
hideUndo	0 = HTML 1 = Text	Edit > Undo	Undo	None	Hides the Undo feature. Does not disable the shortcut.

Full PathTo Parameters Details

The following is a list of each pathTo parameter and the variable they receive when accessed and details of what they must return.

pathToContent

Variables Received:

Variable	Type	Description
All URL Variables	URL	URL variables as provided with URL parameters.
All Cookie Variables	Cookie	Cookies variables as provided with Cookie parameters.
wp_contentID	URL	The contentID as provided in the contentID parameter.

Content Returned:

Raw HTML content to edit.

pathToEmailLinkList

Variables Received:

Variable	Type	Description
All URL Variables	URL	URL variables as provided with URL parameters.

All Cookie Variables	Cookie	Cookies variables as provided with Cookie parameters.
wp_contentID	URL	The contentID as provided in the contentID parameter.

Content Returned:

Tag Name
<wpEmailList>
<wpEmail>
<wpEmailName>
<wpEmailAddress>

For more information see the “Email Links” section.

pathToImageDelete

Variables Received:

Variable	Type	Description
All URL Variables	URL	URL variables as provided with URL parameters.
All Cookie Variables	Cookie	Cookies variables as provided with Cookie parameters.
wp_contentID	URL	The contentID as provided in the contentID parameter.
wpImageName	Form	The image name
wpImageSize	Form	The image size
wpImageWidth	Form	The image width
wpImageHeight	Form	The image height
wpImagePath	Form	The image path

Content Returned:

Nothing

pathToImageList

Variables Received:

Variable	Type	Description
All URL Variables	URL	URL variables as provided with URL parameters.
All Cookie Variables	Cookie	Cookies variables as provided with Cookie parameters.
wp_contentID	URL	The contentID as provided in the contentID parameter.

Content Returned:

Tag Name
<wpImageList>
<wpImage>
<wpImageName>
<wpImageSize>

<wpImageWidth>	
<wpImageHeight>	

For more information see the "Using Images in WysiPad" section.

pathToImageUpload

Variables Received:

Variable	Type	Description
All URL Variables	URL	URL variables as provided with URL parameters.
All Cookie Variables	Cookie	Cookies variables as provided with Cookie parameters.
wp_contentID	URL	The contentID as provided in the contentID parameter.
wpImageName	Form	The image name
wpImageSize	Form	The image size
wpImageWidth	Form	The image width
wpImageHeight	Form	The image height
wpImageFile	Form	The image file.

Content Returned:

Nothing

pathToKeepAlive

Variables Received:

Variable	Type	Description
All URL Variables	URL	URL variables as provided with URL parameters.
All Cookie Variables	Cookie	Cookies variables as provided with Cookie parameters.
wp_contentID	URL	The contentID as provided in the contentID parameter.

Content Returned:

Nothing

pathToLinkList

Variables Received:

Variable	Type	Description
All URL Variables	URL	URL variables as provided with URL parameters.
All Cookie Variables	Cookie	Cookies variables as provided with Cookie parameters.
wp_contentID	URL	The contentID as provided in the contentID parameter.

Content Returned:

Tag Name
<wpLinkList>
<wpLink>
<wpLinkName>
<wpLinkAddress>

For more information see the "Hyperlinks" section.

pathToSaveToServer

Variables Received:

Variable	Type	Description
All URL Variables	URL	URL variables as provided with URL parameters.
All Cookie Variables	Cookie	Cookies variables as provided with Cookie parameters.
wp_contentID	URL	The contentID as provided in the contentID parameter.
wpContentID	Form	The content to save.

Content Returned:

Tag Name
<wpContentID>

For more information see the "Saving to the Server" section.

pathToStyleSheet

Variables Received:

Variable	Type	Description
All URL Variables	URL	URL variables as provided with URL parameters.
All Cookie Variables	Cookie	Cookies variables as provided with Cookie parameters.
wp_contentID	URL	The contentID as provided in the contentID parameter.

Content Returned:

Raw Cascading Style Sheet.