

# Table of Contents

<b>Part I Overview</b>	<b>4</b>
<b>Part II Website Hosting</b>	<b>4</b>
1 FTP Access .....	4
2 Web Access .....	4
3 Your Domain Name .....	5
4 Email .....	5
<b>Part III Dynamic Components</b>	<b>6</b>
1 Include .....	7
2 Root .....	7
3 QRCBookSingleRoomWizard .....	7
Google Conversion Tracking .....	9
4 LinearAvailabilityGrid .....	11
5 FormToEmail .....	11
6 WebForm .....	13
7 Random Content Rotator .....	13
InitialiseRandomContent .....	13
DisplayRandomContent .....	14
8 Special Access Components .....	14
Complete Example .....	14
LoggedIn .....	15
NotLoggedIn .....	16
User Access Groups .....	16
HasPermission.....	16
NoPermission.....	17
9 Directory Hosting Components .....	17
Data Fields .....	19
Article .....	19
Listing .....	20
Area .....	21
POI .....	21
Town .....	21
Conditional Tag Pairs .....	22
IfViewsSet.....	22
IfAreaSelected.....	23
IfLocated .....	23
IfFormSubmitted.....	23
IfGeoSearchResults .....	24
IfArticleSearchResults.....	24
IfListingSearchResults.....	25
IfCategorySearchResults.....	25
IfCookieExists.....	26
IfListing .....	27

IfRequestParam.....	28
IfImage .....	28
IfFileExists.....	29
<b>Cookies .....</b>	<b>29</b>
GetCookie .....	30
<b>Searches .....</b>	<b>30</b>
GeoSearch.....	30
PerformGeoSearch.....	30
PerformGeoSearch2.....	31
GeoSearchResults .....	31
Article Search.....	32
PerformArticleSearch.....	33
ArticleSearchResults.....	34
ArticleSearchResultsNavbar.....	34
Listing Search.....	34
PerformListingSearch.....	35
ListingSearchResults.....	38
ListingSearchResultsNavbar.....	38
LinearAvailabilityGrid.....	39
<b>Set and Get .....</b>	<b>40</b>
Set .....	41
Get .....	41
Inc .....	41
IfSet .....	41
IfNotSet .....	42
<b>BreadCrumb Navigation Bar .....</b>	<b>42</b>
AddBreadCrumb.....	43
AddGeoBreadCrumbs.....	44
BreadCrumbNavBar.....	44
<b>Miscellaneous Utility Tags .....</b>	<b>44</b>
FeatureList.....	45
ListingTypeList.....	45
Random .....	46
Currency .....	46
TownList .....	47
AreaList .....	47
SelectedLocation.....	48
RegionID.....	48
RegionOrTownPhoto.....	49
<b>Relaxas Only Tags .....</b>	<b>49</b>
GoogleAPIKey.....	49
<b>SearchIndex .....</b>	<b>50</b>
<b>PageIndex .....</b>	<b>50</b>
<b>AddStyleSheet .....</b>	<b>50</b>
<b>RedirectTo .....</b>	<b>51</b>
<b>SiteURL .....</b>	<b>51</b>
<b>SecureURL .....</b>	<b>51</b>
<b>SetView .....</b>	<b>51</b>
<b>SetArticle .....</b>	<b>53</b>
<b>SetBoundaries .....</b>	<b>53</b>
<b>Map .....</b>	<b>53</b>
<b>GoogleMap .....</b>	<b>55</b>
<b>NavProfile .....</b>	<b>55</b>
<b>Title Keywords and Description .....</b>	<b>55</b>

RequestParam .....	56
Image .....	56
ImageLabel .....	56
<b>10 Notes .....</b>	<b>57</b>
Date Formats .....	58
<b>Index</b>	<b>0</b>

## 1 Overview

**QRes Connect** is an online service which provides all the tools you need for real-time availability and bookings on your website.

## 2 Website Hosting

Every QRes Connect subscription comes with full website hosting. You may choose to host your entire website with us, or just your actual booking wizard page. Whichever you choose, the administration tools are the same.

If you have a trusted website designer who manages your website for you, they will require the following information as well as your SiteID and password.

### 2.1 FTP Access

Your website is stored as a series of files on the QRes Connect server. You can gain access to these files by using the industry standard FTP protocol.

The FTP Server Address is: **ftp.qres.net**

To log on, you will need to use the SiteID and password issued to you with your QRes Connect subscription.

Many different programs which provide FTP access, these are often referred to as "FTP clients". Here's a couple of them:

[Smart FTP](#) (free)  
[FileZilla](#) (free)

#### **Windows Explorer:**

You can also use the Windows Explorer. To set it up, do the following:

1. Go to the Start menu, and select Run
2. Type "explorer" and press enter
3. Go to the Tools menu, select Folder Options
4. Click the View tab
5. Check "Display the full path in the address bar" and click OK
6. In the address bar, type in "**ftp://YourSiteID@ftp.qres.net/**" and press enter. (change **YourSiteID** to your actual SiteID).
7. Enter your password, and you will see your files. You can add/delete/copy files just as you normally do in Windows Explorer.

### 2.2 Web Access

We also provide access to your website's files via a browser interface.

Go to <http://qres.net> and log to the customer tools using your SiteID and password.

Follow the link to the Web Based File Manager and log in.

## 2.3 Your Domain Name

If you have a domain name such as **myhotel.com**, you can arrange to point it to your new website.

### Finding your registrar's website

Every domain name is registered through a **registrar**. You may or may not know who your domain name registrar is.

If you're not sure who your domain name registrar is:

1. Go to <http://domains.qres.net>
2. Type your domain name into the search box and press **Go!**
3. Click the **Whois** link next to your domain name

In the readout, look for a "Registrar" entry. This is your registrar.

Also look for a "Referral URL". This is your registrar's website, which is where you must go to manage your domain name.

### Updating your DNS Details

To make your domain name point to your new website, you must update the "DNS Details" or "Delegation" on your registrar's website.

The information you must change is:

Primary Name Server: **ns1.qres.net**

Secondary Name Server: **ns1.qres.net**

Once you have done this, then website users will start to be directed to your website on QRes Connect instead of its old location. It can take up to several days for the DNS servers on the internet to "forget" your old website's location and to start using its new location.

### IMPORTANT

Once you have changed delegation, you must let us know (send an email to [info@qres.net](mailto:info@qres.net)) so we can enable it on the QRes Connect server.

## 2.4 Email

If you delegate your domain name to QRes Connect, then all email for your domain name will also come through the QRes Connect mail server.

We provide complete email service which includes an unlimited number of mailboxes.

You may choose to simply redirect all email to your existing ISP email account, or alternatively, set up mailboxes on the QRes Connect mail server and access them directly.

To administer these settings, log in at <http://qres.net> and follow the instructions under "Your Mail Server". This will only be visible once you have delegated your domain name to us and let us know about it.

## 3 Dynamic Components

The QRes Connect hosting system looks and behaves exactly like any other website hosting service.

But in addition to this, we provide **dynamic components** which enhance the functionality of your website.

Components can be embedded into any .htm or .html page simply by adding its **tag**. The most common is the QRCBookSingleRoomWizard tag, here is an example HTML page to show how it works:

```
<html>
  <head>
    <title>The Lazy Days Resort</title>
  </head>
  <body>
    <h1>Our Booking Form:</h1>
    [ *QRCBookSingleRoomWizard* ]
  </body>
</html>
```

Every time this page is requested by a website user, the tag is replaced with the actual booking wizard. The booking wizard is just 1 example of a dynamic component.

Dynamic component tags are specific keywords enclosed in [**\*** and **\***] brackets.

### Parameters

Some components have optional parameters which change their behaviour and/or appearance, and some also have corresponding CSS (style sheets) which also change their appearance.

Parameters are added inside the tags, like in this example which tells the booking wizard component to display 15 days in the availability calendar, instead of the default of 12:

```
[ * QRCBookSingleRoomWizard ShowAvailabilityDays=15 * ]
```

### Stylesheets

Some components automatically insert a link in your document's HEAD section to a particular stylesheet.

These stylesheets are located in `/global/css/` and usually have the same name as the component, with a ".css" extension.

If your website doesn't contain the particular CSS file referenced in its `/global/css/` directory, then a default one is automatically returned by the web server (usually with boring grey colours and fonts). You can get access to these default CSS file by pulling up `http://yoursiteid.qrconnect.net/global/css/componentname.css` in a web browser. You can edit these file and place a copy of them into your own site's `/global/css/` directory to control the appearance of these components.

Alternatively, you can use the `CustomStyle` parameter of these components to suppress the reference to the stylesheet altogether. You can then use your own stylesheet(s) to customise the components' appearance.

## 3.1 Include

This tags allows you to include the contents of another HTML file.

Example:

```
[*include file="header.htm" *]  
<li>Here is some page content</li>  
[*include file="footer.htm" *]
```

[IgnoreIfNotFound](#) (optional)

Add this parameter if you don't want an error message displayed if the file cannot be found.

Example:

```
[* Include file="banner.htm" IgnoreIfNotFound=true *]
```

## 3.2 Root

This tag maps a url to the root directory of the website.

It makes absolute URL's safe even for secure pages, where the base URL is **qconnect.net/<SiteID>/** rather than **www.yourdomain.com/**

Example:

```
<link rel="shortcut icon" href="[*Root*]/favicon.ico" type="image/x-icon" />
```

## 3.3 QRCBookSingleRoomWizard

This is the main accommodation booking wizard component.

It displays availabilities and allows bookings which are sent directly to QRes or the QRes Messenger on your computer if you're online, or queues bookings for when you next connect to the internet.

It uses a style sheet called [/global/css/qrcbooksingleroomwizard.css](#) to control it's appearance, however you can override this.

### Parameters

All parameters are optional. Multiple parameters can be included in the tag

#### [Jump to Label](#)

This tag allows customisation of the Calendar link

```
[* QRCBookSingleRoomWizard JumpToLabel="Click HERE to select your arrival date" *]
```

#### [Make a Booking Label](#)

This tag allows customisation of the make a Booking Label

```
[* QRCBookSingleRoomWizard MakeABookingLabel="Calculate your BEST DEAL rate" *]
```

### Make a Booking Secondary Label

This tag allows an additional line of text to be added on the booking page

```
[ * QRCBookSingleRoomWizard MakeABookingSecondaryLabel="Enter your Email address to confirm that you have read and agree to the above terms & conditions" * ]
```

### Send copy of Email notification

This tag sends a copy of the hotel email to the named mailbox

```
Example: [ * QRCBookSingleRoomWizard CopyEmail="info@qres.net" * ]
```

### Send copy of Email notification

This tag sends a copy of the hotel email to the named mailbox

```
Example: [ * QRCBookSingleRoomWizard CopyEmail="info@qres.net" * ]
```

### Daily rather than Nightly Charge

This tag turns "Night" into "Day"

```
Example: [ * QRCBookSingleRoomWizard DailyCharge=True * ]
```

### Hide Options Tab

This tag suppresses the options tab if there are no options in the rules

```
Example: [ * QRCBookSingleRoomWizard OptionsVisible=False * ]
```

### DisplayTariffs

This is the text at the top-right above the availability display. By default it displays "Nightly rates in <your currency>".

```
Example: [ * QRCBookSingleRoomWizard DisplayTariffs="False" * ]
```

### NightlyRatesLabel

This is the text at the top-right above the availability display. By default it displays "Nightly rates in <your currency>".

```
Example: [ * QRCBookSingleRoomWizard NightlyRatesLabel="Weekly Rates" * ]
```

### MaxNights

This controls the maximum number of nights which can be selected in the Nights drop-down list. The default value is 30.

```
Example: [ * QRCBookSingleRoomWizard MaxNights=60 * ]
```

### DefaultNights

This controls the default selection for the number of nights in the Nights drop-down list. The default value is 1.

```
Example: [ * QRCBookSingleRoomWizard DefaultNights=7 * ]
```

### CustomStyle

Set this to any value to prevent a link to the default CSS file from being inserted into the document's HEAD section automatically. See the StyleSheet section below for more details.

```
Example: [ * QRCBookSingleRoomWizard CustomStyle=True * ]
```

### ShowAvailabilityDays

This specifies the number of days shown in the availability matrix. The default is 12.

```
Example: [ * QRCBookSingleRoomWizard ShowAvailabilityDays=14 * ]
```

### BookingSourceID

This specifies an alternate Booking Source ID. The default is the Site ID.

```
Example: [ * QRCBookSingleRoomWizard BookingSourceID="Agents" * ]
```

### Debug

If you give any value to this parameter, you will see a debugging log for autoquote calculations. This information includes which functions are called, and any logging you add using the "Log" function in the autoquote code.

Example: `[* QRCBookSingleRoomWizard Debug=1 *]`

### ShowTariffForDays

Set this value to the number of nights you wish to show tariffs for. You must have tariff ranges entered for this number of nights, otherwise it will show as blank.

Example: `[* QRCBookSingleRoomWizard ShowTariffForDays=2 *]`

### RoundTariffs

Set this value to True if you wish to round the displayed tariffs to the nearest dollar.

Example: `[* QRCBookSingleRoomWizard RoundTariffs=True *]`

### AcknowledgementText

This is a string which is included in the acknowledgement which is sent to the guest after the finish the booking wizard.

If left blank, the text which appears is:

`Thankyou for requesting a booking at [Establishment Name], a confirmation will be sent shortly. The details of your request are as follows:`

## Stylesheet

By default, a link to the file `/global/css/qrcbooksingleroomwizard.css` is added to the head section of your HTML document if you use this component. If your website doesn't contain this CSS file, then a default one is returned by the web server. You can get access to the default CSS file by pulling up `http://yoursiteid.qrconnect.net/global/css/qrcbooksingleroomwizard.css` in a web browser.

You can edit this file and place it into the `/global/css/` directory of your website to control the appearance of your booking wizard.

Alternatively, you can use the `CustomStyle` parameter to suppress the reference to `/global/css/qrcbooksingleroomwizard.css`, and add your own stylesheet reference in the HEAD section of the document.

### 3.3.1 Google Conversion Tracking

The QRCBookSingleRoomWizard tag supports basic Google Conversion Tracking.

To make this work, simply insert the tracking code into a file called **BasicGoogleConversionTracking.inc** in the `/global/` directory of your website.

The basic Google Conversion Tracking code looks like this:

```
<!-- Google Code for lead Conversion Page -->
<script language="JavaScript" type="text/javascript">
<!--
var google_conversion_id = 1067251425;
var google_conversion_language = "en_GB";
var google_conversion_format = "1";
var google_conversion_color = "FFFFFF";
if (1) {
  var google_conversion_value = 1;
}
```

```
var google_conversion_label = "lead";  
//-->  
</script>  
  
<script language="JavaScript"  
src="https://www.googleadservices.com/pagead/conversion.js">  
</script>  
<noscript>  
  
</noscript>
```

## 3.4 LinearAvailabilityGrid

This tag displays an availability/pricing grid.

It uses a style sheet called `/global/css/linearavailabilitygrid.css` to control its appearance, however you can override this.

### Parameters

#### CustomStyle

Set this to any value to prevent the link to the default CSS file from being inserted into the document's HEAD section.

Example: `[* LinearAvailabilityGrid CustomStyle=True *]`

#### ShowAvailabilityDays

This specifies the number of days shown in the availability matrix. The default is 12.

Example: `[* LinearAvailabilityGrid ShowAvailabilityDays=14 *]`

#### TopLeftLabel

The text which appears in the top-left cell of the grid.

Example: `[* LinearAvailabilityGrid TopLeftLabel="Properties" *]`

#### MonthFormat

[Date format](#) of the month row text

Example: `[* LinearAvailabilityGrid MonthFormat="dddd" *]`

#### DayFormat

[Date format](#) of the day cell text

Example: `[* LinearAvailabilityGrid TopLeftLabel="Properties" *]`

#### CellFormat

Numeric format of the day cell text

Example: `[* LinearAvailabilityGrid CellFormat="$.2f" *]`

#### ListingLink

URL of listing pages. The component adds the `ql` parameter for the selected listing.

Example: `[* LinearAvailabilityGrid ListingLink="listing.htm" *]`

#### TruncRoomName

Truncate the length of room names to this number of characters.

Example: `[* LinearAvailabilityGrid TruncRoomName=20 *]`

## 3.5 FormToEmail

**This tag is no longer supported, and will be removed in the future.**

**Use [WebForm](#) instead.**

This component allows you to make any HTML form you wish, and have it's results emailed to you when the user submits it.

To use it, make any HTML form you wish, and use the `[*FormToEmail*]` tag in place of the standard HTML `<form>` tag.

Example:

```
[*FormToEmail redirectTo="index.html" *]
<table>
  <tr valign="top"><td>Your name: </td><td><input name="from"></td></tr>
  <tr valign="top"><td>Email address: </td><td><input name="email"></td></tr>
  <tr valign="top"><td>Message:</td><td><TEXTAREA
NAME="message"></td></tr>
  <tr valign="top">
    <td> </td>
    <td><input type="submit" value="submit" columns=50></td>
  </tr>
</table>
</form>
```

### Parameters:

#### [RedirectTo](#)

This is the page you wish to redirect the user to after they have submitted the form.

Example: `[* FormToEmail RedirectTo="thankyou.htm" *]`

## 3.6 WebForm

This set of components allows you to make any HTML form you wish, and have it's results emailed to you when the user submits it.

The example below shows how to use it.

All tags are required.

A hidden URL parameter called "sub" is required.

Example:

**[\*WebForm\*]**

**[\*WebFormInput\*]**

```
<form method="post">
  <input type="hidden" name="sub" value="1">
  <table>
    <tr valign="top"><td>Your name: </td><td><input name="from"></td></tr>
    <tr valign="top"><td>Email address: </td><td><input name="email"></td></tr>
    <tr valign="top"><td>Message:</td><td><TEXTAREA
NAME="message"></td></tr>
    <tr valign="top">
      <td> </td>
      <td><input type="submit" value="submit" columns=50></td>
    </tr>
  </table>
</form>
```

**[\*/WebFormInput\*]**

**[\*WebFormSubmitted\*]**

```
<li>Thanks for entering your message!
```

**[\*/WebFormSubmitted\*]**

### Parameters:

None.

## 3.7 Random Content Rotator

The content rotator allows you to display random content, for example rotating banner ads.

It requires 2 tags. The first tag, InitialiseRandomContent, tells the system what content to insert.

The second tag, DisplayRandomContent displays the random content. It may be used more than once if required to render multiple different elements.

### 3.7.1 InitialiseRandomContent

This **non-visible** tag initialises the random content generator.

It randomises the order of the different elements which you specify. Later, you can use the DisplayRandomContent tag to display one or more of these elements.

Each element is a string separated by either a space or a new line, as in the following example.

Example:

```
[ * InitialiseRandomContent
  "<a href=" "http://www.wikipedia.org" ">Wikipedia</a>"
  "<a href=" "http://www.google.org" "></a>"
  "<a href=" "http://www.yahoo.com" ">Yahoo</a>"
* ]
```

NOTES:

- The entire string for each entry must be contained within double-quotes.
- If the content to be displayed contains any double-quote marks (") then use double-double-quote marks (") as in the above example.

Weight

You can specify different weightings for each element, by adding a number after the Pipe key (|) as in the following example.

Example:

```
[ * InitialiseRandomContent
  "<a href=" "http://www.wikipedia.org" ">Wikipedia</a>" | 25
  "<a href=" "http://www.google.org" "></a>" | 50
  "<a href=" "http://www.yahoo.com" ">Yahoo</a>" | 5
* ]
```

In this example, Google is twice as likely to appear first as Wikipedia. And Wikipedia will, on average, appear first 5 times more frequently than Yahoo.

### 3.7.2 DisplayRandomContent

This tag displays an element from the random content list.

Before using this tag, use the DisplayRandomContent tag to specify a list of elements. These elements are sorted randomly (according to the weight of each which you can also specify)

It randomises the order of the different elements which you specify. Later, you can use the tag to display one or more of these elements.

**Example**

```
[ * DisplayRandomContent * ]
```

## 3.8 Special Access Components

There are a range of components which allow restricted access to various pages of your website.

### 3.8.1 Complete Example

Below is a comprehensive example of how you can grant access to certain pages of your website, to authorized users only.

For this example, only users who you have added to your "Agents" group can make bookings. You can pre-authorise specific email addresses for access.

See the following pages for more detailed information about the tags.

```

<html>
<h1>Agent's Booking Page</h1>

[*Notloggedin*]
  <!-- This section appears if the user is not logged in -->
  [*LoginSignup*]  <!-- This is a login & signup form for users -->

[*/Notloggedin*]

[*loggedin*]
  <!-- This section appears if the user is logged in -->

  [*HasPermission Agents*]
  <!-- This section appears if the user has access to the "Agents" group -->
  Greetings, Agent!<br>

  [*Validated*]
  <!-- This section appears if the user has validated their email address -->
  [*QRCBookSingleRoomWizard BookingSourceID="Something"*]
  [*/Validated*]

  [*NotValidated*]
  <!-- This section appears if the user has not validated their email
address -->
  You haven't validated your email address. <br>
  Upon signing up with this website, you should have received an email
containing a password.<br>
  Please enter that password below to continue.<br>

  [*LoginSignup*]  <!-- This asks for the password -->

  [*/NotValidated*]

[*/HasPermission*]

  [*HasNoPermission Agents*]
  <!-- This section appears if the user doesn't have access to the "Agents"
group -->
  <P>You don't have access. Please contact blah@blah.com if you are an agent.
  [*/HasNoPermission*]

  <!-- This link logs the user out -->
  <br><a href="?u_logout=1">log out</a>

[*/Loggedin*]

</html>

```

### 3.8.2 LoggedIn

This pair of tags allows you to specify blocks of HTML (including other components) which only appear if the user is logged in.

Example:

```

<html>
<h1>My page</h1>

[*LoggedIn*]
<li>This text appears if the user is logged in only.</li>

```

```
[*/LoggedIn*]  
</html>
```

### 3.8.3 NotLoggedIn

This pair of tags allows you to specify blocks of HTML (including other components) which only appear if the user is not logged in.

Example:

```
<html>  
<h1>My page</h1>  
  
[*/NotLoggedIn*]  
<li>This text appears if the user is not logged in.</li>  
[*/NotLoggedIn*]  
  
</html>
```

### 3.8.4 User Access Groups

A user access group is identified by a Group ID of your choice.  
Group ID's must be less than 16 characters long and must not contain spaces.  
They are case sensitive.

To give website users access to a group:

1. log in at [qres.net](http://qres.net) using your SiteID and password.
2. Click the "Manage access groups" link

Add permissions for your selected Group ID and email addresses as required.

You may enter a \* as a wildcard for the email address, for example:

```
*@qres.net
```

would give access to the group for any user whose email address ends in @qres.net

To have permission to a group, a user must have logged in using their password at least once.  
This ensures that the specified email address belongs to them.

#### 3.8.4.1 HasPermission

This pair of tags allows you to specify blocks of HTML (including other components) which only appear if the user is logged in and has permission to the specified group.  
If no group is specified, then the "Default" GroupID is assumed.

Example:

```
<html>  
<h1>My page</h1>  
  
[*/HasPermission*]  
<li>This text appears if the user is logged in and has access to the "Default"  
permission GroupID.</li>
```

```
[*/HasPermission*]

[*HasPermission Agents*]
<li>This text appears if the user is logged in and has access to the "Agents"
permission GroupID.</li>
[/HasPermission*]

</html>
```

### 3.8.4.2 NoPermission

This pair of tags allows you to specify blocks of HTML (including other components) which appear if the user does not have permission to the specified group.

If no group is specified, then the "Default" GroupID is assumed.

Example:

```
<html>
<h1>My page</h1>

[*NoPermission*]
<li>You do not have access to the "Default" group.</li>
[/NoPermission*]

[*NoPermission Agents*]
<li>You do not have access to the "Agents" group.</li>
[/NoPermission*]

</html>
```

## 3.9 Directory Hosting Components

There are a range of content management components suitable for "Directory" style websites including:

- Corporate/chain websites
- Secure intranet booking systems for centralised bookings or book-ahead services
- Regional tourism directories such as [www.clarencecoast.info](http://www.clarencecoast.info)
- Global accommodation directories such as [www.wheretostay.net](http://www.wheretostay.net)

### Geo Referenced pages

Each page of your website can be geo-referenced, which means that it has geographical area (defined by latitude and longitude) associated with it.

Geo-referencing is automatic when you load a listing, an article, a region or a town.

You can also use the SetView tag to explicitly geo-reference a page.

Any geo-referenced page can optionally display a map using the Map tag.

### Standard URL Parameters

Certain URL parameters are "built in" to the hosting system and are added/used/required by a number of components.

These can be used as either GET or POST http variables.

Values are case-sensitive

ql	Listing ID
qt	Town string
qp	Point Of Interest ID
qr	Area ID (Countries, State/Counties, Tourism Regions)
qa	Article ID
qv	Encoded navigation string
srf	Search Results "From" (default is 1)
sub	Indicates to various components that a form was submitted.
sc	Listing Class (in a listing search)
df	Date from (in an availability display) - format must be dd_mm_yyyy
sk	Search Keywords
st	Listing Category Type (in a listing search)
tl	Tariff Lower (for accommodation/price searches)
lu	Tariff Upper (for accommodation/price searches)

## 3.9.1 Data Fields

### 3.9.1.1 Article

This tag adds a field from the current article.

Example:

```
[* Article field=Title *]  
[* Article format="<B>Date:</B> %s <br/>" field=FeatureDate dateformat="d mmmm yyyy"  
*]
```

## Parameters

### Field

This defines the field which you wish to display. Available fields are:

```
ArticleID  
Title  
Location  
RealLatitude  
RealLongitude  
IdealZoom  
BodyText  
SynopsisText  
Keywords  
GroupID  
FeatureDate (date)  
ValidFrom (date)  
ValidUntil (date)
```

### Format (optional)

This is the output format, including any prepended or appended text.

The advantage of using this for captions etc, is that if the field itself is blank or null, then the caption is also suppressed.

Use the token %s to specify where the field value appears in the format string.

### DateFormat (optional)

This defines the format for displaying dates. Default is "**dd mmm yyyy**" (example: "18 Jun 2006")

### MaxLength (optional)

Cuts the value off at a maximum number of characters (to the nearest word).

### StripTags (optional)

Set this to any value to remove HTML from the text.

### 3.9.1.2 Listing

This tag adds a field or an image link for the current listing.

Example:

```
[* Listing field=ListingText *]
[* Listing format="<B>Phone:</B> %s <br>" field=Phone *]
[* Listing field="Categories" Delimiter="<br>" *]
```

## Parameters

### Field

This defines the field which you wish to display. Available fields are:

```
ListingID
ListingText
Directions
LastUpdate (date)
EDFClassID
EDFClass
Name
CurrencyID
CurrencySign
Address
Town
TownAsString
StateID
State
CountryID
Country
Postcode
WebsiteAddress
Email
Phone
Fax
Mobile
Proprietor
RealLatitude
RealLongitude
Categories (list with optional delimiter)
ThirdPartyBookingSystemURL
```

### Format (optional)

This is the output format, including any prepended or appended text.

The advantage of using this for captions etc, is that if the field itself is blank or null, then the caption is also suppressed.

### DateFormat (optional)

This defines the format for displaying dates. Default is "dd mmm yyyy" (example: "18 Jun 2006")

### Delimiter (optional)

This defines the delimiter for a list of items. The default is a comma.

### Image (optional)

This outputs an image link (full <img> tag) to a listing image.

### MaxLength (optional)

Cuts the value off at a maximum number of characters (to the nearest word).

### StripTags (optional)

Set this to any value to remove HTML from the text.

### 3.9.1.3 Area

This tag adds a field or an image link for the current region.

Example:

```
[* Area field=Heading *]
```

#### Parameters

##### Field

This defines the field which you wish to display. Available fields are:

```
AreaID  
Heading
```

##### Format (optional)

This is the output format, including any prepended or appended text.

The advantage of using this for captions etc, is that if the field itself is blank or null, then the caption is also suppressed.

##### MaxLength (optional)

Cuts the value off at a maximum number of characters (to the nearest word).

##### StripTags (optional)

Set this to any value to remove HTML from the text.

### 3.9.1.4 POI

This tag adds a field or an image link for the current region.

Example:

```
[* POI field=Heading *]
```

#### Parameters

##### Field

This defines the field which you wish to display. Available fields are:

```
POIID  
Heading  
Name
```

##### Format (optional)

This is the output format, including any prepended or appended text.

The advantage of using this for captions etc, is that if the field itself is blank or null, then the caption is also suppressed.

##### MaxLength (optional)

Cuts the value off at a maximum number of characters (to the nearest word).

##### StripTags (optional)

Set this to any value to remove HTML from the text.

### 3.9.1.5 Town

This tag adds a field or an image link for the current town.

Example:

```
[* Town field=Name *]
```

#### Parameters

### Field

This defines the field which you wish to display. Available fields are:

```
Name  
StateID  
State  
CountryID  
Country  
Postcode  
AsString  
RealLatitude  
RealLongitude  
IdealZoom  
NameWithClarifiers - includes state/county and country only if different from the current location.
```

### Format (optional)

This is the output format, including any prepended or appended text.

The advantage of using this for captions etc, is that if the field itself is blank or null, then the caption is also suppressed.

## 3.9.2 Conditional Tag Pairs

### 3.9.2.1 IfViewIsSet

This tags outputs text conditionally depending on whether a location has been determined.

Example:

```
[* IfViewIsSet *]  
Some text which only appears if the page has a geographical context.  
[* /IfViewIsSet *]
```

### Parameters

#### Not

Specifies that text is NOT shown if the condition is met.

Example:

```
[* IfViewIsSet Not *]  
Some text which doesn't appear if a country, state/county, region or town has been selected by the user  
[* /IfViewIsSet *]
```

### 3.9.2.2 IfAreaSelected

This tags outputs text conditionally depending on whether an area (country, state or tourism region) has been specifically selected by the user.

Example:

```
[* IfAreaSelected *]
```

Some text which only appears if a country, state/county, region or town has been actively selected by the user

```
[* /IfAreaSelected *]
```

#### Parameters

**Not**

Specifies that text is NOT shown if the condition is met.

Example:

```
[* IfAreaSelected Not *]
```

Some text which doesn't appear if a country, state/county, region or town has been selected by the user

```
[* /IfAreaSelected *]
```

### 3.9.2.3 IfLocated

This tags outputs text conditionally depending on whether a location (country, state/county, or region) has been determined.

Example:

```
[* IfLocated *]
```

Some text which only appears if a country, state/county, region or town has been located.

```
[* /IfLocated *]
```

#### Parameters

**Country**

**State**

**TouristRegion**

**Town**

Example:

```
[* IfLocated Town *]
```

Some text which only appears if a town/country/region/state has been determined.<br>

This could be because a listing was selected, or the user has selected or searched for a town.

```
[* /IfLocated *]
```

**Not**

Specifies that text is NOT shown if the condition is met.

Example:

```
[* IfLocated Not *]
```

Some text which doesn't appear if a country, state/county, region or town has been selected by the user

```
[* /IfLocated *]
```

### 3.9.2.4 IfFormSubmitted

This tags outputs text conditionally depending on whether a location has been detected.

NOTE: The form must include a parameter called "id".

**Also available:** [IfFormSubmitted](#)

Example:

```
[* IfFormSubmitted *]  
Some text which only appears if a form was submitted by the user.<br>  
Forms must include a hidden parameter called "id" with a non-blank value for this to work.  
[* /IfFormSubmitted *]
```

**Not**

Specifies that text is NOT shown if the condition is met.

Example:

```
[* IfFormSubmitted Not *]  
Some text which DOES NOT appear if a form was submitted  
[* /IfFormSubmitted *]
```

**ID**

Tests for a specific form. If a form with a different ID was submitted, it is ignored.

Example:

```
[* IfFormSubmitted ID=searchform *]  
Some text only appears if a form was submitted with a hidden ID parameter of "searchform"  
[* /IfFormSubmitted *]
```

### 3.9.2.5 IfGeoSearchResults

This tags outputs text conditionally depending on the number of Geo search results.

Example:

```
[* IfGeoSearchResults over=1 *]  
Some text which only appears if there is at least 1 geo search result.  
[* /IfGeoSearchResults *]
```

#### Parameters

**Over**

Set this to specify the number minimum number (minus 1) of search results required for text to be shown.

**Under**

Set this to specify the number maximum number of search results required for text to be shown.

### 3.9.2.6 IfArticleSearchResults

This tags outputs text conditionally depending on the number of Article search results.

Example:

```
[* IfArticleSearchResults over=1 *]  
Some text which only appears if there is at least 1 article search result.  
[* /IfArticleSearchResults *]
```

#### Parameters

**Over**

Set this to specify the number minimum number (minus 1) of search results required for text to be shown.

#### Under

Set this to specify the number maximum number of search results required for text to be shown.

### 3.9.2.7 IfListingSearchResults

This tags outputs text conditionally depending on the number of Listing search results.

#### Example:

```
[* IfListingSearchResults over=1 *]  
Some text which only appears if there is at least 1 listing search result.  
[* /IfListingSearchResults *]
```

### Parameters

#### Over

Set this to specify the number minimum number (minus 1) of search results required for text to be shown.

#### Under

Set this to specify the number maximum number of search results required for text to be shown.

### 3.9.2.8 IfCategorySearchResults

This tags outputs text conditionally depending on the number of categories matched in keyword search results.

#### Example:

```
[* IfCategorySearchResults over=1 *]  
Some text which only appears if there is at least 1 listing search result.  
[* /IfCategorySearchResults *]
```

### Parameters

#### Over

Set this to specify the number minimum number (minus 1) of categories matched required for text to be shown.

#### Under

Set this to specify the number maximum number of categories matched for text to be shown.

### 3.9.2.9 IfCookieExists

This tags outputs text if a cookie of the given name exists.

Example:

```
[* IfCookieExists name="MyCookie" *]  
Some text which only appears if MyCookie has been set to something  
[* /IfCookieExists *]
```

#### Parameters

##### Name

The cookie name which you want to test is required.

##### Not

Specifies that text is NOT shown if the condition is met.

Example:

```
[* IfCookieExists name="MyCookie" Not *]  
Some text that DOES NOT appear if the named cookie exists  
[* /IfCookieExists *]
```

### 3.9.2.10 IfListing

This tag outputs text conditionally depending on data for the current listing.

Example:

```
[* IfListing Field="Phone" *]  
Some text which only appears if the current listing has a phone number.  
[* /IfListing *]
```

## Parameters

### AcceptsQResBookings

Example:

```
[* IfListing AcceptsQResBookings *]  
Some text which only appears if the current listing has an active QRes subscription, pricing and availabilities and can accept QRes bookings.  
[* /IfListing *]
```

### HasQResSubscription

Example:

```
[* IfListing HasQResSubscription *]  
Some text which only appears if the current listing has an active QRes subscription, but may not have pricing, availabilities set up properly, and may not accept QRes bookings..  
[* /IfListing *]
```

### HasRoomData

Example:

```
[* IfListing HasRoomData *]  
Some text which only appears if the current listing has room data.  
[* /IfListing *]
```

### HasThirdPartyBookingSystem

Example:

```
[* IfListing HasThirdPartyBookingSystem *]  
Some text which only appears if the current listing has a third party booking system linked to it.  
[* /IfListing *]
```

### Field

Available fields are the same as for the [Listing](#) tag.

Example:

```
[* IfListing Field="Phone" *]  
Some text which only appears if the current listing has a phone number.  
[* /IfListing *]
```

### Not

Specifies that text is NOT shown if the condition is met.

Example:

```
[* IfListing Field="Phone" Not *]  
Some text which only appears if the current listing doesn't have a phone number.  
[* /IfListing *]
```

### 3.9.2.11 IfRequestParam

This tag outputs text conditionally depending on whether the named URL parameter exists

Example:

```
[* IfRequestParam Field="sk" *]
```

Some text which only appears if the **sk** URL parameter has a value

```
[* /IfRequestParam *]
```

#### Parameters

##### Field

Available fields are the same as for the [RequestParam](#) tag.

Example:

```
[* IfRequestParam Field="sk" *]
```

Some text which only appears if the **sk** URL parameter has a value

```
[* /IfRequestParam *]
```

##### Not

Specifies that text is NOT shown if the condition is met.

Example:

```
[* IfRequestParam Field="sk" Not *]
```

Some text which DOES NOT appear if the **sk** URL parameter has a value

```
[* /IfRequestParam *]
```

### 3.9.2.12 IfImage

This tag pair displays text if an image exists, for the current listing or article.

Example:

```
[* IfImage ID=1 *]
```

Some text which only appears if the current listing or article has an image in position 1.

```
[* /IfImage *]
```

#### Parameters

##### ID

This defines the image number which you wish to test for existence.

##### Not

Specifies that text is NOT shown if the condition is met.

Example:

```
[* IfImage ID=1 Not *]
```

Some text does NOT appear if the image in position 1 exists

```
[* /IfImage *]
```

### 3.9.2.13 IfFileExists

This tag pair displays text if a file exists.

Example:

```
[* IfFileExists "/global/logo.gif" *]  
Some text which only appears if the current listing or article has an image in  
position 1.  
[* /IfFileExists *]
```

#### Parameters

**Not**

Specifies that text is NOT shown if the condition is met.

Example:

```
[* IfFileExists "/global/logo" Not *]  
Some text does NOT appear if the image in position 1 exists  
[* /IfFileExists *]
```

### 3.9.3 Cookies

### 3.9.3.1 GetCookie

This tag returns a previously set cookie.

Example:

```
[* GetCookie name=LastSearchURL *]
```

#### Parameters

##### Name

You can use SetCookie to set your own named cookies.

Some pre-defined cookies are created by various components. These include:

```
LastSearchURL  
LastListingSearchURL
```

##### Format (optional)

This is the output format, including any prepended or appended text.

The advantage of using this for links etc, is that if the field itself is blank or null, then the surrounding text is also suppressed.

## 3.9.4 Searches

### 3.9.4.1 GeoSearch

These tags perform and display a search for a town, region, state or county.

Use [PerformGeoSearch](#) at the top of the page to perform the actual search.

Then, later in the page, you can render the results using [GeoSearchResults](#).

#### 3.9.4.1.1 PerformGeoSearch

This **non-visible** tag performs a geographical search (for **regions** and **towns**), which uses:  
"sk" parameter for keyword search,  
"qv" parameter for an existing geographical context.

The search results return regions and towns within the current page's boundaries.

They are sorted by region then town, with significant extra weighting given to those located inside the geographical context if supplied.

If there are one or more matching results, then the first search result becomes the new geographical context.

If no keywords are entered by the user, then the result set is empty.

Example:

```
[* PerformGeoSearch *]
```

### 3.9.4.1.2 PerformGeoSearch2

This **non-visible** tag performs a geographical search (for **areas**, **POIs**, and **towns**), which uses: "sk" parameter for keyword search, "qv" parameter for an existing geographical context.

This search differs from [PerformGeoSearch](#) such that it only returns results which match **every search word entered**.

If there are one or more matching results, then the first search result becomes the new geographical context.

If no keywords are entered by the user, then the result set is empty.

Example:

```
[ * PerformGeoSearch2 * ]
```

### 3.9.4.1.3 GeoSearchResults

This tag displays the results of a geographical search which was performed by the [PerformGeoSearch](#) tag.

Example:

```
[ * GeoSearchResults * ]
```

## Layout control

Default templates are provided which simply list results, however you can supply your own template if you wish to change the layout & content of search items. Within these files, The "Area", "POI", and "Town" tags can be used to access appropriate data fields.

Example

```
[ * GeoSearchResults AreaTemplate="geo_area.htm" POITemplate="geo_poi.htm"
TownTemplate="geo_town.htm" * ]
```

## Parameters

### ResultsPerPage

Set this to specify the number of search results per page. Default is 10.

Example: `[ * GeoSearchResults ResultsPerPage=20 * ]`

### AreaTemplate (optional)

Set this to specify a different filename for the area results template file.

The template file must exist in the /templates subdirectory of your website.

The default value if you don't set one is "geosearch\_region.htm".

Example: `[ * GeoSearchResults AreaTemplate="advanced_geosearch_area.htm" * ]`

### POITemplate (optional)

Set this to specify a different filename for the POI results template file.

The template file must exist in the /templates subdirectory of your website.

The default value if you don't set one is "geosearch\_poi.htm".

Example: `[ * GeoSearchResults POITemplate="advanced_geosearch_poi.htm" * ]`

### TownTemplate (optional)

Set this to specify a different filename for the town results template file.

The template file must exist in the /templates subdirectory of your website.  
The default value if you don't set one is "geosearch\_town.htm".

Example: `[ * GeoSearchResults TownTemplate="advanced_geosearch_town.htm" * ]`

#### [FirstPageIndex \(optional\)](#)

Only shows results from the specified index.

Zero is the first result on the page, 1 is the second, etc.

Example: `[ * GeoSearchResults FirstPageIndex=1 * ]`

#### [LastPageIndex \(optional\)](#)

Only shows results up until the specified index.

Zero is the first result on the page, 1 is the second, etc.

Example: `[ * GeoSearchResults LastPageIndex=1 * ]`

### 3.9.4.2 Article Search

These tags perform and display a listing search.

Use PerformListingSearch at the top of the page to perform the actual search.

Then, later in the page, you can render the results using ListingSearchResults.

### 3.9.4.2.1 PerformArticleSearch

This **non-visible** tag performs a listing search, which uses:  
"sk" parameter for keyword search,  
"qv" parameter for an existing geographical context.

The search results return articles within the current page's boundaries.

This tag automatically adds a cookie called "LastArticleSearchURL" which is the full URL the current page, which would be useful for a "Return to search results" link.

Example:

```
[* PerformArticleSearch *]
```

## Parameters

### ResultsPerPage

Set this to specify the number of search results per page. Default is 10.

Example: `[* PerformArticleSearch ResultsPerPage=20 *]`

### Group

Set this to limit results to certain group(s).

Groups are case sensitive.

Separate groups by comma

Example: `[* PerformArticleSearch Group="News,Features" *]`

### ExcludeGroups

Set this to limit results to certain group(s).

Groups are case sensitive.

Separate groups by comma

Example: `[* PerformArticleSearch ExcludeGroups="LocalEvents,Site" *]`

### IncludeUnlocated

Set this to any value to include non-located articles (articles without a location specified).

Example: `[* PerformArticleSearch IncludeUnlocated=True *]`

### RandomOrder

Set this to randomise the order of articles

Example: `[* PerformArticleSearch RandomOrder=true *]`

### PadResults

Set this to include articles outside the current area, to pad out the number of articles up to a specified number (default is zero, maximum is 10);

Example: `[* PerformArticleSearch PadResults=3 *]`

#### 3.9.4.2.2 ArticleSearchResults

This tag displays the results of a listing search which was performed by the PerformArticleSearch tag.

Example:

```
[* ArticleSearchResults *]
```

#### Layout control

Each item in the list is rendered using the template: `/global/templates/articlesearch.htm`

You can create this file in your own `/global/templates/` directory to override the default version. Within this file, The "Article" tag can be used to access appropriate data fields.

#### Parameters

##### Template (optional)

Set this specify a different filename for the search results template file.

The template file must exist in the `/global/templates/` subdirectory of your website.

The default value if you don't set one is "articlesearch.htm".

```
Example: [* ArticleSearchResults Template="advanced_articlesearch.htm" *]
```

##### Mod2Divider (optional)

Set some text to display after every second search result item, for example if your results are in a 2 column table.

```
Example: [* ArticleSearchResults Mod2Divider="</tr><tr>" *]
```

#### 3.9.4.2.3 ArticleSearchResultsNavBar

This tag displays a navigation bar for a search which was performed by the PerformArticleSearch tag.

Example:

```
[* ArticleSearchResultsNavBar *]
```

By default, a CSS file called `/global/css/articlesearchresultsnavbar.css` is inserted into the document's HEAD section which formats the navigation bar. You can create your own version of this css file in your own `/global/css/` directory to override the default style, or alternatively use the CustomStyle parameter to suppress the CSS reference.

#### Parameters

##### MinResults

Only show if the number of search results exceeds this number.

```
Example: [* ArticleSearchResultsNavBar MinResults := 11 *]
```

##### CustomStyle

Set this to any value to prevent a link to the default CSS file from being inserted into the document's HEAD section automatically.

```
Example: [* ArticleSearchResultsNavBar CustomStyle=True *]
```

#### 3.9.4.3 Listing Search

These tags perform and display a listing search.

Use PerformListingSearch at the top of the page to perform the actual search.

Then, later in the page, you can render the results using ListingSearchResults.

#### 3.9.4.3.1 PerformListingSearch

This **non-visible** tag performs a listing search, which uses:  
"sk" parameter for keyword search,  
"qv" parameter for an existing geographical context.

The search results return listings within the current page's boundaries.

They are sorted by region then town, with significant extra weighting given to those located inside the geographical context if supplied.

If there are one or more matching results, then the first search result becomes the new geographical context.

If no keywords are entered by the user, then the result set is empty.

This tag automatically adds a cookie called "LastListingSearchURL" which is the full URL the current page, which would be useful for a "Return to search results" link.

Example:

```
[* PerformListingSearch *]
```

### Parameters

#### ResultsPerPage

Set this to specify the number of search results per page. Default is 10.

Example: `[* PerformListingSearch ResultsPerPage=20 *]`

#### QResOnly

Set this to limit results to listings with an active QRes Connect subscription.

Example: `[* PerformListingSearch QResOnly=true *]`

#### KeywordSearch

Set this to include a keyword search based on the text entered in the "sk" http parameter.

Example: `[* PerformListingSearch KeywordSearch=true *]`

#### EDFClass

Set this to limit results to a particular EDF class.

Example: `[* PerformListingSearch EDFClass=1 *]`

- 1 = Basic accommodation listings
- 2 = Restaurants
- 3 = Attractions
- 4 = Businesses
- 5 = QRes accommodation listings

#### Category

Set this to limit results to a particular category or set of categories, using the Category ID.

Categories are arranged in a hierarchy. If you specify a category which has "children", then all children are included implicitly.

Multiple categories can be specified, use a comma between each one.

#### CountryID

Set this to limit results to a particular Country.

Multiple Country IDs can be added using a comma to separate them

Example 1: [\* PerformListingSearch CountryID="au" \*]

Example 2: [\* PerformListingSearch CountryID="au,nz" \*]

### FeatureID

Set this to limit results to listings that have particular FeatureID(s) enabled.

Example: [\* PerformListingSearch FeatureID=1 \*]

Example: [\* PerformListingSearch FeatureID=1,2,3 \*]

Note that specific FeatureIDs are only applicable to specific EDFClasses.

Current available FeatureIDs:

ID	Feature	EDFClass
1	Shower/bath	1
2	Games room	1
3	Toilets	1
4	Air conditioning	1
6	Heating	1
7	Telephone	1
8	Television	1
9	Movies	1
10	Clock radio	1
12	Tea/coffee making	1
13	Refrigerator	1
14	Toaster	1
15	Mini bar	1
16	Cooking facilities	1
17	Microwave	1
18	Laundry	1
19	Disabled facilities	1
20	Common lounge	1
21	Pool	1
22	Spa	1
23	Sauna	1
24	Barbecue	1
25	Car parking	1
26	Playground	1
27	Pets	1
28	Room service	1
29	Restaurant	1
30	Conference facilities	1
31	Courtesy pickup	1
32	Gym	1
33	Tennis court	1
34	Golf course	1
35	Skiing	1
36	Fishing	1
37	Tours	1
38	Horseriding	1
39	Boats	1
40	Non smoking rooms	1
46	Waterfront	1
51	Bowling Green	1
52	Business centre	1
53	Internet access	1
61	Adult Stays	1
62	Adventure Stays	1
63	Backpacker Stays	1
64	Beach Stays	1

65	Boating Stays	1
66	Eco Stays	1
67	Family Stays	1
68	Fishing Stays	1
69	Golf Stays	1
70	Health Stays	1
71	Holy Stays	1
72	Honeymoon Stays	1
73	Kid Stays	1
74	Luxury Stays	1
75	Pet Stays	1
76	Respite Stays	1
77	School Field Trips	1
78	Scuba Stays	1
79	Seniors	1
80	Snow Stays	1
81	Space Stays	1
82	Surfing Stays	1
83	Teenage Stays	1
84	Tourist Parks	1
85	Wheelchair Stays	1
86	4WD Stays	1
101	Licenced	2
102	Take away	2
103	Home delivery	2
104	BYO	2

### 3.9.4.3.2 ListingSearchResults

This tag displays the results of a listing search which was performed by the [PerformListing Search](#) tag.

Example:

```
[* ListingSearchResults *]
```

#### Layout control

Each item in the list is rendered using the template: `/global/templates/listingsearch.htm`

You can create this file in your own `/global/templates/` directory to override the default version. Within this file, The "Listing" tag can be used to access appropriate data fields.

#### Parameters

##### Template (optional)

Set this specify a different filename for the search results template file.

The template file must exist in the `/global/templates/` subdirectory of your website.

The default value if you don't set one is "listingsearch.htm".

Example: `[* ListingSearchResults Template="advanced_listingsearch.htm" *]`

##### Mod2Divider (optional)

Set some text to display after every second search result item, for example if your results are in a 2 column table.

Example: `[* ListingSearchResults Mod2Divider="</tr><tr>" *]`

##### FirstPageIndex (optional)

Only shows results from the specified index.

Zero is the first result on the page, 1 is the second, etc.

Example: `[* ListingSearchResults FirstPageIndex=1 *]`

##### LastPageIndex (optional)

Only shows results up until the specified index.

Zero is the first result on the page, 1 is the second, etc.

Example: `[* ListingSearchResults LastPageIndex=1 *]`

### 3.9.4.3.3 ListingSearchResultsNavbar

This tag displays a navigation bar for a search which was performed by the [PerformListing Search](#) tag.

Example:

```
[* ListingSearchResultsNavBar *]
```

By default, a CSS file called `/global/css/listingsearchresultsnavbar.css` is inserted into the document's HEAD section which formats the navigation bar. You can create your own version of this css file in your own `/global/css/` directory to override the default style, or alternatively use the `CustomStyle` parameter to suppress the CSS reference.

#### Parameters

##### MinResults

Only show if the number of search results exceeds this number.

Example: `[* ListingSearchResultsNavBar MinResults := 11 *]`

### CustomStyle

Set this to any value to prevent a link to the default CSS file from being inserted into the document's HEAD section automatically.

Example: `[* ListingSearchResultsNavBar CustomStyle=True *]`

### RemoveParams

Set this to remove any URL parameters from the links. For example, use it to remove the search keywords parameter if the search is a geo-search.

Example: `[* ListingSearchResultsNavBar RemoveParams=sk *]`

### AddParams

Set this to remove add any URL parameters to the links.

Example: `[* ListingSearchResultsNavBar AddParams="this=that&foo=bar" *]`

#### 3.9.4.3.4 LinearAvailabilityGrid

This tag displays the results of a listing search which was performed by the [PerformListing Search](#) tag as an availability grid.

It uses a style sheet called `/global/css/qrcbookingsingleroomwizard.css` to control its appearance, however you can override this.

### Parameters

All parameters are optional.

#### ListingLink (optional)

The URL to website's listing detail page. This is used to turn the listing's name into a hyperlink.

Example: `[* LinearAvailabilityGrid ListingLink="False" *]`

#### TruncRoomName (optional)

Truncates all room names to a specified number of characters.

Example: `[* LinearAvailabilityGrid TruncRoomName="20" *]`

#### CellFormat (optional)

Controls the format of prices shown in each cell.

Example: `[* LinearAvailabilityGrid CellFormat="%0.0f" *]`

#### CustomStyle (optional)

Set this to any value to prevent a link to the default CSS file from being inserted into the document's HEAD section automatically. See the [StyleSheet](#) section below for more details.

Example: `[* LinearAvailabilityGrid CustomStyle=True *]`

#### NameTemplate (optional)

Set this to load your own custom template to alter what appears in place of each listing's name. The template must exist in your

Leave blank to simply display the listing's name.

Example: `[* LinearAvailabilityGrid NameTemplate="grid_template.htm" *]`

### Stylesheet

By default, a link to the file `/global/css/linearavailabilitygrid.css` is added to the head section of your HTML document if you use this component. If your website doesn't contain this CSS file, then a default

one is returned by the web server. You can get access to the default CSS file by pulling up `http://yoursiteid.qrconnect.net/global/css/linearavailabilitygrid.css` in a web browser.

You can edit this file and place it into the `/global/css/` directory of your website to control the appearance of your booking wizard.

Alternatively, you can use the `CustomStyle` parameter to suppress the reference to `/global/css/qrcbookingsingleroomwizard.css`, and add your own stylesheet reference in the HEAD section of the document.

### 3.9.5 Set and Get

Allows you to set any arbitrary named value, and get it again later.

Example use:

This may be useful if you use a common included header file, but there is something inside header which is variable.

For example, to have a header which uses a different logo image for different content pages, your header file might contain:

```
...

...
```

And your content page could contain:

```
...
[* set MyLogoFile="blackandwhitelogo.jpg" *]
[* include file="header.htm" *]
<li>This page has a black & white logo.
...
```

And a different content page could contain:

```
...
[* set MyLogoFile="colourlogo.jpg" *]
[* include file="header.htm" *]
<li>This page has a colour logo.
...
```

Notes:

You can set as many different named values as you wish.

You need to **set** a value before you can **get** it.

You can set multiple values in a single **set** tag, for example:

```
[* set MyLogoFile="blackandwhite.jpg" MyLogoFileHeight="100" MyLogoFileWidth="200" *]
```

However, you can only get one value per **get** tag.

### 3.9.5.1 Set

This tag sets a string variable which you can retrieve later.

Example:

```
[* Set MyVariableName="Some string value" *]
```

#### Parameters

You decide the names of the parameters.  
They can be any name you want.

### 3.9.5.2 Get

Retrieves a string value previously set by [Set](#)

Example: `[* Get MyVariableName *]`

### 3.9.5.3 Inc

This tag increments a numeric variable.  
If the variable empty or not numeric, then it is assumed to be zero.

Example:

```
[* Inc MyVariableName *]
```

### 3.9.5.4 IfSet

This tags outputs text if a specified variable has been set.  
If the variable has not been set, or if it has been set to nothing, then the text between the tags is ignored.

Example:

```
[* IfSet MyVariableName *]
```

Some text which only appears if MyVariableName has been set to a something

```
[* /IfSet *]
```

#### Parameters

The variable name which you want to test is required.

### 3.9.5.5 IfNotSet

This tag outputs text if a specified variable has not been set. If the variable has been set to any value, then the text between the tags is ignored.

Example:

```
[* IfNotSet MyVariableName *]  
Some text which only appears if MyVariableName has been set to a something  
[* /IfNotSet *]
```

### Parameters

The variable name which you want to test is required.

## 3.9.6 BreadCrumb Navigation Bar

The breadcrumb system creates a Yahoo style structural navigation bar, like this:

**[Home](#) -> [Australia](#) -> [New South Wales](#) -> [Sydney](#) -> [The Lazy Days Motel](#)**

Breadcrumbs are extremely valuable tools, not only functioning as navigational shortcuts "up" and "down" the site tree, but also helping users orient themselves in the site. Further reading [here](#).

Use the [BreadCrumb](#) tag to add or insert links, then use the [BreadCrumbNavBar](#) tag to render the actual navigation bar.

You can also use the [AddGeoBreadCrumbs](#) tag to automatically add breadcrumbs for the nearest region, or the state/county and town for the page's current geographical context.

### 3.9.6.1 AddBreadCrumb

Adds a link to the breadcrumb navigation bar.

Example:

```
[* BreadCrumb Position="Start" Name="Home" Caption="Home" URL="[*URL "/"*]" *]
```

#### Parameters

##### Position

Indicates where you want to add the node.

Valid values are **Start**, **End** or a numerical value to insert at a specific position, where 0 (zero) inserts it at the start before the first node, 1 inserts before the second node, etc.

##### Name (optional)

This is a name for the node. Names must be unique, so if you add a node which already exists, it is overwritten.

##### Caption

This is the visible caption of the node.

##### URL

When clicked, the node sends the user to this address.

### 3.9.6.2 AddGeoBreadCrumbs

Adds Country, State/County, Region, Town and Listing [breadcrumbs](#) according to the current geographical location of the page.

Example:

```
[* AddGeoBreadCrumbs *]
```

At least one URL parameter is required to add any nodes at all.  
If the page has no geographical context, then no nodes are added.

#### How it works:

If a town is selected, then that town's State/County appears.

If a listing is selected, then that listing's State/County, town, and listing appears.

If an article is selected, then that article's state/country, and town appears

If an area is selected, then that area is used.

If nothing is selected, then the highest ranking region and town in the visible area are used.

#### However:1

Regions & Towns are only shown if the zoom level is equal to or below their maximum thresholds.

### Parameters

[URL](#) (optional)

Indicates the URL to link to. Default is the current page.

### 3.9.6.3 BreadCrumbNavBar

Renders all nodes which you have previously added using the [BreadCrumb](#) tag, and will appear something like this:

**[Home](#) -> [Australia](#) -> [Accommodation](#) -> [The Lazy Days Motel](#) -> [Bookings](#)**

Styles can be found in [/global/css/default.css](#).

Example: `[* BreadCrumbNavBar *]`

### Parameters

[Delimiter](#)

Specifies the string which appears between each node. If left blank, defaults to "->".

Example 1: `[* BreadCrumbNavBar Delimiter=" | " *]`

Example 2: `[* BreadCrumbNavBar Delimiter="<img src=/global/icons/arrow.gif>" *]`

## 3.9.7 Miscellaneous Utility Tags

The following pages contain various useful components.

### 3.9.7.1 FeatureList

Returns a table of the current listing's features.

Example:

```
[* FeatureList Cols = 2 *]
```

#### Parameters

##### Columns

The number of columns for the table. Default is 3.

### 3.9.7.2 ListingTypeList

Returns a drop-down list of listing types

Example:

```
[* ListingTypeList *]
```

Note: all parameters entered other than those specified below will become parameters of the <select> tag.

#### Parameters

##### EDFClass

Limits the list to the specified class. See the [PerformListingSearch](#) page for possible values.

##### Level

Listing types are arranged in a hierarchy.

This parameter specifies which level of the hierarchy to return.

Level 1 is the default, which is the highest level.

##### IncludeCountForAll

Set this parameter to any value to include the total number of listings, on the first line.

##### IncludeCount

Set this parameter to any value to include the number of listings next to each type.

### 3.9.7.3 Random

This tag randomly picks a random "thing" within the current view and returns it's name.

Example:

```
[* Random field=Town InTheTop=10 *]
```

#### Parameters

**Field** (required)

Defines the "thing" you want to return.

Current available values are:

- Town
- ListingType

**InTheTop** (optional)

This is the output format, including any prepended or appended text.

The advantage of using this for captions etc, is that if the field itself is blank or null, then the caption is also suppressed.

Default is 100.

### 3.9.7.4 Currency

This tag returns a currency sign based on the current location.

Example:

```
[* Currency *]
```

#### Parameters

**Field** (default is HTML)

Available values are:

- ID
- Sign
- HTML

### 3.9.7.5 TownList

Returns a drop-down list of towns.

Example:

```
[* TownList *]
```

#### Parameters

**Max** (optional)

This is the maximum number of towns in the list.

Default is 100.

Maximum is 1000.

**AreaID** (optional)

Instead of using the page's current geographic context, uses this Area ID instead.

**IncludeAll** (optional)

Add an option to the top of the list which includes the null value "All"

**SortAlpha** (optional)

Set to a non-blank value to sort alphabetically.

Any other parameters become part of the <select> tag.

### 3.9.7.6 AreaList

Returns a drop-down list of towns.

Example:

```
[* AreaList Level=5 *]
```

#### Parameters

**Level** (required)

5 = Continent

10 = Country

15 = MultiState Area

20 = State or County

40 = Region

**IncludeAll** (optional)

Add an option to the top of the list which includes the null value "All"

**Max** (optional)

This is the maximum number of towns in the list.

Default is 100.

Maximum is 1000.

### 3.9.7.7 SelectedLocation

Returns the name of the currently selected location (Town, POI, Country, State, Region or Tourism Region).

Example:

```
[* SelectedLocation *]
```

### Parameters

**Format** (optional)

This is the output format, including any prepended or appended text.

The advantage of using this for captions etc, is that if the field itself is blank or null, then the caption is also suppressed.

Use the token %s to specify where the field value appears in the format string.

### 3.9.7.8 RegionID

Returns the ID of the current Region.

If you don't have a town, POI or region selected explicitly, the system automatically works out the most dominant region in the current view.

Example:

```
[* RegionID *]
```

### 3.9.7.9 RegionOrTownPhoto

Returns an image for the current region or town.

NOTE:

Region photos must exist in the following location and filename:  
/global/areas/<areaid>.jpg

Town photos must exist in the following location and filename:  
/global/towns/

Example:

```
[ * TownList * ]
```

#### Parameters

**Max** (optional)

This is the maximum number of towns in the list.  
Default is 100.  
Maximum is 1000.

**AreaID** (optional)

Instead of using the page's current geographic context, uses this Area ID instead.

**IncludeAll** (optional)

Add an option to the top of the list which includes the null value "All"

**SortAlpha** (optional)

Set to a non-blank value to sort alphabetically.

Any other parameters become part of the <select> tag.

## 3.9.8 Relaxas Only Tags

### 3.9.8.1 GoogleAPIKey

Returns the Google API Key which has been added to the SiteDomains table.

### 3.9.9 SearchIndex

This tag returns the current search result item number.  
Useful in search templates.

Search indexes are zero-based, so the first search result is 0 (zero) the second is 1, etc.  
Use the "add" parameter to add 1 to the index for display purposes.

Example:

```
[* SearchIndex add=1 *]
```

#### Parameters

##### Add

Adds the specified number to the search index number.

##### OddOrEven

Instead of returning a number, use this parameter to return the words "odd" or "even"

### 3.9.10 PageIndex

This tag returns the current search result item number **for the current page**.  
Useful in search templates.

Search indexes are zero-based, so the first search result is 0 (zero) the second is 1, etc.  
Use the "add" parameter to add 1 to the index for display purposes.

Example:

```
[* PageIndex add=1 *]
```

#### Parameters

##### Add

Adds the specified number to the search index number.

##### OddOrEven

Instead of returning a number, use this parameter to return the words "odd" or "even"

### 3.9.11 AddStyleSheet

This tag adds a reference to a stylesheet to the document's <head> section.  
If the stylesheet reference already exists, it won't add a duplicate reference.

There is only 1 parameter, which is the absolute path of the stylesheet reference you wish to insert.

Note: the stylesheet's path is automatically mapped, so the use of [\*Root\*] is not required.

Example: `[* AddStyleSheet "/global/css/something.css" *]`

### 3.9.12 RedirectTo

This tag ceases the current page and redirects to the specified destination

Example: `[ * RedirectTo "http://www.mywebsite.com/" * ]`

### 3.9.13 SiteURL

This tag returns a full URL to the path or document specified.

It automatically determines the best domain name to use, and adjusts the path accordingly.

There is only 1 parameter, which is the absolute path of the document or file you wish to link to.

Example: `[ * SiteURL "/something/" * ]`

Where "something" is a subdirectory of your website's root directory.

### 3.9.14 SecureURL

This tag returns a full SSL (Secure Sockets Layer) URL to the path or document specified.

It automatically uses the system's secure hostname (**qrconnect.net**) and adjusts the path accordingly.

There is only 1 parameter, which is the absolute path of the document or file you wish to link to.

Example: `[ * SecureURL "/something/" * ]`

Where "something" is a subdirectory of your website's root directory.

### 3.9.15 SetView

The SetView tag doesn't produce any HTML output.

It geo-references the page, ie it provides a geographical context for other components, such as Map, to use.

Because of this, the SetView tag should generally be the first tag to appear on a page.

Example: `[ * Setview Height=200 Width=250 Lat=0251500S Long=1310000E Zoom=18 * ]`

Example: `[ * Setview Height=200 Width=250 RegionID=au NoBoundaries=true * ]`

#### Parameters

##### Height

The height of the view in pixels.

##### Width

The width of the view in pixels.

##### RegionID

The ID of the Country/State/Region to centre the view on. If you specify a RegionID, then Lat, Long and Zoom are ignored.

##### Lat

The latitude of the centre point of the view area.

### Long

The latitude of the centre point of the view area.

### Zoom

This number, between 1 and 20, defines the zoom.

Level 20 is extremely zoomed out, showing the entire earth

Level 10 is zoomed in to street level, which is about the highest practical magnification level for this system.

Level 1 is zoomed in far enough to count an ant's teeth.

Zoom level 10 = 1024 degrees per 1048576 pixels = 1024 pixels per degree

Zoom level 11 = 2048 degrees per 1048576 pixels = 512 pixels per degree

Zoom level 12 = 4096 degrees per 1048576 pixels = 256 pixels per degree

Zoom level 13 = 8192 degrees per 1048576 pixels = 128 pixels per degree

Zoom level 14 = 16384 degrees per 1048576 pixels = 64 pixels per degree

Zoom level 15 = 32768 degrees per 1048576 pixels = 32 pixels per degree

Zoom level 16 = 65536 degrees per 1048576 pixels = 16 pixels per degree

Zoom level 17 = 131072 degrees per 1048576 pixels = 8 pixels per degree

Zoom level 18 = 262144 degrees per 1048576 pixels = 4 pixels per degree

Zoom level 19 = 524288 degrees per 1048576 pixels = 2 pixels per degree

Zoom level 20 = 1048576 degrees per 1048576 pixels = 1 pixel per degree

### NoBoundaries

Add this parameter (set to any value, such as "true") if you don't want the user to be restricted from panning the map outside of the area you have specified. This is not required if you already use the SetBoundaries before the SetView tag.

### 3.9.16 SetArticle

The SetArticle tag doesn't produce any HTML output.

It allows you to set a current article for the page.

Example: `[ * SetArticle ArticleID=HomePageIntroText * ]`

#### Parameters

##### ArticleID (required)

The article ID.

### 3.9.17 SetBoundaries

#### NOTE:

Bounds are automatically established when you use the SetView tag, so you probably never need to use this tag.

It is useful if you want the map to start in a certain position by default, but still allow it to pan beyond the default.

The SetBoundaries tag doesn't produce any HTML output. It provides latitude/longitude boundaries for the page, to prevent users from panning a map outside the bounds specified.

Example: `[ * SetBoundaries top=0500000S bottom=0600000S left=1200000E right=1600000E * ]`

#### Parameters

##### Top

The new top boundary of the view, in dddmmssX format, where ddd=degrees, mm=minutes, ss=seconds, X is "N" for north or "S" for south.

##### Bottom

The new bottom boundary of the view, in dddmmssX format.

##### Left

The new left boundary of the view, in dddmmssX format, where ddd=degrees, mm=minutes, ss=seconds, X is "E" for east or "W" for west.

##### Right

The new right boundary of the view, in dddmmssX format.

### 3.9.18 Map

Draws a map including navigation buttons (images), based on the current view.

Styles can be found in `/global/css/map.css`.

The navigation widgets images are GIF files located in `/global/icons/map` which can also be overridden with your own images.

Example: [ \* Map \* ]

## Parameters

### URL

The base URL for links on the map & navigation widgets.

### ImageFormat

Can be "JPEG" or "PNG" - defines the type of image file you want the map to be. JPEG is a smaller file size but lower quality, PNG is high quality but larger file size (slower to download).

### ImgHeights

The height of the navigation images in pixels.  
Set this if you're using your own navigation images and they're not 16 pixels high.

### ImgWidths

The width of the navigation images in pixels.

### RegionLink

The URL of the links for clickable region labels on the map.  
Insert %s where you want the Region ID to appear.

### TownLink

The URL of the links for clickable town labels on the map.  
Insert %s where you want the Region ID to appear.

### ListingMarkerOffsetX

The number of pixels to offset listing markers on the X axis. Default is zero.

### ListingMarkerOffsetY

The number of pixels to offset listing markers on the Y axis. Default is -24.

### ArticleMarkerOffsetX

The number of pixels to offset article markers on the X axis. Default is zero.

### ArticleMarkerOffsetY

The number of pixels to offset article markers on the Y axis. Default is -24.

### HideArticleMarkerText

Set this to any value to suppress the number shown in the article marker.

### 3.9.19 GoogleMap

Inserts a Google map including based on the current view.

Example: `[* GoogleMap *]`

#### NOTE:

To work properly, the javascript must be initialised in the <body> tag.

Example:

```
<body onload="load()" onunload="GUnload()">
```

The best/typical way to achieve this when using a common header is as follows:

```
<body[*IfSet IncludeGoogleMap*] onload="load()" onunload="GUnload()"[*/IfSet*]>
```

Then in each page which has a map on it, add the tag:

```
[* Set IncludeGoogleMap=true *]
```

#### Parameters

##### ShowMarker

Shows a pinpoint marker based on the current listing location.

### 3.9.20 NavProfile

Returns a string which represents the geographic context of the current page.

This string is universally identified by the name "qv", and can be used as either a POST or GET parameter

POST Example:

```
<input type="hidden" name="qv" value="[* NavProfile *]">
```

GET Example:

```
<a href="map.htm?qv=[* NavProfile *]">
```

#### Parameters

##### DiscardSelection

Set this to any value to discard the selected region or town.

### 3.9.21 Title Keywords and Description

Allows you to set the HTML page's Title, Keywords and Description meta tags.

Example use:

This may be useful if you use a common header file, but you wish to assign a different Title, Keywords or Description to the page.

(Also see [Get and Set](#) for similar functionality with your own page variables)

```
[* Title "This is the page title" *]  
[* Keywords "Keyword1,Keyword2,Keyword3" *]  
[* Description "This is the page's description." *]
```

### 3.9.22 RequestParam

This tag returns a HTTP parameter of the current page.

Example:

```
[* RequestParam field=sk *]
```

#### Parameters

##### Field

This defines the HTTP Get or Post parameter which you wish to output. The field is named in the request parameter,

##### Format (optional)

This is the output format, including any prepended or appended text.

The advantage of using this is that if the field itself is blank or null, then the caption is also suppressed.

### 3.9.23 Image

This tag returns a link to an image for a listing, article, region or town.

Example:

```
[* Image ID=1 *]
```

#### Parameters

##### ID

This defines the image number which you wish to display.

##### Format (optional)

This is the output format, including any prepended or appended text.

The advantage of using this for captions etc, is that if the image doesn't exist, then the extra stuff in the format parameter is also suppressed.

##### FilenameOnly (optional)

Set this to some value to specify that a filename is returned instead of the entire <img> tag.

##### ThumbHeight (optional)

Set this if you wish to return a thumbnail image. Height is in pixels.

##### ThumbWidth (optional)

Set this if you wish to return a thumbnail image. Width is in pixels.

### 3.9.24 ImageLabel

This tag returns a URL to an image which contains text.

Useful for phone numbers, email addresses etc where you want text to be readable but not readable by robots.

Example:

```
[* ImageLabel caption="Robots can't read this!" *]
```

## Parameters

### Caption

This is the text you want to render to an image.

### ImageFormat

This can be JPG or PNG. Default is PNG.

### FontName (Default is Verdana10)

The following fonts are currently available (they are case sensitive):

```
FontName=Verdana8  
FontName=Verdana8B  
FontName=Verdana9  
FontName=Verdana9B  
FontName=Verdana10  
FontName=Verdana10B  
FontName=Verdana11  
FontName=Verdana11B  
FontName=Verdana12  
FontName=Verdana1B  
FontName=TrebuchetMS8  
FontName=TrebuchetMS10  
FontName=TrebuchetMS12
```

Example:

```
[* ImageLabel fontname="Verdanall" caption="Robots can't read this!" *]
```

Please contact us if you wish to have a specific font added to the list.

### FontColor (optional)

The color of the text, in HTML format.

### BGColor (optional)

The color of the background, in HTML format.

## 3.10 Notes

### 3.10.1 Date Formats

Date/Time Format Strings specify the formatting of date/time values when they are converted to strings.

Date Time Format Strings are composed from specifiers that represent values to be inserted into the formatted string.

#### Examples

The date format "dd mmm yyyy" would return a date in the form "08 Jun 2006"

The date format "d mmmm yyyy" would return a date in the form "8 June 2006"

#### Format Specifiers

In the following table, specifiers are given in lower case. Case is ignored in formats, except for the "am/pm" and "a/p" specifiers.

Specifier	Displays
c	Displays the date using the format given by the ShortDateFormat global variable, followed by the time using the format given by the LongTimeFormat global variable. The time is not displayed if the date-time value indicates midnight precisely.
d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).
ddd	Displays the day as an abbreviation (Sun-Sat) using the strings given by the ShortDayNames global variable.
dddd	Displays the day as a full name (Sunday-Saturday) using the strings given by the LongDayNames global variable.
dddddd	Displays the date using the format given by the ShortDateFormat global variable.
ddddddd	Displays the date using the format given by the LongDateFormat global variable.
e	(Windows only) Displays the year in the current period/era as a number without a leading zero (Japanese, Korean and Taiwanese locales only).
ee	(Windows only) Displays the year in the current period/era as a number with a leading zero (Japanese, Korean and Taiwanese locales only).
g	(Windows only) Displays the period/era as an abbreviation (Japanese and Taiwanese locales only).
gg	(Windows only) Displays the period/era as a full name. (Japanese and Taiwanese locales only).
m	Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mm	Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mmm	Displays the month as an abbreviation (Jan-Dec) using the strings given by the ShortMonthNames global variable.
mmmm	Displays the month as a full name (January-December) using the strings given by the LongMonthNames global variable.
yy	Displays the year as a two-digit number (00-99).
yyyy	Displays the year as a four-digit number (0000-9999).
h	Displays the hour without a leading zero (0-23).
hh	Displays the hour with a leading zero (00-23).
n	Displays the minute without a leading zero (0-59).
nn	Displays the minute with a leading zero (00-59).
s	Displays the second without a leading zero (0-59).
ss	Displays the second with a leading zero (00-59).
z	Displays the millisecond without a leading zero (0-999).

- zzz Displays the millisecond with a leading zero (000-999).
- t Displays the time using the format given by the ShortTimeFormat global variable.
- tt Displays the time using the format given by the LongTimeFormat global variable.
- am/pm Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour before noon, and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
- a/p Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before noon, and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
- ampm Uses the 12-hour clock for the preceding h or hh specifier, and displays the contents of the TimeAMString global variable for any hour before noon, and the contents of the TimePMString global variable for any hour after noon.