Atlantis Developer Guide – Template Captcha control

Contents

[Atlantis.Framework.Web.CaptchaCtl 1](#_Toc359580607)

# Atlantis.Framework.Web.CaptchaCtl

Implementing Captcha into the web site has been difficult and technically challenging. This control encapsulates all the complexity required and provides a template interface for customization of how the Captcha can be rendered.

For client side the Atlantis.Framework.Web.Stash architecture can be used to put the script at any location desired, however it is not required to actually use this control.

There is a possible dependency on the bot\_detect\_<version>.js that is located at: [\\g1dwimages001\AtlantisScripts\Atlantis.Web.Controls\Captcha](file:///\\g1dwimages001\AtlantisScripts\Atlantis.Web.Controls\Captcha) if the developer desires to use the controller contained in that file for processing the Captcha.

1. Getting Started

To get started, you will need to add the following entry in the pages, controls section of your web.config:

<configuration>

<configSections>

<!-- Register the BotDetect configuration section -->

<section name="botDetect" requirePermission="false" type="BotDetect.Configuration.BotDetectConfigurationSection, BotDetect"/>

</configSections>

<system.web>

<pages maxPageStateFieldLength="-1" controlRenderingCompatibilityVersion="3.5" clientIDMode="AutoID">

<controls>

<add assembly="BotDetect" namespace="BotDetect.Web.UI" tagPrefix="BotDetect"/>

<add tagPrefix="atlantis" namespace="Atlantis.Framework.Web.Stash" assembly="Atlantis.Framework.Web.Stash"/>

<add tagPrefix="atlantis" namespace="Atlantis.Framework.Web.CaptchaCtl" assembly="Atlantis.Framework.Web.CaptchaCtl"/>

</controls>

</pages>

</system.web>

<system.webServer>

<handlers>

<add name="BotDetectCaptchaHandler" preCondition="integratedMode" verb="GET" path="BotDetectCaptcha.ashx" type="BotDetect.Web.CaptchaHandler, BotDetect"/>

</handlers>

</system.webServer>

<botDetect>

<captchaImage>

<helpLink enabled="false" mode="image"/>

</captchaImage>

</botDetect>

</configuration>

1. Using the Control
   1. Properties
      1. Configuration

**AjaxAuthentication:** Boolean used to turn on standard ajax authentication

**AjaxAuthenticationCallback:** Jscript function to call on validation complete

**AutoBindEvents:** emit jquery binding

**AutoClearInput:** clear input when processing captcha

**AutoFocusInput:** focus input on load

**AutoReloadImage:** reload Captcha image after set interval

**AutoReloadMiliseconds:** interval in miliseconds for reload of image

**CaptchaID**: Unique instance of Captcha on page

**CaptchaImagePlaceHolderID:** placeholder id in template to show where image should render

**CaptchaValidateSelector:** selector used during binding of validation

**CaptchaValueID:** id of the text box used for input of Captcha values

**ImageReloadPlaceHolderID:** placeholder id in template to show where reload image should render

**JQueryDelim:**  jquery delimiter $ can be replaced

* + 1. Child Controls

**LoadingImagePlaceHolder:** Loading Image Placeholder control

**ReloadImage:** Reload img control

**PlaySoundImage:** Play Sound img control

**StashContentControl:** Stash content control

* 1. Using The template functionality

By adding placeholders and setting the appropriate id property you can tell the control where to put the Captcha image, reload image, play sound image, etc.

An example of this is as follows:

<atlantis:CaptchaTemplateControl ID="captchaTemplate1" runat="server" CaptchaImagePlaceHolderID="plcCaptchaLocation" StashRenderLocation="javascriptStash" CaptchaValidateSelector=".captchaValidate" CaptchaValueID="captchaTextBox" ImageReloadPlaceHolderID="plcImageReload" PlaySoundPlaceHolderID="plcPlaySound" CaptchaID="standardAjaxValidation" AjaxAuthentication="true" AjaxAuthenticationCallback="AjaxAuthenticationResults" AutoClearInput="true" AutoFocusInput="true" LoadingText="Loading..." AutoReloadImage="true">

<Template>

<div style="width: 330px;">

<div style="float: left; padding-bottom: 10px;">

<span style="padding-right: 10px;">Access Code:</span>

<asp:PlaceHolder ID="plcCaptchaLocation" runat="server"></asp:PlaceHolder>

</div>

<div style="float: right; padding-right: 20px; padding-top: 20px;">

<asp:PlaceHolder ID="plcImageReload" runat="server"></asp:PlaceHolder>

<br />

<asp:PlaceHolder ID="plcPlaySound" runat="server"></asp:PlaceHolder>

</div>

</div>

<div style="clear: both;">

<span style="padding-right: 10px;">Please Enter the Access Code:</span><br />

<input type="text" name="captchaTextBox" id="captchaTextBox" maxlength="90" style="width: 180px;" />

<br />

<div id="captchaError" style="display: none; width: 200px; text-align: center; padding-top: 5px;">

<span class="bodyText" style="color: #cc0000">

<b>Authentication failed.&nbsp;Please try again.</b>

</span>

</div>

<br />

<div class="g-btn-lg g-btn-prg captchaValidate" style="border: none; cursor: pointer">

Validate Me

</div>

</div>

</Template>

</atlantis:CaptchaTemplateControl>

<asp:PlaceHolder ID="plcCaptchaLocation" runat="server"></asp:PlaceHolder>

plcCaptchaLocation is placed in the attribute CaptchaImagePlaceHolderID of the control … telling the control that where that placeholder is in the template is where the Captcha image should be placed.

This pattern is followed for all TemplatePlaceholderID category properties. This allows the developer to completely control the UI of the Captcha rendering through standard html in the template and placeholders for the control rendered UI.

1. Validation Options
   1. Standard Ajax Validation

This functionality is embedded in the control and through the use of it… no coding or jquery scripting is required except to handle the result of the validation.

Here is an example of the control using standard ajax validation:

<atlantis:CaptchaTemplateControl ID="captchaTemplate1" runat="server" CaptchaImagePlaceHolderID="plcCaptchaLocation" StashRenderLocation="javascriptStash" CaptchaValidateSelector=".captchaValidate" CaptchaValueID="captchaTextBox" ImageReloadPlaceHolderID="plcImageReload" PlaySoundPlaceHolderID="plcPlaySound" CaptchaID="standardAjaxValidation" AjaxAuthentication="true" AjaxAuthenticationCallback="AjaxAuthenticationResults" AutoClearInput="true" AutoFocusInput="true" LoadingText="Loading..." AutoReloadImage="true">

<Template>

<div style="width: 330px;">

<div style="float: left; padding-bottom: 10px;">

<span style="padding-right: 10px;">Access Code:</span>

<asp:PlaceHolder ID="plcCaptchaLocation" runat="server"></asp:PlaceHolder>

</div>

<div style="float: right; padding-right: 20px; padding-top: 20px;">

<asp:PlaceHolder ID="plcImageReload" runat="server"></asp:PlaceHolder>

<br />

<asp:PlaceHolder ID="plcPlaySound" runat="server"></asp:PlaceHolder>

</div>

</div>

<div style="clear: both;">

<span style="padding-right: 10px;">Please Enter the Access Code:</span><br />

<input type="text" name="captchaTextBox" id="captchaTextBox" maxlength="90" style="width: 180px;" />

<br />

<div id="captchaError" style="display: none; width: 200px; text-align: center; padding-top: 5px;">

<span class="bodyText" style="color: #cc0000">

<b>Authentication failed.&nbsp;Please try again.</b>

</span>

</div>

<br />

<div class="g-btn-lg g-btn-prg captchaValidate" style="border: none; cursor: pointer">

Validate Me

</div>

</div>

</Template>

</atlantis:CaptchaTemplateControl>

<script type="text/javascript">

function AjaxAuthenticationResults(result) {

$('#captchaError').hide();

if (!result) {

$('#captchaError').show();

}

else {

alert(result);

}

}

When standard Ajax validation occurs a request is made to the control handler including the input values and instanceID of the Captcha control and then the AjaxAuthenticationCallback jscript method is called thus determining the result of the Captcha call.

* 1. Custom Ajax Validation

For Custom Ajax validation the developer must actually bind the event up for ajax validation passing any additional parameters to a separate page to validate whether the Captcha is valid and then return a json object to the callback as follows:

<atlantis:CaptchaTemplateControl ID="CaptchaTemplateControl2" runat="server" CaptchaID="instance4" SaveValidationDataInSession="true" CaptchaImagePlaceHolderID="plcCaptchaLocation4" ReloadLinkID="change\_code\_instance4" PlaySoundLinkID="speak\_code\_instance4" StashRenderLocation="javascriptStash" CaptchaValueID="cart\_captcha\_value\_instance4" AutoClearInput="true" AutoFocusInput="true" >

<Template>

<div style="width: 500px;">

<div style="float: left; padding-bottom: 10px;">

<asp:PlaceHolder ID="plcCaptchaLocation4" runat="server"></asp:PlaceHolder>

</div>

<div style="padding-left: 10px; float: left;">

<div class="captcha\_button" style="padding-bottom: 5px;" id="speak\_code\_instance4">

<div class="g-btn-lg g-btn-prg" style="width: 100px;">Speak Code</div>

</div>

<div class="captcha\_button" id="change\_code\_instance4">

<div class="g-btn-lg g-btn-prg" style="width: 100px;">Change Code</div>

</div>

</div>

<div style="clear: both;">

<asp:PlaceHolder ID="plcCaptchaLable" runat="server">

<div style="padding-top: 15px;" class="cboth">

<strong>Add Captcha here</strong>

</div>

</asp:PlaceHolder>

<input type="text" name="cart\_captcha\_value\_instance4" id="cart\_captcha\_value\_instance4" maxlength="90" style="width: 180px;" />

</div>

</div>

</Template>

</atlantis:CaptchaTemplateControl>

<div class="g-buttonpane">

<span id="validateCaptcha4" class="g-btn-lg g-btn-prg">Validate Captcha</span>

</div>

$(document).ready(function () {

$('#validateCaptcha4').bind('click.captcha', function () {

var captchaValue = $('#cart\_captcha\_value\_instance4').val();

$.ajax({

dataType: "json",

url: '<%=ValidateCaptchaURL%>',

data: {

captchaValue: captchaValue,

instanceID: 'instance4'

},

success: isValid2

});

});

});

function isValid2(result) {

alert(result.Valid);

if (!result.Valid) {

$('#cart\_captcha\_value\_instance4').trigger('captchaReload');

}

}

The ajax page should do the following call to validate the Captcha:

string captchaValue = Request.QueryString["captchaValue"] as string;

string instanceID = Request.QueryString["instanceID"] as string;

bool isValid = CaptchaTemplateControl.IsCaptchaValid(captchaValue, instanceID);