Blackinton Design-A-Badge IFRAME/Email Integration

Overview

This document contains instructions for integrating the Blackinton Design-A-Badge web technology into a dealer website. Integration is facilitated using the Blackinton Design-A-Badge Integration Extensions with IFRAME user interface integration and Email data return.

User Interface Integration Method
IFRAME

Data Return Method
Email

Design-A-Badge Process "at end" Action
HTTP Redirection

It is important to note that Design-A-Badge Integration Extensions has a small server-side footprint. The server-side code, which will be contained within the "dealer-site", is offered in Classic ASP, ASP.NET (C#), and PHP (release 5.2.0).

Download all code needed for this integration using the link below.

http://www.blackinton.com/integrate/support/support.zip

Client Side JavaScript

Supporting client side JavaScript code can be found within a file named "head.js". This code must be copied from the file and placed directly into the page that will host the <IFRAME>. The JavaScript code contains remote scripting tags, so it cannot be placed in a remote script tag itself. You must cut and paste this code in to the hosting page.

Minor modifications must be made to the code after it is copied into the hosting page in preparation for use. These changes are summarized below.

The lines of code that must be changed are found at the top of the "head.js" file. It is also important that only the values indicated below are modified. When making the changes read the instructions closely, incorrect modification may result in runtime script errors and could lead to unexpected results.

dealerId:

This variable must contain your valid dealer identification number. This is a value provided to you previously by Blackinton and is the same number used to access the dealer portal.

var dealerId = '44453XX';

_emailTo:

This variable contains the email addresses that will be used to return the data related to the design a web visitor creates using the Design-A-Badge web application. Multiple email addresses must be separated by a semicolon.

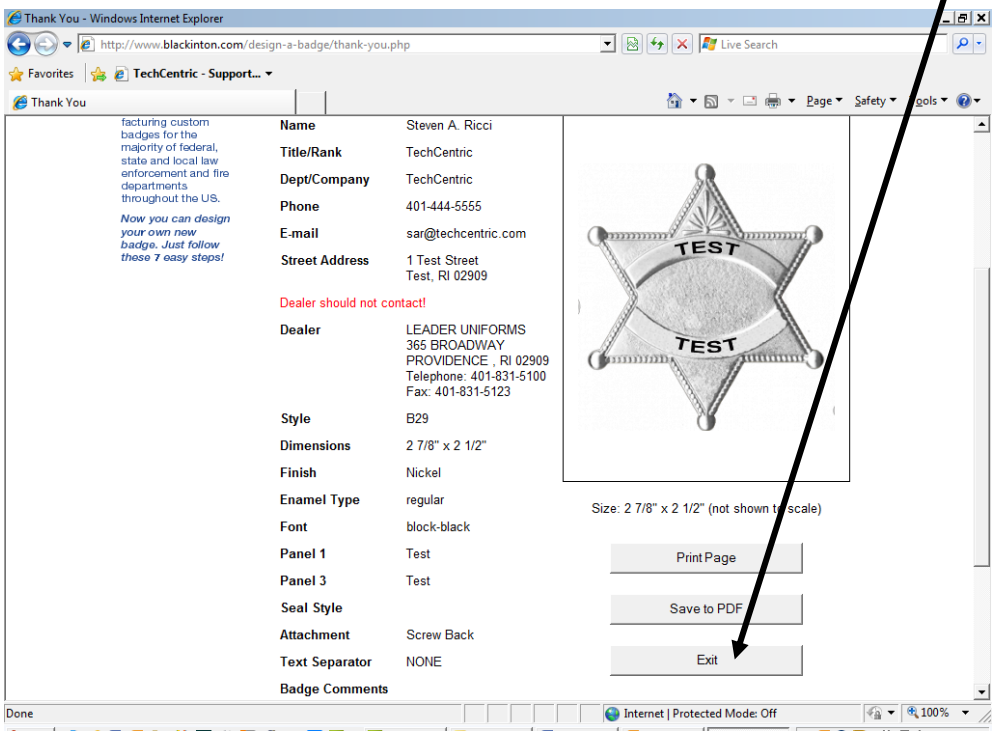
This field also has special usage when used in conjunction with feature switches (see below).

var _emailTo = 'user@domain.com;user2@domain.com';

_buttonText:

This variable contains the value that will be placed on the **BUTTON FACE**.

var _buttonText = 'Exit';



_switches:

var _switches = '10000000'

This variable works in a binary fashion (feature dip-switch), where each position within the string must contain a '1' or a '0'. Each position within the variable corresponds to a specification application feature that can be turned on or off.

A value of '1' indicates that the feature represented by that character position is ENABLED or ON. Conversely, a value of '0' indicates that the feature is DISABLED or OFF.

The feature/character position table below summarizes the presently available features.

POSITION	FEATURE
1	**Pricing.** **This feature will render the pricing section at the bottom of the design canvas on the "design-a-badge.php" page. Pricing is based on Blackinton suggested retail pricing. The pricing is controlled by Blackinton.** **0=on screen price is not enabled, 1=on screen pricing is enabled**
2	**Rendering Footprint.** **This feature will determine if surrounding content will be shown or suppressed. Content would include the header and left region that surround the core interactive portion of Design-A-Badge interface.** **0=surrounding content is enable, 1=disabled**
3	**HTTP POST Data Format.** **This feature will determine how the data posted by the DAB service module will be returned to the dealer website. This does not affect the HTTP post, just the contents of the POST.** **This will also shorten the steps of the entire Design-A-Badge process to 3. There is no contact information step or confirmation step. This is best used when a shopping cart integration feel is desired.** **1=Conventional POST. All fields, including the Transaction Number are individual HTML encoded fields.**

4	**QTY Field will be rendered on the Design Canvas when this switch is set to "1".** **This feature will render at QTY field on the Design Canvas, enabling the web visitor to specify how a QTY. If this switch is set to "0", then a QTY of 1 is transmitted.** **1=QTY Field is rendered.**
5	**Make "Address" field on the Contact Information page a required field. When set to "1", this field cannot be blank. By default the "Address" field is not required.**
6	**Reply-To Address on emails.** **When set to "1", the REPLY-TO address field of the email sent to the customer will be the value of the _emailTo field (see above), and the REPLY-TO address field of the email sent to the dealer will be the value of the email address specified by the customer on the Contact Information step.** **The FROM address on emails sent from Blackinton will be:** ***Design-A-Badge@blackinton.com*** **When set to "0", the FROM address will be:** **noreply@blackinton.com**
7	**Reserved for future features, should be set to 0**
8	**Reserved for future features, should be set to 0**

dealerURL:

This variable contains the filename and path of the required server side code (see section *server side code* below). This file will be utilized to handle redirection of the website visitor at the end of the Design-A-Badge process. The only modifications to this variable would be for "server side technology" and "path" related.

For Classic ASP, the filename should be "redirect.asp". For ASP.NET, the filename should be "redirect.aspx". For PHP, the filename should be "redirect.php".

The location the redirect server file is placed on the dealer site must be reflected within the ***dealerURL***.

The example below would be for a Classic ASP website and would require the placement of the file 'redirect.asp' at the same level of the hosting page (the page that will host the Design-A-Badge <IFRAME>).

The getClientURL() is a utility function that determines the base URL automatically.

```
var dealerURL = getClientURL()+'redirect.asp';
```

The ***dealerURL*** could also contain a full URL without using the utility function.

```
var dealerURL = "www.dealersite.com/dab/redirect.asp';
```

<IFRAME> HTML

The coding segment below must be placed within the hosting page along with the code from the "head.js" file (see above). This coding segment should be placed within the <BODY> section of the hosting page in the location you would like the Design-A-Badge web application to render. The screen footprint for the Design-A-Badge web application is approximately 800px.

Copy and paste the code below, do not try to type it visually, as you may make a typographical error.

```
<iframe scrolling='auto' id='dabframe' frameborder='0' width="100%" height="100%">
  <p>This browser does not support the HTML element IFRAME.  Please obtain a browser that is compatible.</p>
</iframe>
```

This code listed above can be found within the file body.htm, which is included in the supporting file download (see overview section above).

Server Side Code

Supporting server side code is contained within the ZIP file which can be downloaded using the link provided within the overview section of this document.

After selecting the appropriate supporting files based on your web hosting platform, the supporting file(s) must be placed within your dealer site in a location that is accessible from the Internet. The supporting file(s) must be able to accept an HTTP POST from the Internet.

The file is offered in three server technologies:

- Classic ASP,
- ASP.NET (C#)
- PHP.

The name of the file, *redirect.asp*, *redirect.aspx*, or *redirect.php* contains a single line of server side code. The example below is shown using Classic ASP. The line of code redirects the web visitor's browser to the URL you provide. This redirection will occur when the visitor clicks the "EXIT" button on the last step of the Design-A-Badge process.

In this example, redirection would be to ***www.google.com***.

```
<%
response.redirect("http://www.google.com")
%>
```

Additional Support

For additional support regarding this type of integration as well as other integration options, please email support@techcentric.com.