



Passing User Attributes to WCO

Integrating with Janrain

Version 0.90

© 2014 CrownPeak Technology, Inc. All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission from CrownPeak Technology.

Document History

Author/Editor	Date	Reason for Change	Version
Fahd Shaaban	8/28/2014	Draft	0.90

Table of Contents

Overview	4
WCO Parameters	5
Benefits of passing parameters to WCO	5
Configuring this WCO Feature.....	6
Configuring WCO to integrate with a 3rd Party Provider.....	7
Janrain integration	7
Retrieving Janrain attributes	7
Passing attributes to WCO	9
Putting it all together.....	11
Notes and Additional Information	12
Implementation Guidelines.....	12
Playbook Notes	12

Overview

CrownPeak Web Content Optimizer provides a number of content optimization capabilities where various types of user attributes can be used to define audiences, segments or target groups and deliver targeted content to these groups of users.

There are four types of user attributes:

- **Ambient:** These attributes are readily available and nothing is required from the user to provide these attributes. They include time of day, date, the referrer, search engine keyword, geo-location, user device, browser type, and browser user agent.
- **Form data:** These attributes are form fields the user provides by voluntarily filling out forms and submitting them. These forms include contact us, feedback, survey, fill out this form to download a white paper, event registration, etc.
- **Behavior data:** These attributes describe the user's interest in site content by looking at visits (views) and clicks (conversions). WCO natively tracks snippet views and conversions, and this data can be used instantly for targeting purposes.
- **Integrations:** These user attributes are typically retrieved via API calls from 3rd party providers, which may include Marketing Automation Platforms such as Eloqua and Marketo, CRMs such as Salesforce and Dynamics, and Identity providers as such Janrain. However, this requires the development of a WCO connector that understands the 3rd Party Service's APIs, which may require time and effort not suitable to a specific project.

It is common for customers to inquire about an integration with 3rd Party services to provide data to WCO for targeting purposes. If a WCO connector exists, then what remains is configuring the connector to be used by Customer for targeting and personalization purposes.

However, if a connector doesn't yet exist, the customer will request one. Depending on current projects and the roadmap, building a connector may not be suitable as it may not be available in time for the customer's project.

An alternative to WCO Connectors, WCO has the ability to accept any user attributes at runtime simply by passing the attributes to WCO. These attributes are processed in real-time and the appropriate content is delivered instantly to the user based on their attributes. These attributes can be anything and can come from anywhere.

This document describes this unique WCO feature and uses Janrain as an integration example.

CrownPeak subscribers should consider leveraging this WCO feature to take advantage of the following capabilities:

- Retrieve user attributes from any provider and pass them to WCO
- Configure WCO Target Groups using these passed attributes or parameters
- Deliver targeted content based on these attributes

WCO Parameters

The ability to pass parameters to WCO is built on WCO's ability to process forms.

Forms are typically processed using an HTTP POST where the form fields are passed in the header request to WCO.

Forms can also be submitted using GET where the form fields are concatenated into a query string and passed in the URL, as follows:

```
http://www.url.com/process?field1=value1&field2=value2
```

WCO takes advantage of this HTTP protocol, and allows sites to instantly pass form fields without or instead of **posting** them.

The typical way to display a WCO snippet is as follows:

```
<script type="text/javascript" src="//snippet.omm.crownpeak.com/s/2c4764a1-b91d-4d37-9aeb-eaabecfa2fc2" ></script>
```

Since the src of the above script tag is a URL, WCO has been configured to accept querystring parameters passed along with the URL and treat these parameters as form fields.

```
<script type="text/javascript" src="//snippet.omm.crownpeak.com/s/2c4764a1-b91d-4d37-9aeb-eaabecfa2fc2?beacon=pension" ></script>
```

It is important to note that unlike normal form submissions which use POST and result in a WCO Conversion, passing form fields in QueryString does not result in a conversion.

Benefits of passing parameters to WCO

The ability to pass parameters to WCO extends its target group configuration options by allowing WCO to accept user attributes from any source:

- Ability to pass one or more parameters to WCO using QueryString approach
- The source of the parameters is irrelevant to WCO
- WCO will accept these parameters and use them instantly for its Target Group criteria
- Passing parameters doesn't skew the conversion metrics since they're not tracked as conversions

Configuring this WCO Feature

The ability to pass user attributes to WCO can be configured to support many business processes. Before starting the configuration, it is beneficial to solidify the scope of the configuration. Use this list to determine how the customer is going to use this capability on their site.

- Does the customer want to target users based on 3rd Party user information?
- Does CrownPeak currently offer a connector for the 3rd Party provider? If not, how long will it take for a connector to be developed?
- Does the 3rd Party provide offer API's to access the required data? Can the 3rd Party provide code samples or documentation?

Configuring WCO to integrate with a 3rd Party Provider

Using Janrain, the following describes how WCO can be configured to integrate with a 3rd Party provider.

Janrain integration

Janrain is an identity provider that offers their customers user management services such as registration, login, forgot password, and profile management.

Integrating Janrain in a CrownPeak site is very simple and easy since the Janrain service is Javascript based, and Janrain provides all the necessary code for users to register, login, edit their profile, or request a new password. There is nothing special that is required within the CrownPeak CMS to enable this integration.

However, optimizing site content using WCO based on Janrain attributes does require an integration, and for the purpose of this document, we will analyze how Janrain attributes can be retrieved and passed to WCO for instant personalization purposes.

Retrieving Janrain attributes

Janrain's services are accessed using client-side javascript. Upon successfully logging in, Janrain exposes a unique `accessToken` that is used to identify the logged in user.

Note: All javascript code examples below utilize the JQuery javascript framework.

Retrieving the `accessToken` is done as follows:

```
janrain.events.onCaptureSessionFound.addHandler(function (result) {  
    alert(result.accessToken)  
});
```

The `accessToken` is a unique identifier that doesn't mean anything to WCO. Janrain provides API calls to retrieve janrain user profile data if an `accessToken` is provided. However, this API call can not be performed client-side due to security restrictions. It must be called server-side.

The easiest way to make this work with WCO is to create a simple server-side API-bridge that accepts the `accessToken` as a querystring parameter, makes the API call to the 3rd Party provider, in this case Janrain, and then returns the data in JSON back to the browser to pass to WCO.

The following is an example of a .Net page that acts as an API-Bridge:

```
<${@ Page Language="C#" $>
<script runat="server">
protected void Page_Load(object sender, EventArgs e)
{
    try
    {
        string szToken = (Request.QueryString["token"] ?? "").ToString();

        System.Net.WebClient wcData = new System.Net.WebClient();
        byte[] bDataBuffer =
wcData.DownloadData("https://customer.janraincapture.com/entity?type_na
me=user&access_token="+szToken);
        string szReturnValue = Encoding.ASCII.GetString(bDataBuffer);
        Response.Write(szReturnValue.Trim().Replace("\r\n",
"").Replace("\n", ""));
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
</script>
```

Making a call to the api-bridge, and passing the accessToken will return a JSON object.

The next step is to parse the JSON object for the user attributes that needed to be targeted.

```
janrain.events.onCaptureSessionFound.addHandler(function (result) {
    var url = "/api-bridge.aspx?token="+result.accessToken;
    $.getJSON(url,function(data){
        alert(data.result.professionalData.professionalGroup);
    });
});
```

The above snippet of code will call the API-Bridge server-side page with the accessToken as a querystring parameter, and parses the returned JSON for the value of the professionalGroup user attribute.

Now that we have the user attribute, all that's left to do is to pass it to WCO.

Passing attributes to WCO

```

janrain.events.onCaptureSessionFound.addHandler(function (result) {
    var url = "/api-bridge.aspx?token="+result.accessToken;
    $.getJSON(url,function(data){
        $('<script type="text/javascript"
src="http://snippet.omm.crownpeak.com/s/deeed5a2-690d-48a5-bf0d-
4bleee9ba38f/2?role='+data.result.professionalData.professionalGroup+' "
>' + '</sc' + 'ript>').appendTo('head');
    });
});

```

WCO is engaged using javascript on the client-side. The snippet javascript code is automatically generated by the CMS for every CMS control (WYSIWYG or TextBox) that is WCO enabled.

The CrownPeak Product team are always working on improving WCO and enhancing it with new capabilities, and will soon provide a CMS user experience dedicated to provide an intuitive and easy way to define and pass parameters to WCO snippets.

In the meantime, in order to take advantage of WCO's ability to accept form fields passed in the QueryString, the Output template will typically need to be modified to pass these parameters to the WCO snippet.

There are two ways to pass parameters to WCO:

Server-Side

In this case, the querystring (form fields and values) is constructed server-side and is simply used when the page loads in the browser

```

<script type="text/javascript"
src="http://snippet.omm.crownpeak.com/s/deeed5a2-690d-48a5-bf0d-
4bleee9ba38f/2?role=doctor"></script>

```

Client-Side

```

$('<script type="text/javascript"
src="http://snippet.omm.crownpeak.com/s/deeed5a2-690d-48a5-bf0d-
4bleee9ba38f?role='+data.result.professionalData.professionalGroup+' ">'
+ '</sc' + 'ript>').appendTo('head');

```

Since the Janrain accessToken, in our integration example, is exposed client-side, we can't use the Server-side approach to pass user attributes to WCO. We have to use the client-side approach.

However, a complication arises with this approach. Since the script tag can't be added to the page until the the accessToken and user attributes are made available, browsers will not process the script tag due to security restrictions. In other words, the client-side code example above will not work because the browser will not allow an external platform like WCO to manipulate the content of the webpage using javascript after it has already loaded.

The way around this is to use JSONP, which gives control back to the page by having WCO call a javascript function on the page to perform the page manipulation. WCO supports this by adding /2 to the URL as follows

```
$( '<script type="text/javascript"
src="http://snippet.omm.crownpeak.com/s/deeed5a2-690d-48a5-bf0d-
4bleee9ba38f/2?role='+data.result.professionalData.professionalGroup+' "
>' + '</sc' + 'ript>' ).appendTo('head');
```

The /2 is parameter that instructs WCO to call a callback function, called wcoCallback:

```
function wcoCallback(snippetId, data){
    $('#' + snippetId).html(data);
}
```

The above function will be called when the WCO snippet is executed, which will also pass the returned WCO content in the data variable. The function will then load the data in the body of a div with an id of snippet.

```
<div id="deeed5a2-690d-48a5-bf0d-4bleee9ba38f"></div>
```

Putting it all together

```

<html>
<head>
<script type="text/javascript">
    function wcoCallback(snippetId, data){
        $('#' + snippetId).html(data);
    }

    janrain.settings.capture.flowName = 'customer_flow';
    // --- define event handlers and start the janrain ui -----
    -----
    function janrainCaptureWidgetOnLoad() {
        janrain.events.onCaptureSessionFound.addHandler(function
(result) {
            var url = "/api-
bridge.aspx?token="+result.accessToken;
            $.getJSON(url,function(data){
                $('<script type="text/javascript"
src="http://snippet.omm.crownpeak.com/s/deeed5a2-
690d-48a5-bf0d-
4bleee9ba38f/2?role='+data.result.professionalData.pr
ofessionalGroup+'>' + '</sc' +
'ript>').appendTo('head');
            });
        });
    };
</script>
</head>
<body>
    <div id="deeed5a2-690d-48a5-bf0d-4bleee9ba38f"></div>
</body>
</html>

```

Notes and Additional Information

More information can be found in

<https://connect.crownpeak.com/blogs/developers/product-update-connectors>

Implementation Guidelines

In general, this integration can be configured 16 - 24 hours depending on requirements and complexity.

Playbook Notes

<https://connect.crownpeak.com>