Oracle WebCenter Sites

Analytics & SEO Configuration

Technical Notes

Pazhanikanthan Periasamy

ORACLE WebCenter Presales Team

Sep 2015

ORACLE®

TABLE OF CONTENTS

REFERENCE DOCUMENT

Name	File Name	Date
Scribe Guide	Scribe Developer Guide March 2013.pdf	March 2013

GLOSSARY

Term	Description	Examples
$WCS_INSTALL_DIR	WebCenter Sites Installation directory	For QSK 2.3.6, **D:\APPS\QuickStart_2.3.6**
$WCS_WEBAPP_DIR	WebCenter Sites Web application directory	For QSK 2.3.6, **D:\APPS\QuickStart_2.3.6\tomcat7034\webapps**
$WCS_CS_DIR	WebCenter Sites 'cs' Web application directory	For QSK 2.3.6, **D:\APPS\QuickStart_2.3.6\tomcat7034\webapps\cs**
$SW_ROOT_DIR	This is the directory where libraries and other assets are available for feature installation / configuration	For example, if you extract "Oracle WebCenter Sites - Analytics & SEO Configuration.zip" under D:\APPS, then this would be **D:\APPS\Oracle WebCenter Sites - Analytics & SEO Configuration**

Overview

Integrating Oracle WebCenter Sites with:

➢ Search Engine Optimization - Scribe SEO (http://scribecontent.com/)
➢ Web Analytics - Google Analytics(www.google.com/analytics/)

Oracle WebCenter Sites

- Integrate into Oracle WebCenter Sites QSK 2.3.6 (Quick Start Kit)

Search Engine Optimization

Scribe SEO API provides a robust set of features for content writers to research, optimize, and socialize their content online. The heart of Scribe is the API. This document details the different REST API requests with results returned in JSON format.

Note: During this module development, I faced issues using JQuery JavaScript AJAX calls to send JSON request via HTTP POST method. So I switched to implementing the module in Java.

Solution overview looks like:
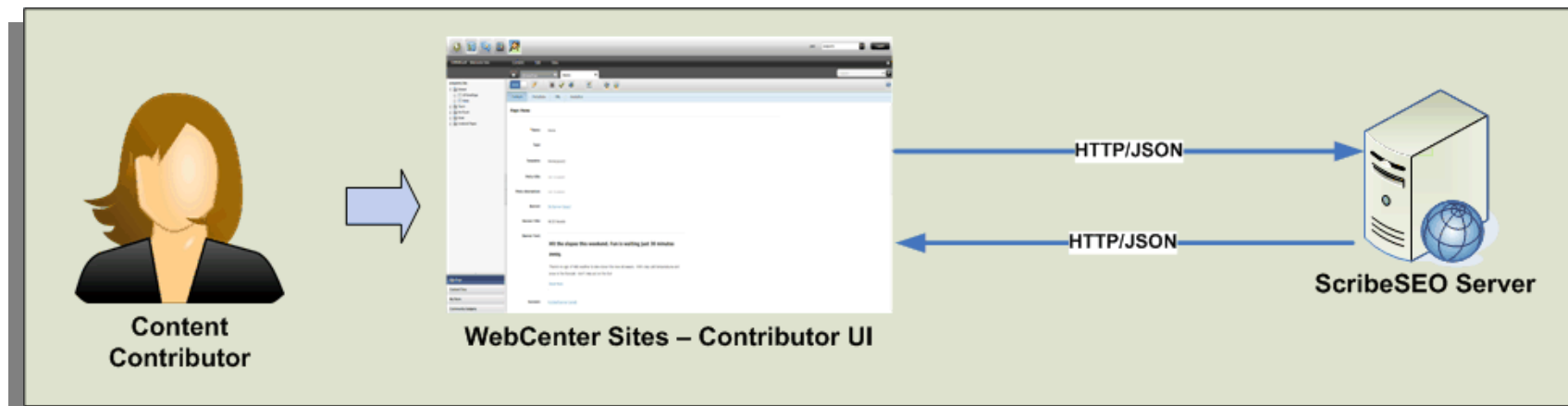
A typical Scribe SEO Request and Response would require:

Feature	Value
URL	http://api.scribeseo.com/analysis/kw/suggestions
Request Parameter [apikey]	scribe-261ad984da6a44fcb6746759be7f0cce
Headers	Content-type: application/json
Method	POST
Input	{ JSON REQUEST STRING }
Response	{ JSON RESPONSE STRING}

Contributor UI Integration

Search Engine Optimisation integration into WebCenter Sites is at an Asset level (Article, Page, etc). See screenshot below (Page Asset in Form View).
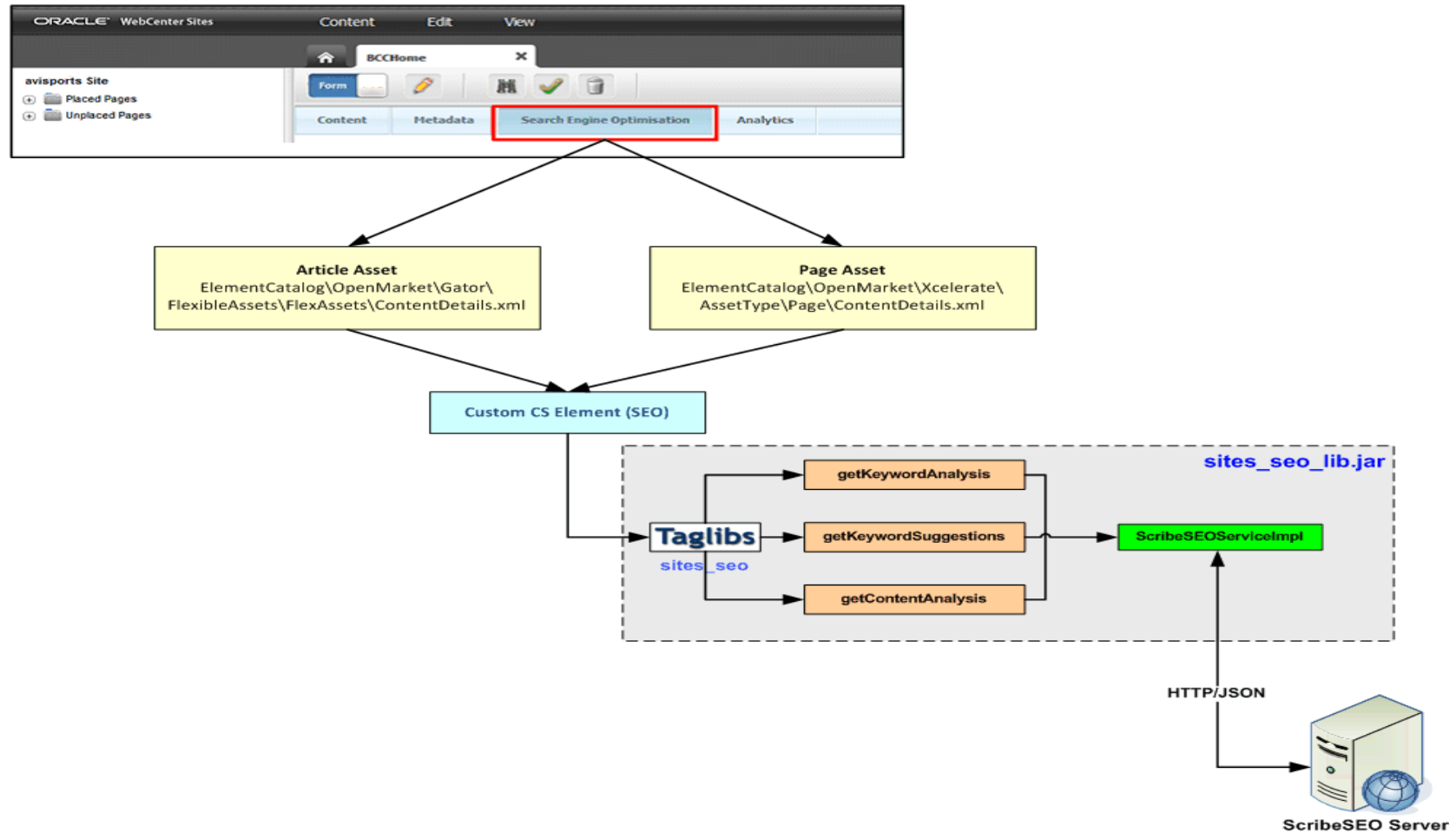

Currently three SEO services have been integrated:

- ➢ Keyword Suggestions

- ➢ Keyword Details

- ➢ Content Analysis

Technical Integration Overview

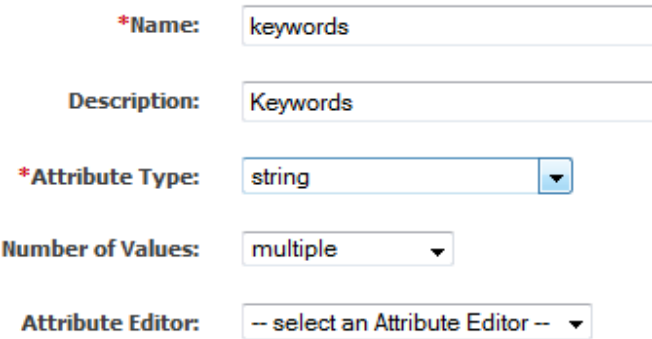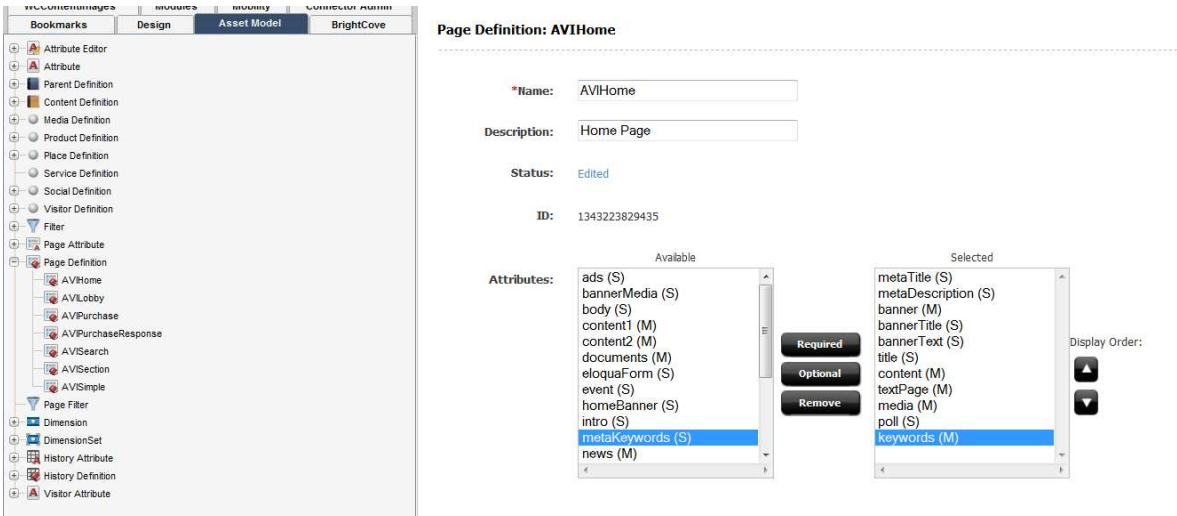Technical implementation overview is as below:

Scribe SEO expects metadata attribute values to be passed into the service calls. Current implementation expects the following attributes to be passed in as parameters. This implementation can be changed by modifying the CSElement **"$SW_ROOT_DIR/CustomElement/features/SEO"**.
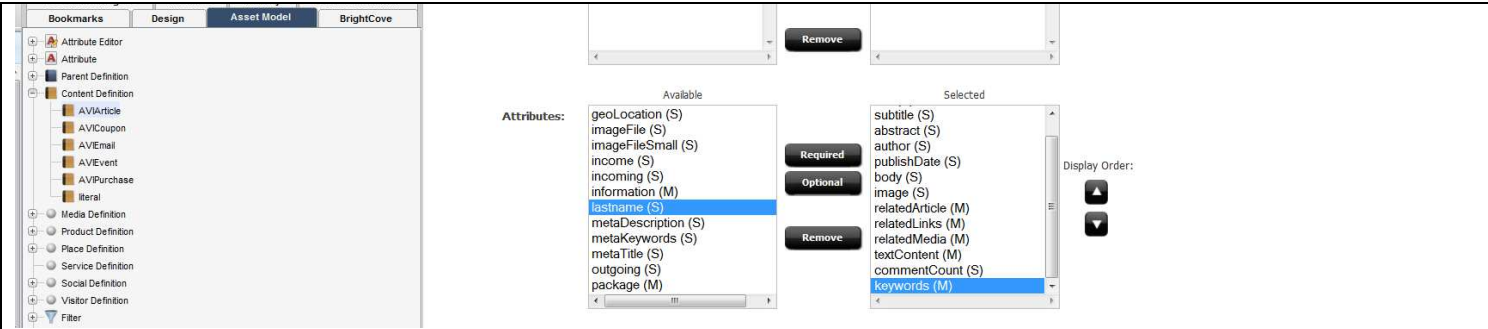
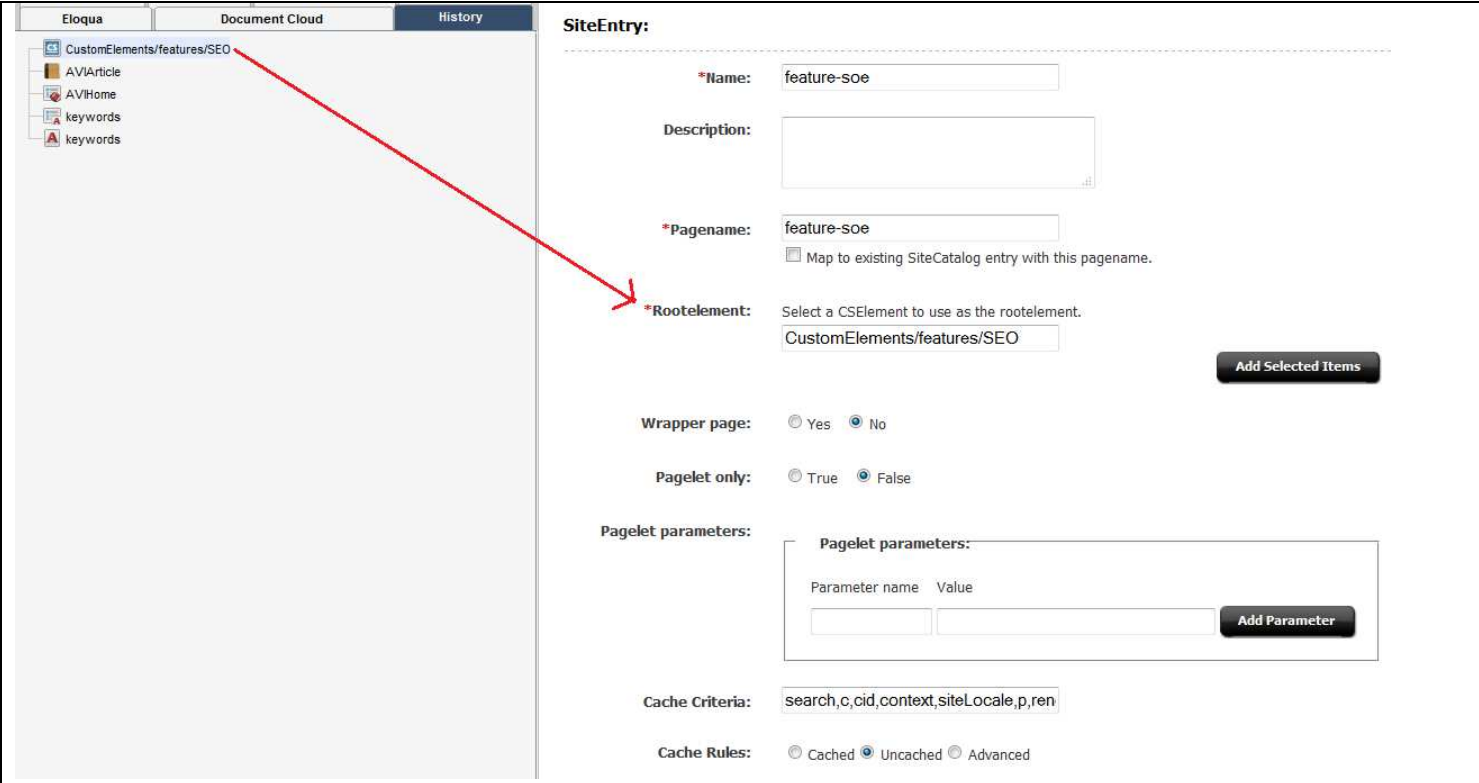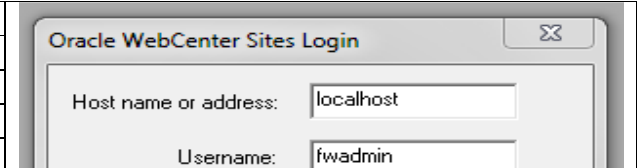| Sl. No.	Scribe SEO Service	Asset Attribute (PageAttribute)	Asset Attribute (ContentAttribute)
1.	getKeywordAnalysis	keywords (multi-valued)	keywords (multi-valued)
2.	getKeywordSuggestions	keywords (multi-valued)	keywords (multi-valued)
3.	getContentAnalysis	➢ metaTitle ➢ metaDescription ➢ title ➢ metaTitle	➢ headline ➢ subheadline ➢ headline ➢ abstract

Integration Steps
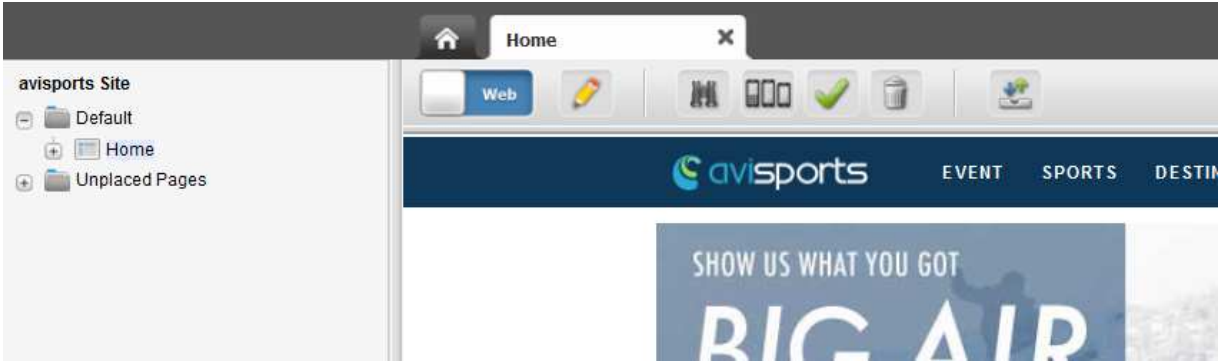
STEP 1	Shutdown Oracle WebCenter Sites application server (if running).
STEP 2	Extract the software package "**Oracle WebCenter Sites - Analytics & SEO Configuration.zip**" as **$SW_ROOT_DIR.**
STEP 3	Copy "**$SW_ROOT_DIR/library/SEO/sites_seo_lib.jar**" & "**$SW_ROOT_DIR/library/SEO/oscache-2.4.jar**" into WebCenter Sites application installation "**$WCS_CS_DIR/WEB-INF/lib**".
STEP 4	Copy the folder "**$SW_ROOT_DIR/web_resources/SEO/reports**" into WebCenter Sites application installation "**$WCS_CS_DIR** ".
STEP 5	Copy file "**$SW_ROOT_DIR/config/SEO/spring_seo_config.xml**" into WebCenter Sites application installation "**$WCS_CS_DIR/WEB-INF**"
STEP 6	Modify Spring application Context XML file "**$WCS_CS_DIR/WEB-INF/applicationContext.xml**" and import the SEO specific config entry. \<?xml version="1.0" encoding="UTF-8"?> \<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd"> \<beans> \<import resource="spring_seo_config.xml" /> \</beans>
STEP 7	This step is required if you need to route outbound calls via a web proxy in your environment. Ensure HTTP / HTTPS Proxy is added as a startup parameters JAVA_OPTS="$JAVA_OPTS -Dhttp.proxyHost=HTTPProxyServerHost -Dhttp.proxyPort=HTTPProxyServerPort -Dhttps.proxyHost=HTTPSProxyServerHost -Dhttps.proxyPort=HTTPSProxyServerPort -Dhttp.nonProxyHosts='localhost|127.0.0.1|10.*.*.*|*.company.com'"
STEP 8	Restart Oracle WebCenter Sites application Server.

STEP 9	To validate that the libraries have loaded open "**$WCS_INSTALL_DIR/sites/logs/sites.log**" and search for keyword "spring_seo_config.xml". You will see 2 lines
	
	`[beans.factory.xml.XmlBeanDefinitionReader] Loading XML bean definitions from ServletContext resource [/WEB-INF/spring_seo_config.xml]`
	`…`
	`Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@8e4cf51: defining beans [seo.context.aware,services.seo.scribe]; root of factory hierarchy`

STEP 10	Open WebCenter Site "avisports" in Administrator mode. Create a "Page Attribute" and "Attribute" named as "keywords"
	 *Name: `keywords` Description: `Keywords` *Attribute Type: `string` ▼ Number of Values: `multiple` ▼ Attribute Editor: `-- select an Attribute Editor --` ▼
STEP 11	Open the Asset Model tab. Expand "Page Definition" and add the "Page Attribute" to Page Definitions "AVIHome", "AVISection", "AVISimple" as an optional field. Expand "Content Definition" and add the "Attribute" to Definitions "AVIArticle" or "Article" as an optional field.

STEP 12	Open WebCenter Site "avisports" in Administrator mode. Create a CSElement as "JSP" named "**CustomElements/features/SEO**". Copy and paste the code from the attached file ""**$SW_ROOT_DIR/ CustomElements/features/SEO.jsp**". Save changes.

	CSElement: CustomElements/features/SEO
	*Name: CustomElements/features/SEO
	Description:
	Status: Created
	ID: 1343225904885
	Rootelement: CustomElements/features/SEO
	ElementCatalog Description:
	Element Storage Path/Filename: CustomElements\features\SEO.jsp
	Element Parameters: eid=1343225904885
	Additional Element Parameters:
	Element Logic:
	`<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>` `<%@taglib prefix="oscache" uri="http://www.opensymphony.com/oscache" %>`
STEP 13	Create a Site Entry named "feature-soe" by selecting the SEO CSElement created above. Ensure that Cache Rule is "Uncached" is selected.

STEP 14	Open Sites Content Explorer.
STEP 15	Login using the following details.

Login using the following details.

Host name or address:	localhost
Username:	fwadmin
Password:	xceladmin
Protocol:	HTTP
Port:	8080

	Application Server URL Path:	CS Servlet (/cs/CatalogManager)	
STEP 16	Navigate and open "ElementCatalog\OpenMarket\Gator\FlexibleAssets\FlexAssets\ContentDetails.xml" (For AVIArticle Assets) and "ElementCatalog\OpenMarket\Xcelerate\AssetType\Page\ContentDetails.xml" (For Page Assets).		
STEP 17	Add the following code to add a new SEO tab. Ensure that there are 3 ending </div> tags after the addition as shown in below.		

```
<!-- SEO CHANGES STARTS -->
<div dojoType="dijit.layout.ContentPane" title="Search Engine Optimisation" REPLACEALL="Variables.tabVanityUrl">
    <CALLELEMENT NAME="CustomElements/features/SEO">
        <ARGUMENT NAME="assetId" VALUE="Variables.id"/>
        <ARGUMENT NAME="assetType" VALUE="Variables.AssetType"/>
        <ARGUMENT NAME="embedInIframe" VALUE="true"/>
    </CALLELEMENT>
</div>
<!-- SEO CHANGES ENDS -->
...

    </div>
    </div>
    </div>
</FTCS>
```

STEP 18	Open WebCenter Site "avisports" in Contributor mode. Open Home Page from the left hand panel.

Edit the page, and toggle the view to Form View to add few keywords for the keywords field. Ensure to fill in the metaTitle and Meta Description fields.

Meta-Title: Sports Home

Meta-Description: Sports Home

Keywords:

Sports ✕

Oracle WebCenter Sites ✕

Add

SEO tab (**Keyword Analysis and Suggestions**) looks like:

STEP 19	SEO tab (**Content Analysis**) looks like: Click on the Keyword Analysis row value to view more suggestions.

Structural & Word Choices Score	very difficult

Keywords Analysis	home Score: 2.93 1 or more suggestion(s) available

| Item	Results
Keyword's score based on the writing style of the author	2.93
Keyword's density	133.33
Suggestions to increase Keyword's score	1. Keyword density is more than 10%
How many times this keyword appears in the user's content	4
Numeric value of Keyword's importance	Important
Grid score of the keyword	A
Text that is presented to the user in relation to the keyword's position in the grid	Congratulations, this term strikes a good balance between search optimization and copywriting best practices.
Prominence of the keyword	Very High
Rank of the keyword in relation to the others	1
Keyword's score based on the content	9.5
Keyword's important in the user's content	2

Google Analytics

Current implementation focuses on integrating WebCenter Sites with Google Analytics. Solution overview looks like:
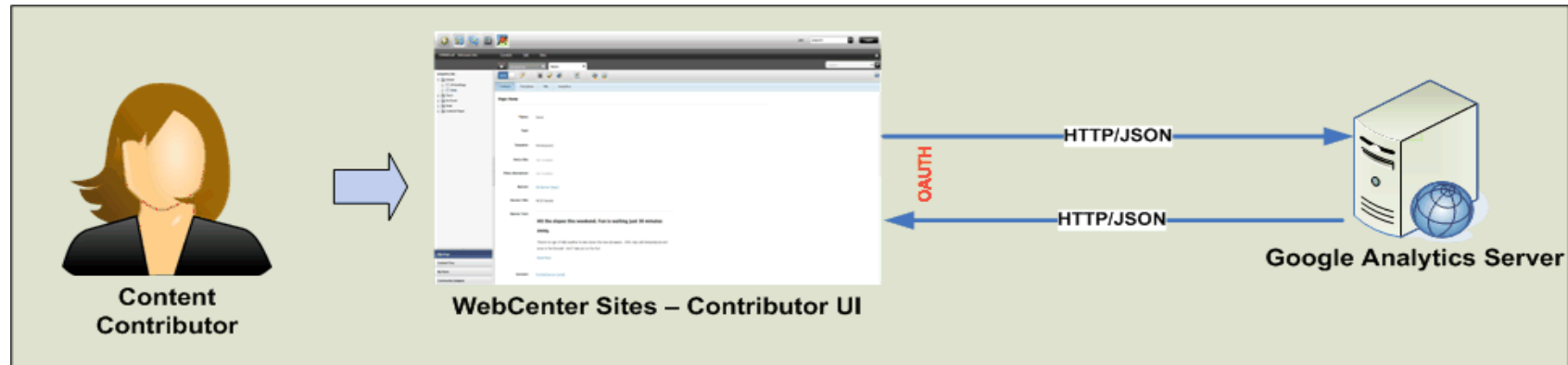
Contributor UI Integration

Google Analytics integration into WebCenter Sites is at an Asset level (Article, Page, etc). See screenshot below (Page Asset in Form View).
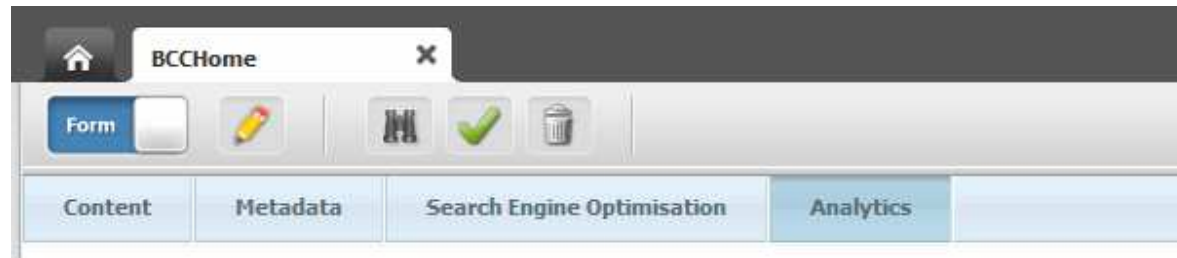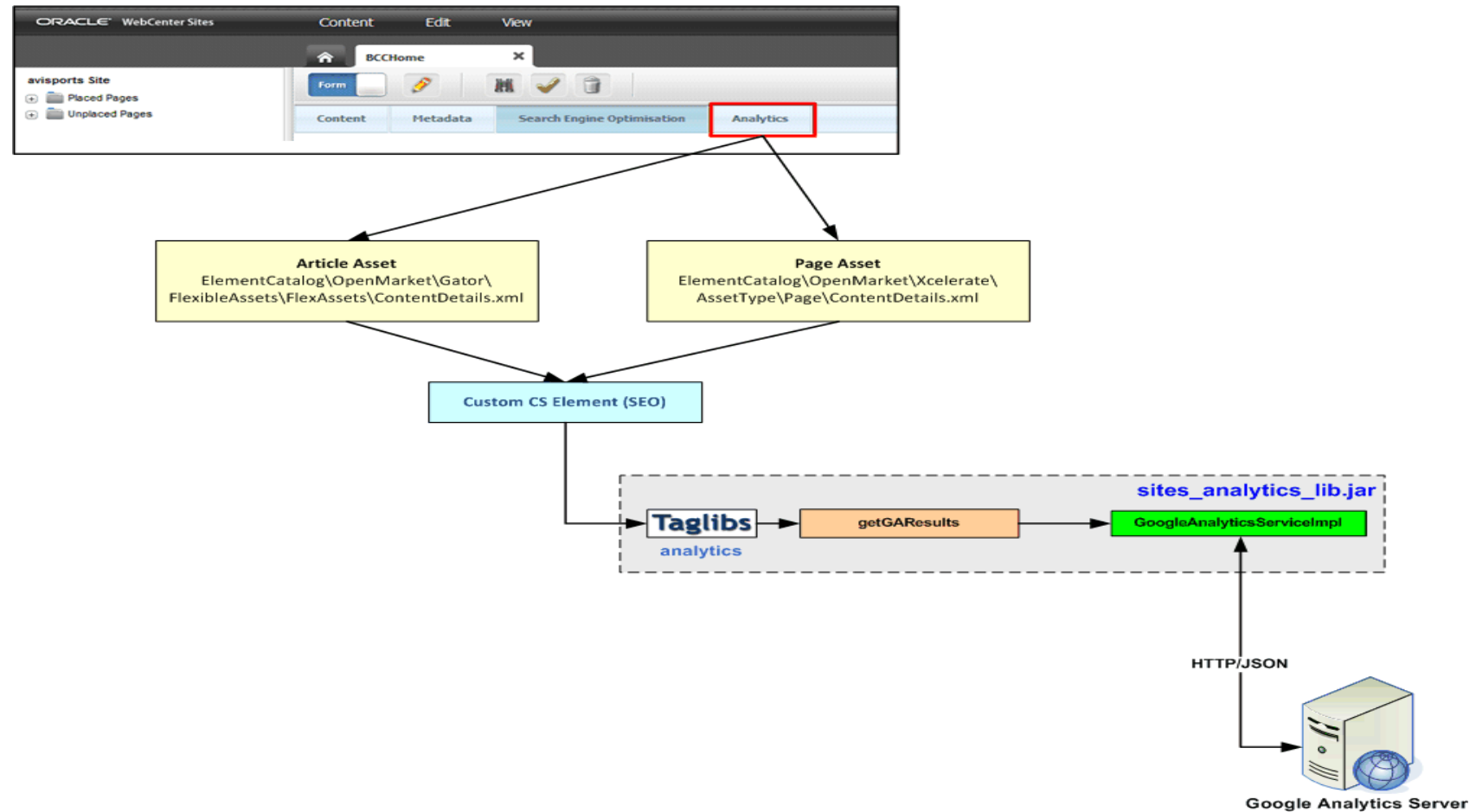

Google Analytics API provides wide range of Analytics data. Currently following Analytics report is displayed:

➢   Page Views by Week

Technical Integration Overview

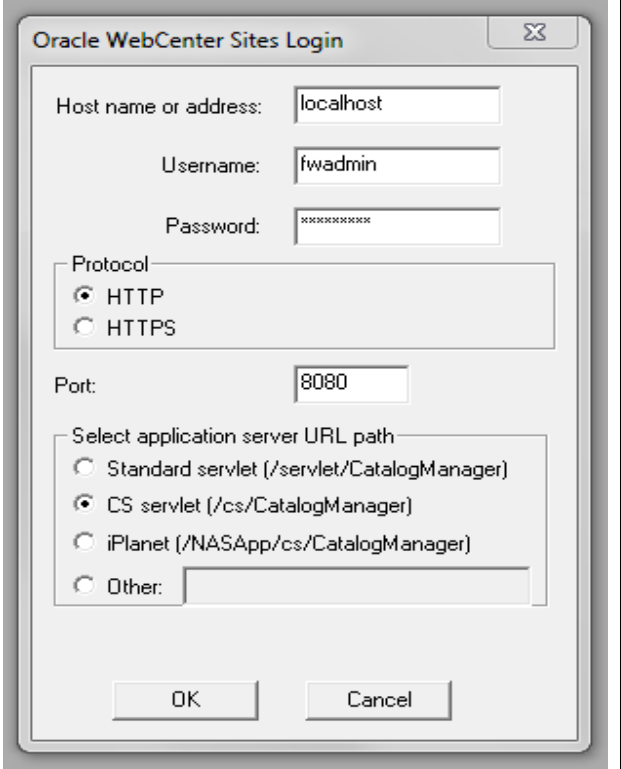Technical implementation overview is as below:

Integration Steps

STEP 1	Shutdown Oracle WebCenter Sites application server (if running).
STEP 2	Extract the software package "**Oracle WebCenter Sites - Analytics & SEO Configuration.zip**" as **$SW_ROOT_DIR.**
STEP 3	Copy all library jars from "**$SW_ROOT_DIR/library/GA**" into WebCenter Sites application installation "**$WCS_CS_DIR/WEB-INF/lib**".
STEP 4	Copy the folder "**$SW_ROOT_DIR/web_resources/GA/reports**" into WebCenter Sites application installation "**$WCS_CS_DIR** ". *Note: Some files may be over-ridden if you have already configured the system for SEO.*
STEP 5	Copy file "**$SW_ROOT_DIR/config/GA/spring_analytics_google_config.xml**" and "**$SW_ROOT_DIR/config/GA/serviceaccount-privatekey.p12**" into WebCenter Sites application installation "**$WCS_CS_DIR/WEB-INF**"
STEP 6	Modify Spring application Context XML file "**$WCS_CS_DIR/WEB-INF/applicationContext.xml**" and import the SEO specific config entry. <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd"> <beans> <import resource="spring_analytics_google_config.xml" /> </beans>

STEP 7	Open "**$WCS_CS_DIR/spring_analytics_google_config.xml**" and update the following according to your environment:
	
	<pre><bean id="service.account.priv.key" class="oracle.webcenter.sites.framework.analytics.model.google.ServiceAccountPrivateKey">
	  <constructor-arg value="$WCS_CS_DIR/WEB-INF/serviceaccount-privatekey.p12" type="java.lang.String" />
	</bean>
	
	<bean id="services.analytics.google"
	class="oracle.webcenter.sites.framework.analytics.services.impl.google.GoogleAnalyticsServiceImpl" init-method="initialise">
	  <property name="serviceAccountPrivateKey" ref="service.account.priv.key"/>
	  <property name="serviceAccountEmail" value="115627830109-
	u51ntr941ro3tkbuip2f3u85aqgbst68@developer.gserviceaccount.com"/>
	  <property name="accountName" value="JSK"/>
	</bean></pre>

STEP 8	This step is required if you need to route outbound calls via a web proxy in your environment. Ensure HTTP / HTTPS Proxy is added as a startup parameters JAVA_OPTS="$JAVA_OPTS -Dhttp.proxyHost=HTTPProxyServerHost -Dhttp.proxyPort=HTTPProxyServerPort -Dhttps.proxyHost=HTTPSProxyServerHost -Dhttps.proxyPort=HTTPSProxyServerPort -Dhttp.nonProxyHosts='localhost|127.0.0.1|10.*.*.*|*.company.com'"
STEP 9	Restart Oracle WebCenter Sites Application Server.
STEP 10	To validate that the libraries have loaded open "**$WCS_INSTALL_DIR/sites/logs/sites.log**" and search for keyword "spring_analytics_google_config.xml". You will see 2 lines **[beans.factory.xml.XmlBeanDefinitionReader] Loading XML bean definitions from ServletContext resource [/WEB-INF/spring_analytics_google_config.xml]** **…** **Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@6dac1cb9: defining beans [seo.context.aware,services.seo.scribe,google.analytics.aware,service.account.priv.key,services.analytics.google]; root of factory hierarchy**
STEP 11	Open WebCenter Site "avisports" in Administrator mode. Create a CSElement as "JSP" named "**CustomElements/features/GA**". Copy and paste the code from the attached file ""**$SW_ROOT_DIR/CustomElements/features/GA.jsp**". Save changes.

	CSElement: CustomElements/features/GA
	*Name: CustomElements/features/GA
	Description:
	Status: Created
	ID: 1343225905480
	Rootelement: CustomElements/features/GA
	ElementCatalog Description:
	Element Storage Path/Filename: CustomElements\features\GA.jsp
	Element Parameters: eid=1343225905480
	Additional Element Parameters:
	Element Logic:
	`<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>`
STEP 12	Create a Site Entry named "feature-ga" by selecting the SEO CSElement created above. Ensure that Cache Rule is "Uncached" is selected.

STEP 13	Open Sites Content Explorer

STEP 14	Login using the following details.	
	Host name or address:	localhost
	Username:	fwadmin
	Password:	xceladmin
	Protocol:	HTTP
	Port:	8080
	Application Server URL Path:	CS Servlet (/cs/CatalogManager)

STEP 15	Navigate and open "ElementCatalog\OpenMarket\Gator\FlexibleAssets\FlexAssets\ContentDetails.xml" (For AVIArticle Assets) and "ElementCatalog\OpenMarket\Xcelerate\AssetType\Page\ContentDetails.xml" (For Page Assets).
STEP 16	Add the following code to add a new Analytics tab. Ensure that there are 3 ending </div> tags after the addition as shown in below.

```
<!-- Google Analytics CHANGES STARTS -->
<div dojoType="dijit.layout.ContentPane" title="Analytics" REPLACEALL="Variables.tabVanityUrl">
    <CALLELEMENT NAME="CustomElements/features/GA">
```

```
            <ARGUMENT NAME="assetId" VALUE="Variables.id"/>
            <ARGUMENT NAME="assetType" VALUE="Variables.AssetType"/>
            <ARGUMENT NAME="embedInIframe" VALUE="true"/>
        </CALLELEMENT>
    </div>
    <!-- Google Analytics CHANGES ENDS -->
    ...


    </div>
    </div>
    </div>
</FTCS>
```
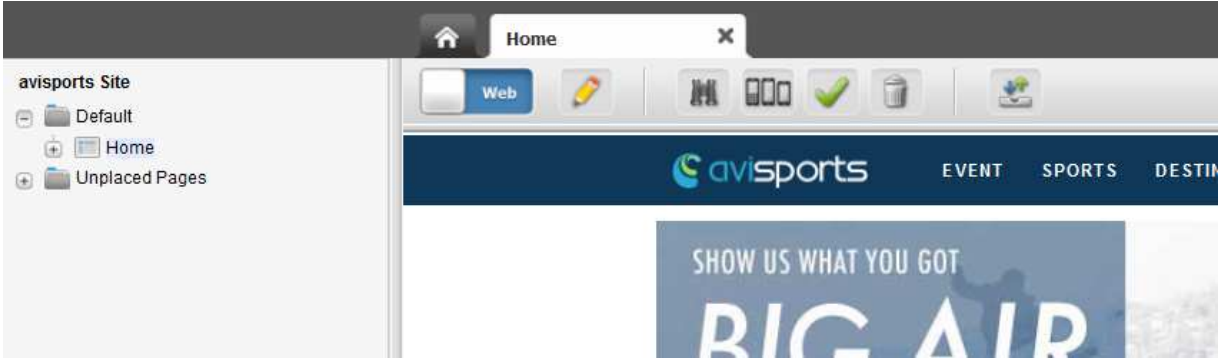
STEP 17	Add the Google Analytics scriplet to the header section of Page and Article Assets. The location of the header section depends on the Template being.

For Page assets, open the CS Element "ElementCatalog\Page\Head ",
For AVIArticle assets, open the CS Element "ElementCatalog\AVIArticle\Head ",
and add the following code

```
<script>
  (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
  })(window,document,'script','//www.google-analytics.com/analytics.js','ga');


  ga('create', 'UA-50963281-1', 'oracle.com');
  ga('send', 'pageview');


</script>
  . . .
```
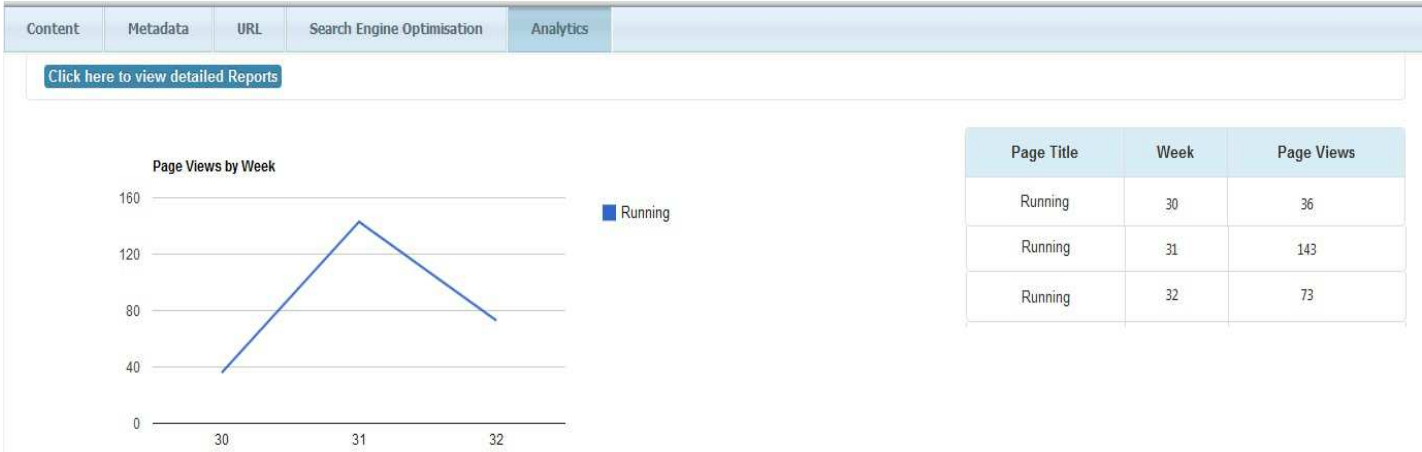
	</cs:ftcs>
STEP 18	Open WebCenter Site "avisports" in Contributor mode. Open Home Page from the left hand panel. Analytics tab looks like:

Debugging Asset Event Listener

STEP 1   Shutdown the application server (Tomcat / WebLogic).

STEP 2   Modify the log4j.properties file under the path "**$WCS_CS_DIR/WEB-INF/classes**" and the following line(s):

```
# PP : to debug Features
log4j.logger.oracle.webcenter.sites.framework.analytics=DEBUG
log4j.logger.oracle.webcenter.sites.framework.seo=DEBUG
```

STEP 3   Restart the application server

STEP 4   The log file (sites.log) contains log.