Oracle WebCenter Sites
Audit Configuration

Technical Notes

Pazhanikanthan Periasamy

paz.periasamy@oracle.com

ORACLE WebCenter Presales Team

August 2015

ORACLE®

TABLE OF CONTENTS

REFERENCE DOCUMENT

Name	Description	Date
Doc ID 1911760.1	Sample Code for Asset Listeners (Including Asset Approvals) and Publishing Event Listeners	May 08 2015

Overview

This document will describe how to register, implement and build audit reports based on WebCenter Sites Asset Listeners including Asset Approvals and Publishing Event Listeners.

Applies to

Oracle WebCenter Sites - Version 11.1.1.8.0 and later. Information in this document applies to any platform.
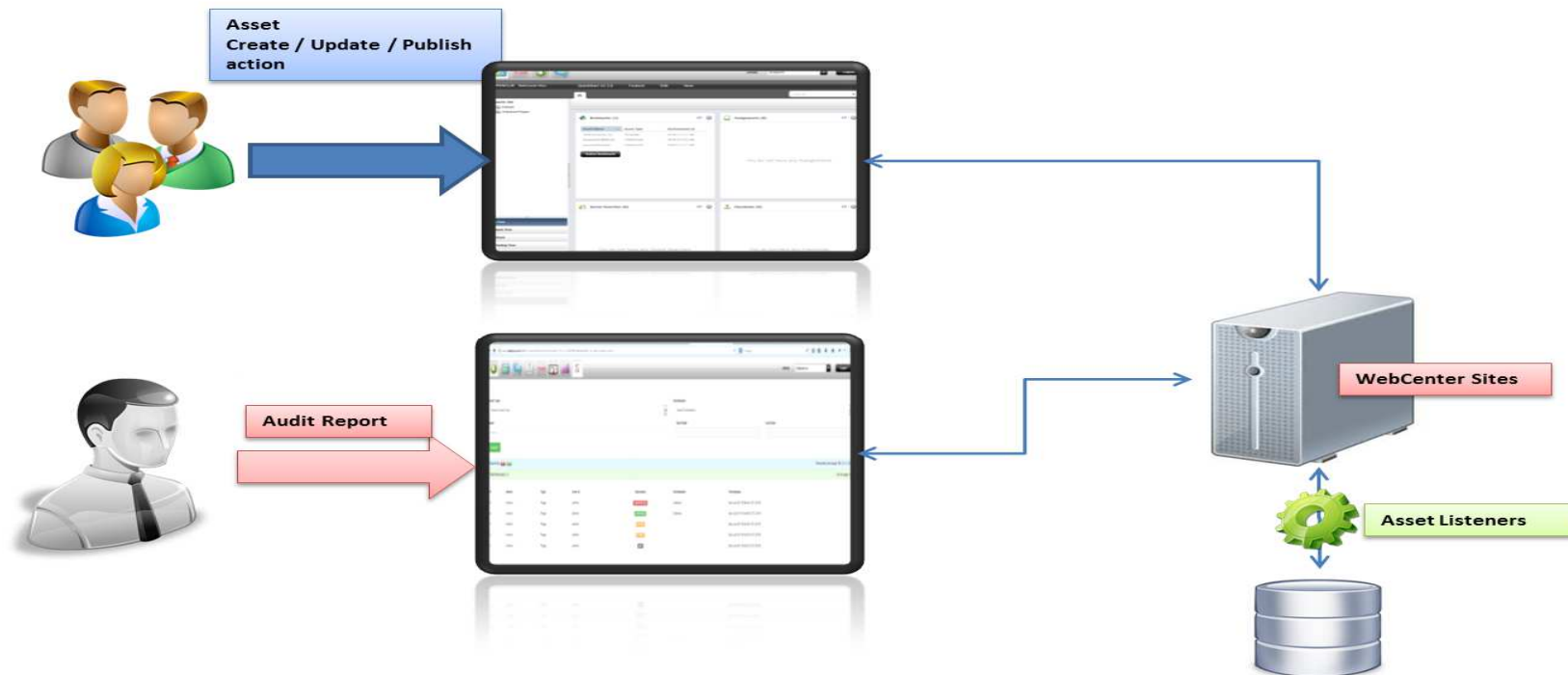
Technical Overview

Figure 1 Technical Overview

- Content Contributors perform **create**, **update** or **publish** operations on any assets (Page, Articles, etc) within WebCenter Sites contributor application as a part of their business as usual activities.
- A custom **Audit listener** listens to these events and then writes the information into an audit schema within the Sites application database.
- Content Contributors can view **asset level** and **consolidated report** with search and download (as PDF/XLS) capabilities.
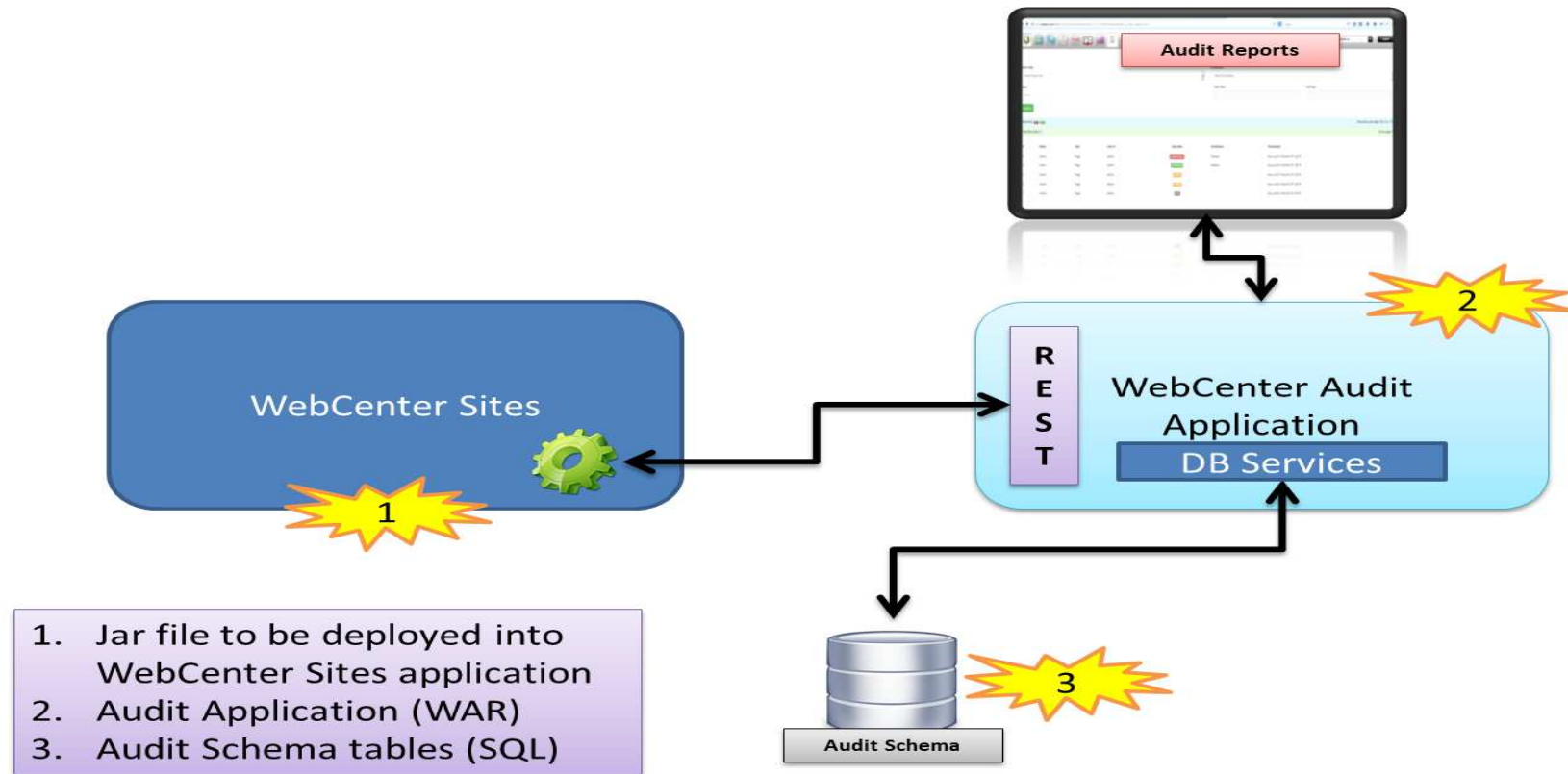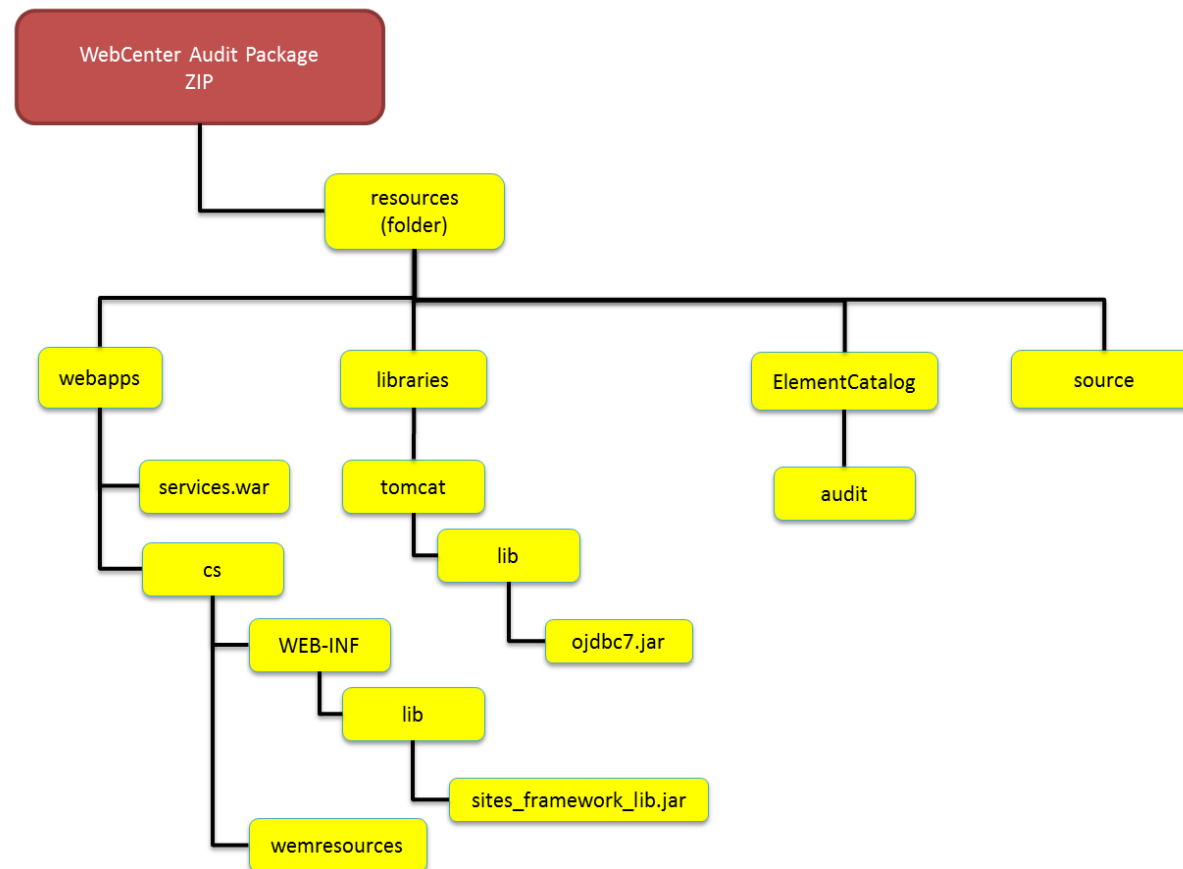
Installation Instructions

Figure 2 Installation Overview

Installation requires a set of steps to be followed in a sequential order. Each step may include a validation step to confirm that the installation has worked.

WebCenter Sites Audit installation is packaged as a ZIP (WCS_Audit_14082015.zip). The structure of the ZIP file is as follows.

```
WebCenter Audit Package ZIP
└── resources (folder)
    ├── webapps
    │   ├── services.war
    │   └── cs
    │       └── WEB-INF
    │           └── lib
    │               └── sites_framework_lib.jar
    │       └── wemresources
    ├── libraries
    │   └── tomcat
    │       └── lib
    │           └── ojdbc7.jar
    ├── ElementCatalog
    │   └── audit
    └── source
```

Schema setup

This installation supports:

- Oracle Database

Oracle Schema setup

Run the following SQL DDL commands within the Sites schema.

```
DROP TABLE audit_record;
CREATE TABLE audit_record (
    audit_id            NUMBER PRIMARY KEY NOT NULL,
    asset_id            NUMBER NOT NULL,
    asset_type          VARCHAR2 (50) NOT NULL,
    asset_name          VARCHAR2 (200) NOT NULL,
    target_asset_id     NUMBER NULL,
    target_asset_type   VARCHAR2 (50) NULL,
    target_asset_name   VARCHAR2 (200) NULL,
    user_id             VARCHAR2 (50) NOT NULL,
    operation           VARCHAR2 (20) NOT NULL,
    audit_timestamp     TIMESTAMP NOT NULL
);


DROP SEQUENCE audit_seq;
CREATE SEQUENCE audit_seq START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;
```

Note: If using JSK/QSK, create a new Oracle schema and run the above DDL scripts.

Application setup

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

STEP 2   (**Tomcat only**) Copy the file "resources/libraries/tomcat/lib/ojdbc7.jar" to tomcat server's lib folder.

STEP 3   Copy the file "resources/webapps/cs/WEB-INF/lib/sites_framework_lib.jar" to sites application WEB-INF/lib folder.

STEP 4   Copy the icons under "resources/webapps/cs/wemresources/images/icons/apps" to sites application
"/cs/wemresources/images/icons/apps" folder.

STEP 5   Configure a datasource for data access within the application:

o   (**Tomcat only**) Open the server configuration file under "tomcat/conf" folder and add the snippet with <Host> entry.

```
<!-- Audit Datasource start -->
<Context path="/services" docBase="services" debug="5" reloadable="true" crossContext="true">
        <Resource name="csDataSource" auth="Container" type="javax.sql.DataSource"
                driverClassName="oracle.jdbc.driver.OracleDriver"
url="jdbc:oracle:thin:@<host>:<port>:<service-name>" username="<<sites application schema user>>"
password="<<sites application schema password>>"
                maxActive="120" maxIdle="10" maxWait="-1"/>
</Context>
<!-- Audit Datasource stop -->
```

The file would look like:

```
<Host name="localhost"  appBase="webapps"
      unpackWARs="true" autoDeploy="true">

  <!-- SingleSignOn valve, share authentication between web applications
       Documentation at: /docs/config/valve.html -->
  <!--
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
  -->

  <!-- Access log processes all example.
       Documentation at: /docs/config/valve.html
       Note: The pattern used is equivalent to using pattern="common" -->
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
         prefix="localhost_access_log." suffix=".txt"
         pattern="%h %l %u %t &quot;%r&quot; %s %b" />

  <!-- Jairam -->
  <Context path="/cs" docBase="cs" debug="5" reloadable="true" crossContext="true">
      <Resource name="csDataSource" auth="Container" type="javax.sql.DataSource"
          driverClassName="org.hsqldb.jdbcDriver" url="jdbc:hsqldb:D:/APPS/QuickStart_2.3.6/database/csdb" username="sa" password=""
          maxActive="120" maxIdle="10" maxWait="-1"/>
  </Context>
  <!--
  <Context path="/csagentservices" docBase="csagentservices" debug="5" reloadable="true" crossContext="true">
  <Resource name="csDataSource" auth="Container" type="javax.sql.DataSource"
  driverClassName="org.hsqldb.jdbcDriver" url="jdbc:hsqldb:D:/APPS/QuickStart_2.3.6/database/csdb" username="sa" password=""
  maxActive="120" maxIdle="10" maxWait="-1"/>
  </Context>
  -->
  <!-- Jairam -->

  <!-- Audit Datasource start -->
  <Context path="/services" docBase="services" debug="5" reloadable="true" crossContext="true">
      <Resource name="csDataSource" auth="Container" type="javax.sql.DataSource"
          driverClassName="oracle.jdbc.driver.OracleDriver" url="jdbc:oracle:thin:@▮▮▮▮▮▮▮:▮▮▮▮:▮▮" username="▮▮▮▮▮▮" password="▮▮▮▮▮▮"
          maxActive="120" maxIdle="10" maxWait="-1"/>
  </Context>
  <!-- Audit Datasource stop -->

      </Host>
    </Engine>
  </Service>
</Server>
```

- o (WebLogic only) The datasource should already exist. If the schema is different from sites application schema, then a new datasource need to be created and targeted to the sites managed server.
- STEP 6   Copy services.war to
  - o (Tomcat) Tomcat's webapp folder.
  - o (WebLogic only) user_projects's application folder.
- STEP 7   Restart the application server.
- STEP 8   (WebLogic only) Deploy the services.war from user_projects's application folder and target to the sites managed server.

Application configuration
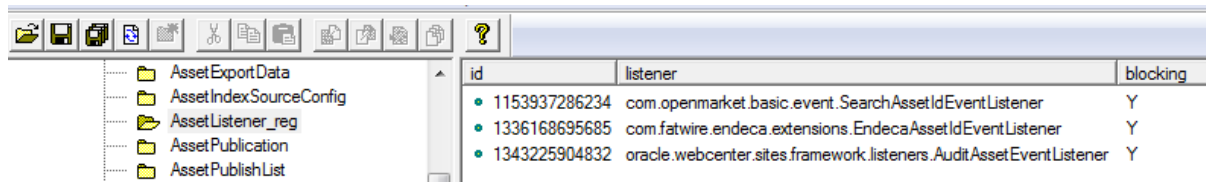
STEP 1   Open Sites Content Explorer. Login using the administrative details.

Host name or address:	Server Host
Username:	Administrator User Name
Password:	Administrator Password
Protocol:	http / https
Port:	Server Port
Application Server URL Path:	CS Servlet (/cs/CatalogManager)

STEP 2   To register the Audit Asset Listener, a row must be added into the 'AssetListener_reg' table.

Listener Class: oracle.webcenter.sites.framework.listeners.AuditAssetEventListener

STEP 3   Create a CSElement named "asset_report" (of jsp) under the path "CustomElements/features ". Copy and paste the code from the attached file "resources/ CustomElements/features/asset_report.jsp".

STEP 4   Create a CSElement named "get_attributes" (of jsp) under the path " CustomElements/features". Copy and paste the code from the attached file "resources/ CustomElements/features/get_attributes.jsp".

STEP 5   Create a Site Entry named "feature-audit-get-attributes " by selecting the "CustomElements/features/get_attribute" CSElement created above. Ensure that Cache Rule is "Uncached" is selected.

STEP 6   Restart the application server.

Asset level Audit configuration

STEP 1   Navigate and open "ElementCatalog\OpenMarket\Gator\FlexibleAssets\FlexAssets\ContentDetails.xml" (For Article Assets) and "ElementCatalog\OpenMarket\Xcelerate\AssetType\Page\ContentDetails.xml" (For Page Assets). Add the following code to add a new tab

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


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

STEP 2   When an asset is viewed in form view, Audit tab looks like:

STEP 3   On clicking the tab, the audit report specific to the current asset is displayed.

Consolidated Audit Application configuration
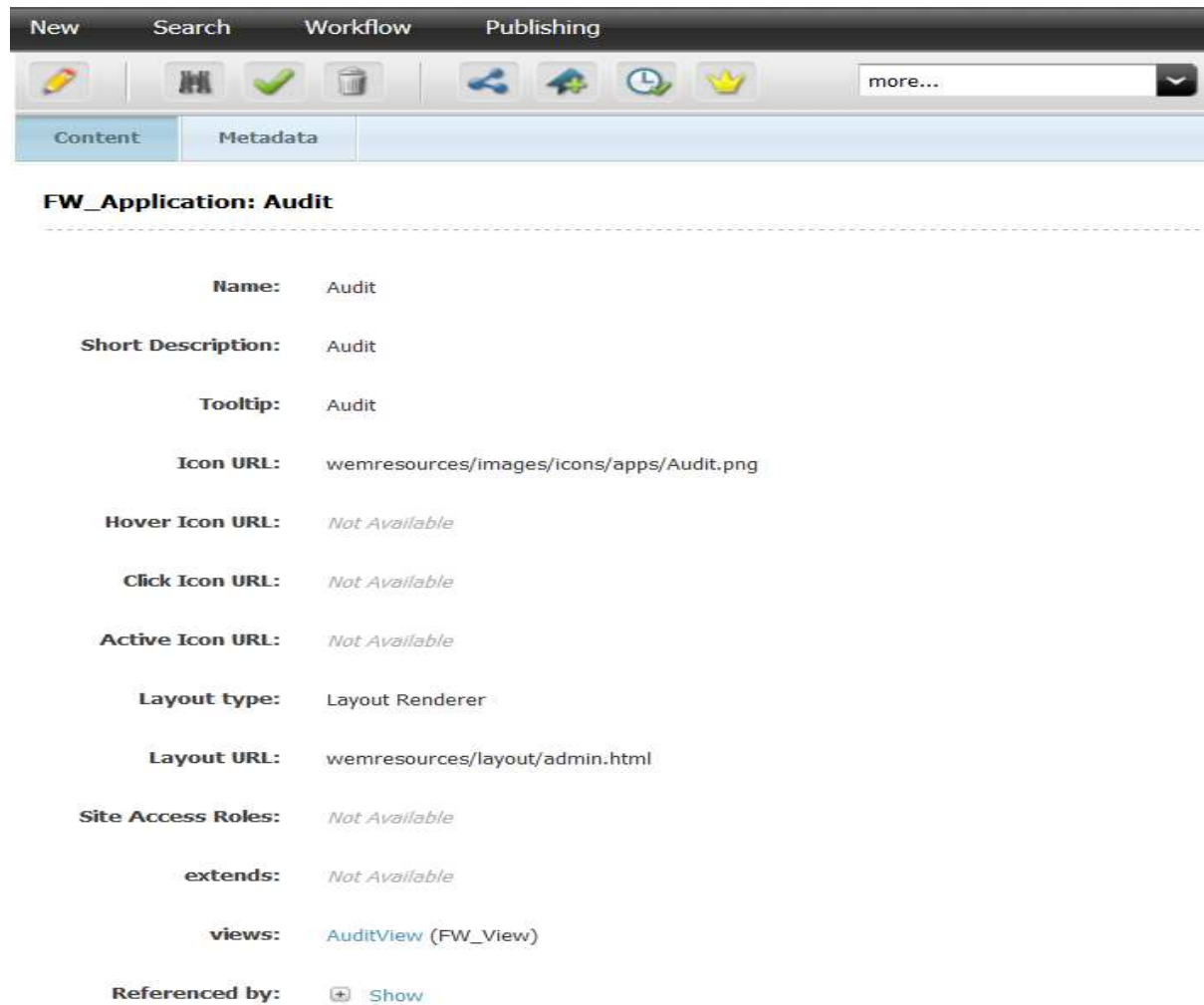
STEP 1   To create a consolidated audit application within the WEM bar, you would need to create a "FW_View" and "FW_Application".

STEP 2   Login into WebCenter Sites as an Administrator and switch to Admin Site and view the Admin view.

STEP 3   Create new FW_View called "AuditView" with the Source URL as: "http://<host>:<port>/services/audit"

FW_View: AuditView

Name:	AuditView
Parent Node:	frame1
View Type:	Iframe
Source Url:	http://localhost:8001/services/audit/
JavaScript:	*Not Available*
Content:	*Not Available*
Related:	*No Related Assets*
Referenced by:	⊕ Show
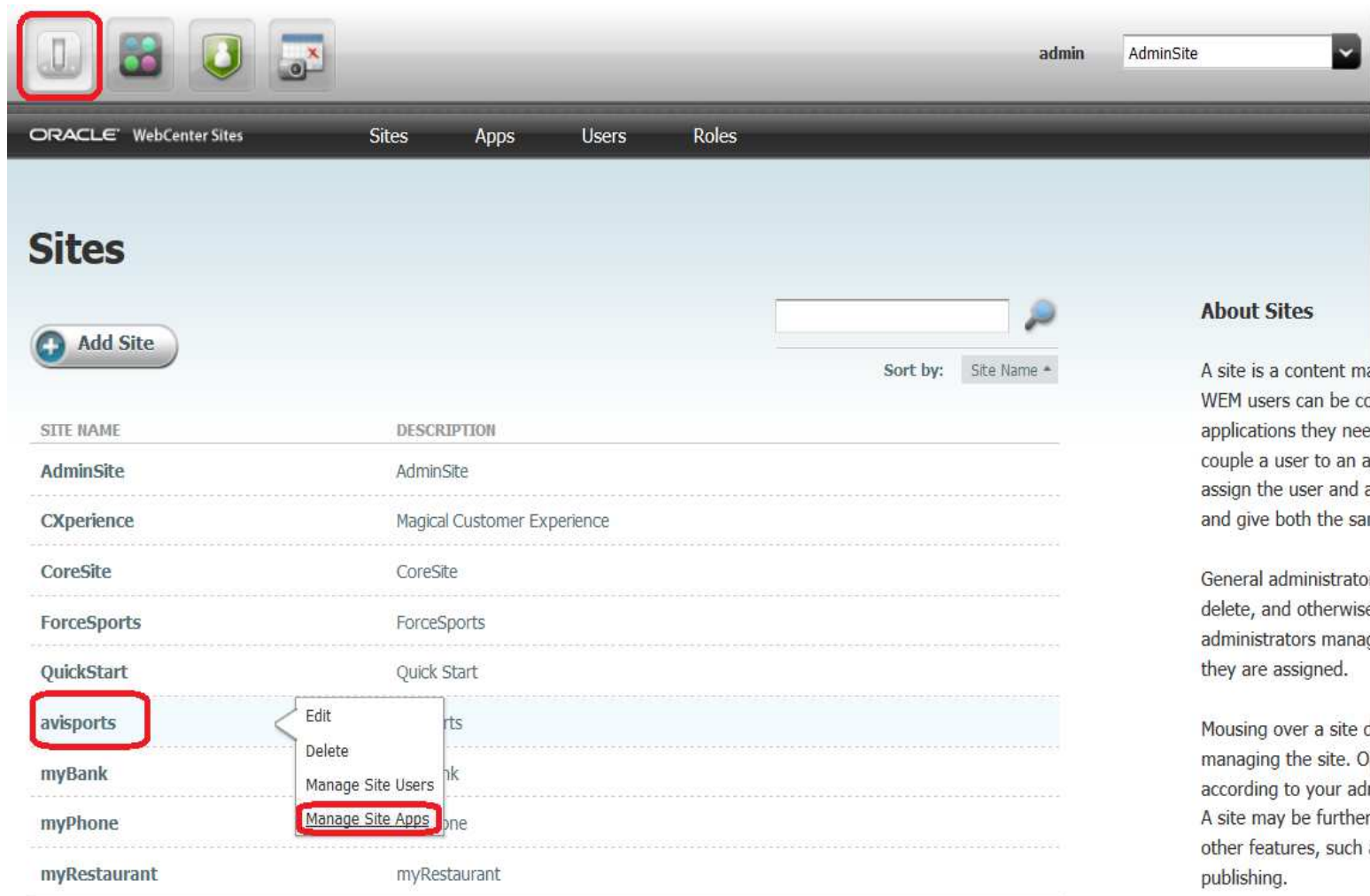
STEP 4   Create new FW_Application called "Audit" with the Icon URL as: "wemresources/images/icons/apps/Audit.png" and views as the "AuditView" created in the above step.

STEP 5   Switch to "WEM Admin" view and select the specific site where the Audit application needs to be added and select "Manage Site Apps".

STEP 6   Hit Assign Apps

STEP 7   Select the Audit application and hit continue.

STEP 8   Assign appropriate roles and hit Save and close.

STEP 9   Now switch to the site where Audit application was added. Refresh the browser window. A new WEM application should appear
on the WEM bar.

STEP 10   Click on the application to see a consolidated audit report.

STEP 11   Users can search based on specific "Asset Type", "Destination", "Name", "Start Date" and "End Date".

STEP 12   Users can also modify the number of results per page.

STEP 13   Users also have export functionality to export the report in excel or pdf format.

Debugging Asset Event Listener

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

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

```
# PP : to debug Asset Event Listener
log4j.logger.oracle.sites.framework.listeners=TRACE
```

STEP 3   Restart the application server

STEP 4   The log file (sites.log) contains logs files specific to the Audit Event listener.