|  |
| --- |
| Oracle Corporation |
| Accessibility |
| QA Test Plan |
| Version 12.1.4.0.0 |
|  |
| **/2013** |

Contents

[Functional Requirement of the Component: 3](#_Toc376541504)

[Testing Approach and Strategy 3](#_Toc376541505)

[Keyboard Testing 3](#_Toc376541506)

[Testing with Assistive Technology (AT) 4](#_Toc376541507)

[OGHAG Helper Toolbar 4](#_Toc376541508)

[Regression Testing 4](#_Toc376541509)

[Browser Compatibility Testing 4](#_Toc376541510)

[Automation Testing 4](#_Toc376541511)

[Functional Areas of Test 5](#_Toc376541512)

[1. Installation and Testing 5](#_Toc376541513)

[2. Keyboard Testing 5](#_Toc376541514)

[3. Dojo Upgrade Testing 8](#_Toc376541515)

[4. Shortcut Key Framework & Drag and Drop 10](#_Toc376541516)

[5. Upgrade 11](#_Toc376541517)

[6. Performance 11](#_Toc376541518)

[Integration 11](#_Toc376541519)

[Security 11](#_Toc376541520)

[Testing Environments 12](#_Toc376541521)

[Questions 12](#_Toc376541522)

# Functional Requirement of the Component:

An accessible website is one that is sufficiently flexible to be used by all people including people with Disabilities (PWDs). Although accessibility is a vital quality attribute for PWDs. Disabled people use assistive technologies, a term used to refer to hardware and software designed to facilitate the use of computers by people with disabilities. These technologies work satisfactorily as long as the page is designed well.

Meet the Oracle accessibility guidelines for Contributor UI. This should include the Contributor framework, form mode, web mode and search screens. Scope in the current release 12.1.4.0.0 is to make the Contributor UI accessible.

The scope of the Accessibility Sites application to meet all the OGHAG rules and WCAG 2.0 Standards by following the Oracle Accessibility Guidelines.

* OGHAG rules

<http://www-apps.us.oracle.com/atg/accessibility/html/OGHAGv13.html>

* WCAG 2.0 rules:

<http://www.w3.org/TR/WAI-WEBCONTENT/wai-pageauth.html#toc>

# Testing Approach and Strategy

The goal of testing for accessibility is to determine the extent to which the product has been properly coded to meet the [OGHAG](http://www-apps.us.oracle.com:1100/atg/accessibility/html/Generic%20HTML%20Checklist.htm) standards, so that your product can produce a VPAT (the conformance statement) that is accurate and does not create a liability for the company.

Approach

A description of how and where testing will be performed and explain any issues that have a major impact on the success of testing and ultimately on the project*.*

### Keyboard Testing

Keyboard testing following all page elements (folder tabs, links, buttons, radio buttons, checkboxes, popup close window buttons and links, expand and collapse trees, selectable lists, form entry fields, frames, panels) to ensure that they are accessible, show visual and physical focus, and can be activated without the use of the mouse.

Link: <http://globaldc.oracle.com/perl/twiki/view/AccessibilityProgram/KeyboardOnlyTesting>

### Testing with Assistive Technology (AT)

Testing with AT is the 'ultimate' test; it's the best way to ensure that product actually works for a disabled user, and it should be performed after all other styles of testing have been done. And in this respect, testing with AT tests all standards. Here in testing we are using JAWS [Job Access with Speech] Screen reader for testing the application.

Link: <http://globaldc.oracle.com/perl/twiki/view/AccessibilityProgram/AboutJaws>

### OGHAG Helper Toolbar

The Toolbar is a Firefox add-in developed by the Accessibility Program Office, to help you perform some testing of your HTML-based products. We test below areas in our application using the OGHAG Helper Toolbar.

* Table Inspector
* Form Helper
* Image Helper
* Structure Helper
* Link Helper
* Color Contrast Analyzer
* Linearise Page

Link: <http://globaldc.oracle.com/perl/twiki/view/AccessibilityProgram/OGHAGHelperToolbar>

### Regression Testing

For every merge of the feature branch build into the Trunk the regression testing will be run. We have identified the related Sanity Suite which needs to be run as a part of regression testing – this is the sanity cycle of Sites plus high priority test cases identified in Accessibility testing.

Every time there is merge from Accessibility FB into trunk – we will run the complete related Sanity Suite and also run the high priority stories completed accessibility

### Browser Compatibility Testing

We support browser FireFox [latest ESR] only for testing the Contributor UI accessibility. In remaining browsers [IE9, IE10, Chrome, Safari7] we see that no functionality is breaking.

Even though there are accessibility issues in other browsers, they are of low priority and will not be fixed in this release 12.1.4.0.0.

### Automation Testing

QA has evaluated the automation feasibility of the accessibility test cases developed. We identified that keyboard driven accessibility test cases can be automated using Selenium tool. Below is our strategy for automation of the accessibility test cases.

1. Phase I: Automation of Contributor UI test cases from the BAT suite using keyboard navigation
2. QA will identify the Sanity suite for accessibility and automate the same using selenium. This suite will be used as a high level regression of the accessibility with respect to keyboard navigation.

Note: Scope of automation of accessibility using keyboard navigation is limited to contributor UI pages.

All areas of test will be evaluated for automation – the process for determining and automating is the following:

1. Tests for each area of the product under test are identified
2. For each manually testable area, the test cases are created
3. When test case is created, it is decided if that test case can be automated
4. If the test case is a candidate for automation, then the keyword AUTO is added to the test case
5. After tests are executed manually and all steps and verification points are added, the test cases are then added to the automation suite for the component (this includes engineering automation as well as QA automation as not all tests are candidates for QA automation)
6. For components of this product that are not possible to be tested manually, code coverage and functional coverage of the API will be substituted the overall summary of testing will be discussed between QA and Engineering so that we all understand what is covered in the scope of product testing.
7. The progress of this should be reported cumulatively in each sprint review in the format of:
   1. Number of test cases for component (NOTE this might include tests from earlier versions that need to be moved forward – in that case the number of new tests should be reported as well)
   2. Percentage of test cases targeted for automation
   3. Percentage of test cases automated

Please refer to the TestLink for test execution results and agilefant for project updates and BugDB for the bugs

TestLink: <https://testlink.us.oracle.com/testlink/login.php>

Agilefant: <http://fwsevm100.us.oracle.com/agilefant/editProject.action?projectId=44>

## Functional Areas of Test

### Installation and Testing

Testing of the installation will ensure that all components are installed properly and verification should include proper jar files are deployed, default properties are configured properly post install, all out of the box assettypes are properly installed and enabled for all sites where applicable. Logging is properly enabled will also be included in install testing, the individual logs will be covered during functional testing to ensure that they are recording and producing relevant information and not displaying any security related vulnerabilities.

If there are any jars specific and files related to accessibility then upgrade team has to make sure that everything is working fine after upgrade.

### Keyboard Testing

Focus of testing is on the Keyboard Navigation between the menus tree tabs, asset form tabs and Segment asset operations, Basic asset, Flex asset.

* Tree navigation and operations(left click and right click)
* Menu bar (Content, Edit and Vew) , Menus.
* UI Tabs navigation and asset from Tabs
* Segment Asset Operations: Copy/Inspect/View/Edit/New/Approve/Delete/Status/Revision Tracking/Access Permission
* Basic assets operations[New,Edit,Inspect,Copy,Status,Permission,Versioning,Approve]
* All the Attribute Editors will be covered as part of testing.
* Flex asset operations [New,Edit,Inspect,Copy,Status,Permission,Versioning,Approve]
* WEM Bar: App Selection,Site Selection,Logout,Profile,WEM bar close and open

Tree

* Tree navigation and operations (left click and right click)
* Tree tabs/panel
* White space right click

Menu

* Menu bar (Content, Edit and View)
* Menus

UI Tabs

* Main tabs (Dashboard, Search, etc.)
* Form view tabs (Content, Metadata, URL, etc.)
* Main tabs drop down to select hidden tabs

Toolbar buttons

* Webmode/Form mode – edit/save, checkin/out, revision tracking
* Preview mode – date selector, segment selector
* Refresh button and Device panel button
* Message in toolbar

WEM Bar

* App selection
* Site selection
* Logout
* Profile
* WEM bar close and open button

Search and Browse Results

* List view
* Thumbnail view
* Dock view
* Search toolbar

Dashboard

* Refresh
* Widget refresh and help
* Navigate between widgets

Device picker

Flexible Menu

Multi column menu e.g. search arrow drop down and the one that opens when you click on Content > New

Pages

* Advanced search
* Approval
* Delete
* Versioning
* Mobile preview
* Multi preview
* Edit profile page
* Login Page
* Site Selection screen

Insite Editing (Web Mode)

* Web Page navigation between various slots and inline edit boxes on the page
* Single valued Slot
* Multi valued slot
* Inline edit box
* Date
* Tooltip bar

CKEditor

* Form mode
* Webmode
* Drag & Drop
* Search to slot and form
* Tree to slot, form and within tree
* Slot
* Dashboard widgets

Forms

New content form

1. Focus on field

2. Single and multi valued attribute editors

* Text box
* Select box
* Type ahead
* File uploader
* Transfer box
* Date text box

Other Tabs in Forms

* URL
* Metadata
* Marketing

Status page and the Workflow actions you can take (Add to workflow, Approve Reject etc)

Segment

Recommendation

### Dojo Upgrade Testing

As a part of the dojo upgrade tested the basic sanity of the below specified components

Admin UI

• Verified the Creation of a site and assign a user to that site

• Verified adding new ACL, Role to the user

• Verified the creation with desired roles, assigning the user to the created site • Assigning user to a site with different roles

* Basic Assets: Verify CRUD operations / Access Permissions/ Searching on Basic assets/ multilingual asset/UTF-8 characters/Associations/Subtypes
* Flex Assets: CRUD operations/ Attribute Editors / Access Permissions/Searching on Flex assets/Creation of UTF-8 characters/multilingual asset

Search

• Creating and sharing the smartlist from undocked search results (Thumbnail/List View)

* Bulk operations delete/approve/bookmark from the undocked search results(Thumbnail/List View)

• Starting and Stopping of Lucene index for global and asset type index

• Advanced Search function based on keywords, asset, subtype and attributes

• Search the basic and flex asset providing the search criteria

Revision Tracking

* RT Operations Checkin/Checkout/UndoCheckout/Show Versions in Form mode and Web mode/Associations/untrack

Workflow

• Workflow states and Processes/assets to place in workflow/Workflow operations[View Status, Finish My Assignment,Abstain from Voting,Delegate Assignment,Show Participants, Remove from Workflow, Set Assignment Deadline, Set Process Deadline]

• Workflow reports

• Placing the multiple asset in workflow Groups

* Approval & Publishing : Initialization of destination/Mirrorsite configuration/Scheduled publishing/Exist dependency of Flex Assets/Approve with Dependencies/Publishing page to delivery CS
* ER: functionality of System Info/Log Viewer/File System Test/Configure log4j
* Mobility : Creation of new site and assign a user to that site/Creation of Device Group, Device Image,SitePlan,Pages,templates,Article assets, Copy Site Navigation -Paste Site Navigation, Change template Layout,Approval and Publish of Site Plan
* Templates: Creation of Typed/Typeless/site entry & CSElement
* Engage Assets: Creation of Promotion,Segment,Recommendation / Visitor attributes/History attributes/Recommendations using Multiple Segments
* VanityURL : Creation of Patterns/Pretty URL's accessed with cached in URL cache tool/Creation of url(absolute URL)after creating the webroot/Pretty URL created during the time of installation/auto generated blob url's
* CKEditor: Creation of Add asset link,Include asset for flex asset in form mode and Webmode / Previewing and Editing in the insite of Asset
* Site Launcher: Copy/Replication of site & Mirroring of site using site launcher by selecting the destination
* Webmode Testing [Insite Editing] is tested here.

### Shortcut Key Framework & Drag and Drop

Keyboard Shortcut FrameWork is implemented in the below components

* + WEM Bar
  + Menu bar
  + Search Box
  + Drag and Drop
  + Deselect the Drag and Drop
  + Open Active Tree
  + Refresh button
  + Focus on the 1st asset in Search result page
  + Search results Refresh Button
  + Active Tab Window
  + Highlight Active tab
  + Focus on the Mode Button
  + Delete
  + Bookmark/unbookmark

Drag and Drop operations requires two actions i.e Select the asset either from search results or Left Navigation tree (Site tree, Content Tree, Marketing tree and My Work)and drop the selected asset into the Drop zone available in the Asset form

Form mode

Drag and drop assets from tree/search grid using keyboard/JAWS

* Drag image assets into image picker using keyboard/JAWS
* Drag assets into pickasset attribute editor using keyboard/JAWS
* Slot properties dialog using keyboard/JAWS
* Able to work with the multivalve attribute using dialog in the webmode with the help of keyboard/JAWS
* Able to save the search/share it using dialog box with the help of keyboard/JAWS
* Create assets/change layout (e.g. page asset) using dialog with the help of keyboard/JAWS
* Dialogs to test using keyboard and JAWS.
* Drag and Drop in webmode[Not yet implemented]

### Upgrade

Open Question – will this upgrade is applicable for accessibility or is that out of scope

If there are any jars specific and files related to accessibility then upgrade team has to make sure that everything is working fine after upgrade.

### Performance

As part of the sites performance testing accessibility performance testing will be a part of sites performance testing.

## Integration

Community as a part of integration will be tested and it should be accessible. The focus would be on Comments Widgets, Reviews Widgets, Ratings Widgets, and Polls.

For all the new implementations, it is the responsibility

## Security

Reference document - <https://stbeehive.oracle.com/content/dav/st/ASQA%20Security/Public%20Documents/Security%20Test%20Specification%202011%20External.pdf>

Functional testing of all the security features:

* 1. Authentication
  2. Authorization
  3. Password encryption
  4. Session Management
  5. Install
  6. No Passwords in plain text
  7. No Stack trace in browser
  8. Testing of SSL
* We need to ensure that all UI fields are tested for XSS and SQL injection
* AddAnalyticsImg also needs to be tested to prevent XSS and SQL injection from being injected as part of page template code.
* Injecting data via REST API should be tested for prevention of XSS and SQL Injections from being injected as part of test data injection

## Testing Environments

NOTE – this needs to be finalized once deployment diagram is provided to QA from Engineering, for now listing the typical install base and will add more specific details of which component within Analytics will be tested on which combinations of the below once this is received.

AppServer:

* Tomcat 7+
* WAS 8.5+
* WL 12c

\*\*final list to be determined once certification matrix is finalized.

OS:

* Linux
* Windows

DB:

* Oracle 12
* Oracle 11gR2
* DB2 9.x
* SQL 2012

Browser:

* IE9
* IE10
* FireFox (latest ESR)
* Chrome (latest)
* Safari 7

# Questions

* + 1. Will the person with a disability test this before releasing?
    2. Screen Readers (like JAWS) are not stable and periodic updates are happening for these screen readers. So, if some features are working in the latest and not working in updated versions so who is responsible for this?
    3. Motor impairments: Prevent the use of a mouse or a standard keyboard [How to test this? ]