Cornell Selenium Test Checklist 1.8 Release Candidate 4

Test Suites	Pass/Fail	Description	Special Conditions	Improvements	JIRA Issues
AdminSecurity	Pass	Tests logging into existing admin account with wrong password, changing password for existing account incorrectly then correctly, adding new admin account, logging into new admin account, checking account privileges of new admin account, and deleting new admin account.	**Must have runtime.properties email.smtpHost and email.replyTo not set.**		
InitialSetup	Pass	Tests changing site name, tagline, and copyright then resetting it to default.	**Must modify EditSiteInfo test case in three places to check for current release number and in four places to check for copyright year (if it has changed since last release).**		
CreateAccounts	Pass	Tests creating self-editor, curator, and editor users and then logging into each account and checking account privileges.	**Must have runtime.properties email.smtpHost and email.replyTo not set.**		
CreateClasses	Pass	Tests creating class group and two sub-classes, then adding values, viewing class hierarchy, and deleting everything created.			
CreateProperties	Pass	Tests creating property group, changing display rank, adding object property, editing object property, adding data property, and deleting everything created.		Need to investigate how to get Selenium to check the Property Group display rank--will need to update AddPropertyGroupPlace and ChangePlaceDisplayRank test cases.	
CreatePeople	Pass	Tests creating a faculty member and librarian, adding info then editing it, uploading images and then changing them, viewing public display, searching for what was created, associating librarian to faculty, entering a name with special characters, and deleting everything created. (Tests check for incorrect values (e.g. picture size too big/small) and canceling out.)		Need to investigate how to get Selenium to check the order of collaborators in ReorderPeople. **Need to figure out why upload of an oversized picture doesn't work on my new laptop but does on the old one.**	
SelfEditing	Not tested for RC4	Tests setting up faculty member as self-editor, making edits, restricting logins, and deleting everything created.	**Must have runtime.properties selfEditing.idMatchingProperty set to http://vivoweb.org/ontology/core#orcidId** This test suite will no longer run unassisted. Must run each test case one at a time because associating a profile with a user account has started flaking out for some reason.		
ProxyEditing	Not tested for RC4	Tests adding proxy editors, associating one with a profile, then two, editing profile, removing proxy permission, and deleting everything created.	**Must have runtime.properties selfEditing.idMatchingProperty set to http://vivoweb.org/ontology/core#orcidId** This test suite will no longer run unassisted. Must run each test case one at a time because associating a profile with a user account has started flaking out for some reason.		
AddNonPersonThings	Pass	Tests creating activity, event, concept, equipment, and location, then adding fields, checking public and index views, searching, and browsing, menu management, and deleting everything created. (Tests check for incorrect values and canceling out.)		Need to flesh out test case CheckPublicView plus a dd test cases for editing/adding to existing organizations, activities, course, events, topics, equipment, and locations.	
AddPublications	Pass	Tests creating book, article, chapter, and conference for faculty member, checking public and index views, searching, and deleting everything created. (Tests check for incorrect values and canceling out.)	This test suite will not run unassisted. Must run each test case one at a time because test case PublicationDisplay flakes out when pressing the "Manage Publications" link.	Need to investigate why PublicationDisplay flakes out when clicking on "Manage Publications." Need to double check tests to make sure all fields are tested plus add test cases for editing/adding to existing publications, reordering and removing authors, other publication types, and additional fields. **Need to change some of the page text due to a different way of saying "select an existing one or create a new one."**	
AddRoles	Pass	Tests adding various roles to faculty member and librarian, editing the roles, checking public and index views, searching, grant display, repair mode, and deleting everything created. (Tests check for incorrect values and canceling out.)	This test suite will not run unassisted. Must run each test case one at a time. * The Grant Display test case will often flake out when pressing the "Manage Grants & Projects" link. * The CheckRepairMode test case will often flake out on an assertConfirmation command.	Need to investigate why GrantDisplay and CheckRepairMode flake out .	
CheckVisualizations	Not tested for RC4	Tests adding variety of publications for a faculty member without co-authors, then with co-authors, checking publication and network display, adding variety of grants for a faculty member without co-investigators, then with co-investigators, checking co-investigator network display, viewing temporal graphs, and deleting everything created.	**Must have runtime.properties visualization.temporal enabled.**		

CheckMapOfScience	Not tested for RC4	Tests adding variety of publications for a faculty member and checking map of science.			
CustomEducationForm	Pass	Tests adding multiple education backgrounds for a faculty member, editing them, checking public and index views, searching, and deleting everything created.			
CustomPositionsForm	Pass	Tests adding multiple positions for a faculty member, editing them, checking public and index views, searching, and deleting everything created.			
CustomAwardsForm	Pass	Tests adding multiple awards for a faculty member, editing them, checking public and index views, searching, and deleting everything created.			
CustomAdvisesForm	Pass	Tests adding multiple advisees for faculty member, editing them, checking public and index views, searching, and deleting everything created.			
OntologyManagement	Pass	Tests adding and deleting an ontology.			
ProcessRDFData	Pass	Tests adding RDF from an internet file and from a local file, deleting both, and exporting RDF.	The test suite will stop when it gets to test case ExportRDFFile because it is expecting a human to select to save the file.	Need to investigate how ExportRDFFile test case can save a file without human intervention.	
SearchTextDiscovery	Not tested for RC4	New test as of release 1.8			
MultiLabels	Not tested for RC4	Tests adding multiple labels for a person and selecting one to use.	**This test is a combination of manual and automated steps-- the test suite contains the test file.** 1) Run test case RebuildSearchIndex (to rebuild the search index). 2) Run test case AddTestFaculty (to create Jane Faculty). 3) Get the n#### from the Resource URL on the profile page. 4) Edit TestMultipleLabels.txt to change n#### to the correct number. 5) Run test case AddRDFData (to load labels for Jane Faculty). 6) Run test case ManageLabels (to remove first 2 labels and use last one). 7) Run test case DeleteTestFaculty (to remove Jane Faculty). 8) Run test case RemoveRDFData (to remove labels).	Make changes from Jim to fully automate this test... 1) Skip the "AddTestFaculty" step. 2) Execute "AddRDFData" step, but the file to ingest is TestMultipleLabels.n3, not TestMultipleLabels.txt. 3) Execute "Manage Labels" step. No change. 4) Skip the "DeleteTestFaculty" step. 5) Execute "RemoveRDFData", but the file to remove is TestMultipleLabels.n3, not TestMultipleLabels.txt	
LanguageSupport	Not tested for RC4	Tests display of foreign characters (German) with Firefox set for English and Firefox set for German.	**This test is a combination of manual and automated steps-- the test suite contains the test file.** 1) Run test case RebuildSearchIndex (to rebuild the search index). 2) Run test case AddRDFData (to create Jane Faculty and her overview using test file *test-ut8*). 3) Run test case CheckEnglishOverview (to verify that the English overview is displaying). 4) Open up Firefox's *Options* and use the *Content* tab to set the language to *German (de).* 5) Run test case CheckGermanOverview (to verify that the German overview is displaying now). 6) Go back into Firefox to set the language back. 7) Run test case CheckEnglishOverview (to verify that the English overview is back). 8) Run test case RemoveRDFData (to remove Jane Faculty and her overview using test file *test-ut8*).	Need to investigate how this test suite could be further automated.	
LinkedOpenData	Not tested for RC4	Tests linked open data.	**Run "ant deployTestApp" at any point before this test.**		
SearchExclusion	Not tested for RC4	Tests excluding a class from the search.	**This test is a combination of manual and automated steps-- the test suite contains the test file.** 1) Run test case RebuildSearchIndex (to rebuild the search index). 2) Run test case CreateTestFacultyMember (to create test person "Faculty, Jane"). 3) Run test case SearchFound (to search for *faculty* , who will show up in search results). 4) Stop tomcat. 5) Copy TestSearchExclusion.n3 to [data]\rdf\display\everytime\ (data is where runtime.properties file is). 6) Start tomcat. 7) Run test case RebuildSearchIndex (to rebuild the search index). 8) Run test case SearchNotFound (to search for *faculty* , which will result in no hits). 9) Stop tomcat. 10) Remove [data]\rdf\display\everytime\T estSearchExclusion.n3. 11) Start tomcat. 12) Run test case RebuildSearchIndex (to rebuild the search index). 1 3) Run test case SearchFound (to search for *faculty* , who will show up in search results again). 14) Run test case DeleteTestFacultyMember (to remove the Jane Faculty test person).	Need to investigate how this test suite could be further automated.	

SearchBoost	Not tested for RC4	Tests individual boosting by creating 5 people and boosting one of the people in the middle of the search results to the top. Tests class boosting by having concept show up before person.	**Run the SearchBoost test suite to test individual boosting (the class boasting test is manual and is listed below).** This test suite will not run unassisted. Must run each test case one at a time. * After CreateTestPeople tester must update BoostIndividual.txt to add the n#### field for Echo Foxtrot. * When running test case TestIndividualBoost, must review search results manually (the test case has pauses built in). At first Echo Foxtrot is in the middle of the search results. After boosting him, he is at the top. Then he goes to the end (was middle prior to 1.8). **To test class boosting, must do the manual steps below.** 1) Rebuild the search index. 2) Create foaf:person "Jonson, Ben". 3) Create a skos:Concept named "Jonson, Ben". 4) Search for *jonson* . The person should be first result and the concept second. 5) Stop tomcat. 6) Copy BoostClass.n3 to [data]/rdf/tbox/filegraph/ (in folder where runtime.properties lives). 7) Start tomcat. 8) Rebuild the search index. 9) Search for *jonson* and now the concept should be the first result. 10) Stop tomcat. 11) Remove /rdf/tbox/filegraph/BoostClass.n3. 12) Start tomcat. 13) Rebuild the search index. 14) Search for *jonson* . The person should be first result again. 15) Remove the test person and concept.		
ShortViews	Not tested for RC4	Tests the implementation and use of short views.	**This test is a combination of manual and automated steps-- the test suite contains the test files.** 1) Copy view-browse-faculty.ftl, view-index-department.ftl, view-index-faculty.ftl, and view-search-faculty.ftl to [vivo]/themes/wilma/templates/. 2) Copy shortview_config.n3 to [vivo]/productMods/WEB-INF/resources/. 3) Deploy VIVO ("ant all"). 4) Start tomcat. 5) Run the test suite (which includes a rebuild of the search index). 6) Set everything back and redeploy.	Need to investigate why CheckResults flakes out on the menu testing and how this test suite could be further automated.	
OpenSocialGadgets	Not tested for RC4	Tests the implementation and use of open social gadgets.	**This test is a combination of manual and automated steps-- the test suite contains the test files if no other location is given.** 1) Open phpMyAdmin and select the vivo database. 2) Select the Import option. 3) Browse to find [vivo]/vitro-core/opensocial/shindig_orng_tables.sq l and load it. 4) Browse to find [vivo]/vitro-core/opensocial/shindig_example_gadgets.sq l and load it. 5) Within the folder where runtime.properites lives, create a shindig folder. 6) Within the shindig folder, create two sub-folders: conf and openssl. 7) Place the shindig encryption key (file securitytokenkey.txt) in the openssl folder. 8) Copy the setenv.bat file into the [tomcat]/bin folder (edit it as needed). 9) Update the build.properties file to uncomment the open social settings and set them appropriately: OpenSocial.shindigURL = http://localhost:8080/shindigorng OpenSocial.tokenService = lib-hlm7.library.cornell.edu:8777 OpenSocial.tokenKeyFile = c:/VIVO/data/shindig/openssl/securitytokenkey.txt 10) Deploy VIVO ("ant all"). 11) Run "ant orng". 12) Start tomcat. 13) Run the test suite (which includes a rebuild of the search index). 14) Set everything back (at a minimum comment the stuff in build.properties), delete and recreate the database, and redeploy. *(Note: See [vivo]/doc/setting_up_orng.html for more information.)*	Need to investigate how this test suite could be further automated.	

SparqlUpdateApi	Not tested for RC4	Tests sparql queries.	**This test is a combination of manual and automated steps-- the test suite contains the test files.** 1) Copy TestSparqlApi.html to [VIVO]/productMods/. 2) Deploy VIVO ("ant all"). 3) Restart tomcat. 4) Run test case TestApiDisabled. 5) Copy EnableSparqlUpdateApi.n3 to [Vitro]/webapp/rdf/auth/everytime/. 6) Deploy VIVO ("ant all"). 7) Restart tomcat. 8) Run test case TestApiWorks. 9) Run test case TestApiFailure. 10) Remove [VIVO]/productMods/TestSparqlApi.html. 11) Remove [Vitro]/webapp/rdf/auth/everytime/EnableSparqlUpdateApi.n3. 12) Deploy VIVO ("ant all"). 13) Restart tomcat.	Need to investigate how this test suite could be further automated.	