Solr Plugin

Table of Contents

The solr plugin is one of the OFBiz plugins.

1. Solr Apache OFBiz® plugin

The solr plugin is one of the OFBiz plugins.

This document describes the Ofbiz solr plugin, an Ofbiz (http://ofbiz.apache.org/) implementation of the Apache Solr search platform (http://lucene.apache.org/solr/). The solr plugin includes an Ofbiz service-based wrapper layer to the Apache Solr webapp queries as well as the native Apache Solr web interface itself. Currently, the solr plugin focuses on Product data. Note: This document is a work in progress; information is subject to change.

1.1. Installation

To install solr in an Ofbiz setting, simply extract the solr directory and and sub-folders to the hot-deploy folder. Afterward, the solr home system property (solr.solr.home) must be set to the value hot-deploy/solr manually using one of the following methods: solr.solr.home in batch/script file: Add the parameter "-Dsolr.solr.home=hot-deploy/solr" to the Java command invocation for ofbiz.jar. e.g.: "%JAVA_HOME%\bin\java" -Xms128M -Xmx512M -XX:MaxPermSize=512m -Dsolr.solr.home=hot-deploy/solr -jar ofbiz.jar solr.solr.home in Ant build configuration: In your root Ofbiz build.xml file, add the element "<jvmarg value="-Dsolr.solr.home=hot-deploy/solr"/>" to the "<java jar="ofbiz.jar"...>" invocation of the appropriate Ant target(s) (run, start, run-install, etc.). e.g.: <target name="start" description="Start OFBiz"> <java jar="ofbiz.jar" fork="true"> <jvmarg value="${memory.initial.param}"/> <jvmarg value="${memory.max.param}"/> <jvmarg value="${memory.maxpermsize.param}"/> <jvmarg value="-Dsolr.solr.home=hot-deploy/solr"/> </java> </target> * It may be possible to specify solr home using other methods (JNDI, web.xml), but at the time of this writing, this was the most reliable method known.

1.2. Configuration

The solr plugin can run out-of-the-box without configuration, but many files, settings and interfaces allow custom settings. Some of these include: Ofbiz configurations: System properties: ofbiz.solr.eca.enabled - Global solr ECA toggling boolean (true/false, see Data Indexing) Config files: solr/config/solrconfig.properties - Ofbiz solr service behavior control ofbiz-component.xml - Standard Ofbiz component config Apache Solr configurations: System properties: solr.solr.home - Solr home (see Installation) Config files: solr.xml - Base solr config conf/schema.xml - Solr index schema webapp/WEB-INF/web.xml - Dual Ofbiz/Solr webapp config Interfaces: /solr/admin/ - Webapp admin interface (see below) * It is possible to set extensive native Solr configuration using the admin webapp interface noted above. It should be accessible at the address: http://localhost:8080/solr/admin/ (where 8080 is your server's http port) Please refer to the Apache Solr documentation for usage of this interface and other native Solr configuration details.

1.3. Data Indexing

The solr plugin indexes data such as Products into the Apache Solr database using services defined in the file: servicesdef/solrservices.xml The initial indexing may need to be performed or scheduled manually, but subsequent indexing may be partially or fully automated, though automated methods are disabled by default and must be enabled. Note that in general, solr services can only

successfully run in contexts where the solr webapp is loaded and accessible. There are two methods for indexing data: Index rebuilding service (rebuildSolrIndex): The rebuildSolrIndex is the most important data import service. It reindexes all Ofbiz Products existing in the system into the solr index. rebuildSolrIndex MUST be run AT LEAST once after installation and also following any data load operation that loads new products using the Ofbiz "install" starting mode (run-install, load-demo, etc.). To do this, one can simply use the Webtools backend to invoke the service manually (e.g., /webtools/control/setSyncServiceParameters?SERVICE_NAME=rebuildSolrIndex&POOL_NAME=pool&RUN_SYNC=Y) or summon the Webtools Job Scheduler. Once the initial indexing has been performed, one can then use the Job Scheduler to invoke rebuildSolrIndex on a regular basis (every hour, every midnight, etc.) to update the Solr index. ECAs/SECAs (addToSolr, for Product data): Although the rebuildSolrIndex is always necessary for the initial data import, one may also use ECAs and SECAs to import subsequent data changes automatically at every individual data (e.g. Product) update instead of running rebuildSolrIndex periodically. This is done by defining ECAs or SECAs that trigger the addToSolr service. The addToSolr service simply accepts a single "instance" parameter, a GenericValue. At the time of this writing, any entity value having a valid "productId" field designating a Product value may be passed; this will trigger reindexing for the specific product. By default, the addToSolr service implementation (and any service intended for use with ECAs) is disabled globally and succeeds silently. It can be enabled by setting the "solr.eca.enabled=true" in "config/solrconfig.properties". This property is provided for easy toggling. It can also be specified on the command line or using ant through the true/false "ofbiz.solr.eca.enabled" system property (which has priority over file-based "solr.eca.enabled") in the same way described for the solr home system property (see Installation); this allows toggling them per Ant target. Some stock/default/example ECAs are provided and are functional once enabled. They can be found in: entitydef/eecas.xml Custom ECAs may also be added to this file, and custom SECAs may be added to the file: servicedef/secas.xml Special care may need to be taken to ensure the addToSolr is not triggered during "install" sequences (run-install/load-demo) because it requires the solr webapp to be accessible. addToSolr will automatically attempt to prevent reindexing in cases where the solr webapp is unavailable. However, it is possible that this behavior results in needlessly prevented reindexing (and thus lost updates) in some unknown complex server configurations. If this is the case, this behavior can be disabled by setting the option "solr.eca.useSolrWebappLoadedCheck=false" in the properties file "config/solrconfig.properties". However, if the above useSolrWebappLoadedCheck option is specified, you will have to manually disable or comment the ECA definitions/services to prevent them from executing during data-load operations; the "solr.eca.enabled" property and "ofbiz.solr.eca.enabled" system property can be used for this purpose. One easy way is to set "solr.eca.enabled=true" in the property files and then specify "-Dofbiz.solr.eca.enabled=false" for any relevant Ant build install/run-install/load-demo targets. Note that simply commenting the ECAs/SECAs or preventing their inclusion using ofbiz-component.xml may eliminate some service call overhead compared to using the property toggling. Ideally, a method exploiting ofbiz-component.xml or eecas.xml to prevent ECAs from running in inappropriate contexts would be ideal; however, at the time of this writing, such a method could not be established. * To reset and clear the solr index completely, it is sufficient to shut down the server and delete the folders: solr/data/index solr/data/spellchecker and any other such folders (must delete the whole folders, not only contents).

1.4. Data Querying

Solr queries can be done using two methods: Solr Ofbiz services: Simply invoke (manually or in code) the query services found in the file: servicesdef/solrservices.xml These include solrProductsSearch, solrKeywordSearch and others. Note that in general, solr services can only successfully run in contexts where the solr webapp is loaded and accessible. Solr native admin webapp interface: One can also perform native Solr queries and diagnostics using the standard admin interface, accessible as described under Configuration. Please refer to the Apache Solr documentation for usage of this interface.

1.5. Implementation Concerns

The structure of the solr/webapp directory closely mirrors the contents of the solr.war distribution; however, please note that it contains some Ofbiz-specific modifications such as those found in web.xml. The same is true for other folders such as solr/conf. Therefore, any library updates must be done with care to preserve these modifications.

1.6. Known Bugs, Limitations and Issues

In general, solr services can only successfully run in contexts where the solr webapp is loaded and accessible. ECA indexing may be problematic due to data loading and unknown complex server configurations; various toggling options are provided to work around this issue (see Data Indexing for details). In the future - and ideally - this could be addressed using ofbiz-component.xml (or eecas.xml) directives. The indexing services often produce the log warning "Problem reading product features" due to their use of the stock Ofbiz getProductFeatureSet service. * Please report any other issues encountered. Thank you.