Entity Engine Configuration Guide

Written by: David E. Jones
Minor corrections by Pawel H Debski and Les Austin

Table of Contents

Related Documents

Entity Engine Guide

Framework Configuration Guide

Service Engine Configuration Guide

Introduction

This document describes the configuration of the Entity Engine. It starts with an introduction of general ideas and then goes through each part of the entityengine.xml file and explains the available elements and their usage. The entityengine.xml file used for OFBiz applications has examples of a number of different options and is located in

framework/entity/config/entityengine.xml

The configuration of the Entity Engine is done through a simple XML file called **`entityengine.xml`** that must exist somewhere on the classpath. This file is used to define parameters for persistence servers such as EJB server parameters or JDBC server parameters. It is also used to specify which XML entity model, entity group, and field type model files will be used for that server. The default file for the OFBiz distribution can be found in

framework/entity/config/entityengine.xml

Each application that uses the Entity Engine does so through a **`Delegator`** instance. The delegator name must be passed to the static factory method that is then passed to the constructor if a new instance is needed. This delegator name is used to look up settings for that delegator in the **`entityengine.xml`** file. Each delegator uses an entity model reader and an entity group reader that specifies a group name for each named entity in the entity model file. The **`entityengine.xml`** file contains the settings that map each group name to a **`GenericHelper`** for that group. **`GenericHelper`** is an interface that must be implemented for each type of data source (i.e.: JDBC, EJB, SOAP, HTTP, et cetera).

The settings for each named **`GenericHelper`** are specified **`datsource`** elements in the **`entityengine.xml`** file. For a JDBC helper these would include database connection parameters such as either the JNDI data source parameters or the JDBC parameters including the driver name, the JDBC URI, and the username and password for the database. An EJB helper would contain JNDI parameters such as the context provider URL, the initial context factory, and the URL package prefixes.

The **`Delegator`** is the primary access method for Entity Engine services. Each service request is dispatched to the helper that corresponds to the entity that the service is requested for according to the group name of the entity specified in the entity group XML file and the helper for that group specified in the **`entityengine.xml`** file. The default entity group XML file for the OFBiz entity model can be found in **`ofbiz/commonapp/entitydef/entitygroup.xml`**.

Resource Loaders

The **`resource-loader`** tag is used to configure a named resource loader that can be used elsewhere to load XML and other resources. It has the following attributes.

Attribute Name	Required?	Description
name	Y	The name of the resource loader. Used in other tags in the "loader" attribute.
class	Y	The class to use that extends the abstract class org.ofbiz.entity.config.ResourceLoader. Available classes include FileLoader, UrlLoader, and ClasspathLoader, all in the same package as the ResourceLoader class.
prepend-env	N	The name of a Java environment property to put at the very beginning of the full location, before the prefix. This is optional.

prefix	N	A string to put before the location when making the full location. This is optional. If used will go after the prepended environment property and before the location specified for each resource.

JTA Elements

For JTA support the Entity Engine needs access to the **`javax.transaction.UserTransaction`** and optionally the **`javax.transaction.TransactionManager`** interface implementations for the TX Manager being used. These are retrieved through the **`org.ofbiz.entity.TransactionFactory`** class. This class uses the class specified with the **`class`** attribute of the **`transaction-factory`**. This class may be changed depending on which application server or transaction manager you are running OFBiz on.

The default TX Manager is JOTM from ObjectWeb and the factory class for JOTM is **`org.ofbiz.entity.transaction.JotmFactory`**. There is also a special class for Weblogic:

`org.ofbiz.entity.transaction.WeblogicFactory`,

which must be changed to include the Weblogic specific stuff and then compiled with weblogic.jar on the classpath.

The most widely useful transaction factory class is the **`org.ofbiz.entity.transaction.JNDIFactory`** class. This class uses some additional elements from the entityengine.xml file to locate the UserTransaction and TransactionManager objects in JNDI.

The **`user-transaction-jndi`** and **`transaction-manager-jndi`** tags are used to specify the object names in JNDI and the name of the JNDI Server to use, as configured above. Both tags have two required attributes: **`jndi-server-name`** and **`jndi-name`**. An example jndi-name for the UserTransaction object is **`java:comp/UserTransaction`** and for the TransactionManager object is **`java:comp/TransactionManager`**.

Delegator Elements

The GenericDelagator is created through a factory method that takes a String argument containing the delegator name. This delegator name is used to look up a **`delegator`** tag in the entityengine.xml file.

Attribute Name	Required?	Description
name	Y	The name of the Delegator. Used to look up this tag by delegator name.
entity-model-reader	Y	The name of the entity-model-reader to use for this delegator.
entity-group-reader	Y	The name of the entity-group-reader to use for this delegator.
entity-eca-reader	N	The name of the entity-eca-reader to use for this delegator. If not specified no Entity ECAs will be used.
distributed-cache-clear-enabled	N	Used to specify if the distributed cache clear should be enabled. If not specified defaults to "false". If set to true the other DCC attributes will be used.
distributed-cache-clear-class-name	N	Used to specify if the name of the class to use for DCC and that implements the distributed cache clear interface. If not specified defaults to "org.ofbiz.entityext.cache.EntityCacheServices", which is a good default for most cases that uses the Service Engine for configuration and remote calls.
distributed-cache-clear-user-login-id	N	Used to specify if the userLoginId (to use for any security checks) is needed related to distributed cache clear operation. If not specified defaults to "admin".

The delegator tag must contain one or more **`group-map`** tags specifying a datasource to use for each group of entities that the delegator will know about from the entity-group-reader. The delegator uses this file to assign a group to each entity. When an operation on an entity is performed it looks up the group and the datasource helper that corresponds to the group and uses that Helper to perform lower level Data Source operations. With this technique when the application is written it does not have to know which Helper is responsible for a given entity, and can therefore handle an entity or groups of entities assigned to different Data Sources with a simple configuration.

Entity Model XML Files

The **`entity-model-reader`** tag is used to configure each named entity model reader. The tag has a **`name`** attribute used to specify the entity model reader's name. Each reader may load from multiple resources that are each specified with a **`resource`** tag inside the main tag.

Each **`resource`** tag has two required attributes: **`loader`** which specifies which resource-loader to use and **`location`** which specifies the location that the resource-loader will use inside itself to load the resource.

Entify Group XML Files

The **entity-group-reader** tag is used to configure each named entity group reader. The tag has a **name** attribute used to specify the entity group reader's name. The group reader uses XML files to get the entity-group mappings.

The tag has two optional attributes: **loader** which specifies which resource-loader to use and **location** which specifies the location that the resource-loader will use inside itself to load the resource. There is also a sub-element named "resource" that has these same attributes and can be used to specify addition resources for addition files.

Entity ECA XML Files

The **entity-eca-reader** tag is used to configure each named entity ECA reader. The tag has a **name** attribute used to specify the entity ECA reader's name. The ECA reader uses XML files to get the entity ECA rule definitions. Each reader may load from multiple resources that are each specified with a **resource** tag inside the main tag.

Each **resource** tag has two required attributes: **loader** which specifies which resource-loader to use and **location** which specifies the location that the resource-loader will use inside itself to load the resource.

Field Type XML Files

The **field-type** tag is used to configure each named field type. The tag has a **name** attribute used to specify the field type's name. The group reader uses a single XML file to get the field type information.

The tag has two required attributes: **loader** which specifies which resource-loader to use and **location** which specifies the location that the resource-loader will use inside itself to load the resource.

Data Source Elements

Many datasources can be configured using one **datasource** tag for each datasource. This tag has the following attributes, and may contain the following sub-elements:

Attribute Name	Required?	Description
name	Y	That name of the datasource.
helper-class	Y	There can be many types of datasource helpers; the main one used is the JDBC/DAO helper. You can code you own helpers and use them by implementing the **org.ofbiz.entity.GenericHelper** interface. For JDBC/DAO helpers the class will be **org.ofbiz.entity.GenericHelperDAO**.
field-type-name	Y	The name of the field-type to use. Must match the name of an existing field-type tag as defined above.
schema-name	N	The name of the schema to use in the database. If not specified no schema name will be used. This is a case sensitive value and a different database may expect a different casing.
check-on-start	N	Check the datasource on startup? Must be true or false, defaults to true.
add-missing-on-start	N	Add missing entities and fields to the datasource on startup when checking is done? Must be true or false, defaults to false.
use-foreign-keys	N	Use/Create foreign keys for "one" relationships? Must be true or false, defaults to true.
use-foreign-key-indices	N	Use/Create indices for foreign keys (i.e. an index on the foreign key columns)? Note that creating foreign keys is not required for this to work and that indices are created for type "one" relationship definitions. Must be true or false, defaults to true.
check-fks-on-start	N	Check foreign keys at startup and add missing as needed? Must be true or false, defaults to false. Some databases have a hard time with this and do not return a full list of foreign keys resulting in duplicate foreign keys being added to the database.
check-fk-indices-on-start	N	Check foreign key indices at startup and add missing as needed? Must be true or false, defaults to false.
use-pk-constraint-names	N	Use constraint names for Primary Keys? Some databases have a problem with this, but work fine if they assign their own names. Must be true or false, defaults to true.
constraint-name-clip-length	N	Used to specify max length of a constraint name. Constraint names are clipped to this length. When playing with this watch for duplicate constraint names. Must be an integer, defaults to 30.
fk-style	N	Used to specify the foreign key syntax style, either naming the foreign key constraint, or naming the foreign key itself. Most databases use the name_constraint syntax, but SAP DB is an exception to that and there may be others. Must be either "name_constraint" or "name_fk". Defaults to name_constraint.

use-fk-initially-deferred	N	Used to specify whether or not to use the INITIALLY DEFERRED option available in many databases when creating foreign keys. Not all databases support this option. When enabled and supported the foreign keys will not be checked until a transaction is committed, as opposed to checking foreign keys as operations are done inside a transaction. Must be set to "true" or "false". Defaults to true.
join-style	N	Used to specify the syntax to use when doing table joins in view-entity operations. Many databases are adopting the ANSI JOIN standard, but before that was introduced theta joins were much more common. Two theta join styles are supported: Oracle and MS SQL. Must be "ansi", "ansi-no-parenthesis", "theta-oracle" or "theta-mssql". Defaults to "ansi".
use-indices	N	Used to specify whether or not declared indices/indexes should be created in the database. Must be "true" or "false", defaults to "true".
check-indices-on-start	N	Used to specify whether or not indices/indexes should be checked when the server starts. Will add missing declared indexes. Note that many JDBC drivers do not support this sort of meta-data check and this may cause problems. Must be "true" or "false", defaults to "false".
alias-view-columns	N	This is used to compensate for a variation seen in some JDBC drivers where column names returned for aliased fields (especially in view entities) be either be the alias name, or the full text of what makes up that alias name. Most databases return the alias name, so this will generally be set to true. Must be "true" or "false", defaults to "true".

Sub-Element Name	How Many	Description
sql-load-path	0 to many	Used to specify a list of full paths to directories that will be searched for XML and SQL files to import into the data source by the **`install`** page in the WebTools webapp. Each tag has two attributes: path for the path location, and prepend-env to optionally specify a Java environment property to prepend to the specified path.
inline-jdbc	0 or 1	Used to specify the JDBC parameters to be used either by the connection pool or if no connection pool is available then by directly loading the driver (very slow). You must specify either inline-jdbc or jndi-jdbc, but not both, for the datasource.
jndi-jdbc	0 or 1	Used to specify the jndi-server and jndi-name to get a Connection or XAConnection from JNDI. You must specify either inline-jdbc or jndi-jdbc, but not both, for the datasource.
ANY	0 or 1	Any tag may go inside the datasource tag to specify parameters for other GenericHelper implementations. These will not be checked at load time unless the DTD is modified to describe them.

The **`inline-jdbc`** tag has the following attributes:

Attribute Name	Required?	Description
jdbc-driver	Y	The JDBC driver class for the database.
jdbc-uri	Y	The URI used to specify the type and location of the database.
jdbc-username	Y	The username to connect to the database as.
jdbc-password	Y	The username's password.
isolation-level	N	This is used by Tyrex to specify the transaction isolation level. The standard JDBC transaction isolation levels are available: • "None" • "ReadCommitted" • "ReadUncommitted" • "RepeatableRead" • "Serializable"

The **`jndi-jdbc`** tag has the following attributes:

Attribute Name	Required?	Description
jndi-server-name	Y	The name of the JNDI Server to use as configured in this file with the jndi-server tag, described above.
jndi-name	Y	The name of the Connection or XAConnection object in JNDI.

The data source retrieved from JNDI should be pooled and transactional, with connections already enlisted if a JTA transaction has been started. It can be either a DataSource or an XADataSource object.

If the JNDI elements are not specified the ConnectionFactory will get the JDBC parameters from entityengine.xml file and try to use Tyrex for the transaction manager and connection pool. If Tyrex is not available the Entity Engine will hobble along creating a JDBC connection manually each time one is requested. A warning will be printed when this happens.
