<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.6. Platform Specific Code</td>
<td>25</td>
</tr>
<tr>
<td>6.7. Apache OFBiz Themes</td>
<td>25</td>
</tr>
<tr>
<td>6.7.1. What is a theme</td>
<td>25</td>
</tr>
<tr>
<td>6.7.2. How to define it</td>
<td>25</td>
</tr>
<tr>
<td>6.7.3. Structure of Theme.xml</td>
<td>25</td>
</tr>
<tr>
<td>6.7.4. The common-theme</td>
<td>29</td>
</tr>
<tr>
<td>6.7.5. Create your own theme</td>
<td>29</td>
</tr>
<tr>
<td>6.7.6. Backware compatibility with OFBiz 16.11 and above</td>
<td>31</td>
</tr>
<tr>
<td>7. Core APIs</td>
<td>36</td>
</tr>
<tr>
<td>8. Development environment</td>
<td>37</td>
</tr>
<tr>
<td>8.1. Setup your environment</td>
<td>37</td>
</tr>
<tr>
<td>8.1.1. Java SE</td>
<td>37</td>
</tr>
<tr>
<td>8.1.2. IDE</td>
<td>37</td>
</tr>
<tr>
<td>8.1.3. Database</td>
<td>37</td>
</tr>
<tr>
<td>8.2. Web tools</td>
<td>37</td>
</tr>
<tr>
<td>9. Testing</td>
<td>38</td>
</tr>
<tr>
<td>9.1. Unit Tests</td>
<td>38</td>
</tr>
<tr>
<td>9.2. Integration Tests</td>
<td>38</td>
</tr>
<tr>
<td>10. Deployment</td>
<td>39</td>
</tr>
<tr>
<td>11. Security</td>
<td>40</td>
</tr>
<tr>
<td>11.1. Passwords and JWT (JSON Web Tokens) usage</td>
<td>40</td>
</tr>
<tr>
<td>11.1.1. How are set and used passwords and JWT in Apache OFBiz</td>
<td>40</td>
</tr>
<tr>
<td>11.2. Impersonation</td>
<td>43</td>
</tr>
<tr>
<td>11.2.1. What is Impersonation in Apache OFBiz</td>
<td>43</td>
</tr>
<tr>
<td>11.3. CSRF defense</td>
<td>45</td>
</tr>
<tr>
<td>11.3.1. How is done the CSRF defense in Apache OFBiz and how to adapt it if needed</td>
<td>45</td>
</tr>
<tr>
<td>12. Appendices</td>
<td>47</td>
</tr>
<tr>
<td>13. From Mini Language to Groovy</td>
<td>48</td>
</tr>
<tr>
<td>13.1. Groovy DSL (dynamic scripting library)</td>
<td>48</td>
</tr>
<tr>
<td>13.1.1. How to get Groovy support in your IDE</td>
<td>48</td>
</tr>
<tr>
<td>13.1.2. Known Fields</td>
<td>48</td>
</tr>
<tr>
<td>13.2. Known Methods</td>
<td>49</td>
</tr>
<tr>
<td>13.3. Services</td>
<td>50</td>
</tr>
<tr>
<td>13.3.1. From MiniLang to Groovy</td>
<td>50</td>
</tr>
<tr>
<td>13.3.2. Getting started</td>
<td>51</td>
</tr>
<tr>
<td>13.4. Checking Fields</td>
<td>52</td>
</tr>
<tr>
<td>13.5. Setting Fields</td>
<td>53</td>
</tr>
<tr>
<td>13.6. Starting Services</td>
<td>54</td>
</tr>
<tr>
<td>13.7. Preparing Service Results</td>
<td>55</td>
</tr>
<tr>
<td>13.8. Database Communication</td>
<td>55</td>
</tr>
<tr>
<td>13.9. Permissions</td>
<td>59</td>
</tr>
</tbody>
</table>
1. Introduction

Welcome to the Apache OFBiz developer manual. This manual provides information to help with customizing and developing OFBiz. If you are new to OFBiz and interested in learning how to use it, you may want to start with the "Apache OFBiz User Manual".

OFBiz is a large system composed of multiple subsystems. This manual attempts to introduce the overall architecture and high level concepts, followed by a detailed description of each subsystem. In addition, the manual will cover topics necessary for developers including the development environment, APIs, deployment, security, and so on.

1.1. Main systems

OFBiz at its core is a collection of systems:

- A web server (Apache Tomcat)
- A web MVC framework for routing and handling requests.
- An entity engine to define, load and manipulate data.
- A service engine to define and control business logic.
- A widget system to draw and interact with a user interface.

On top of the above mentioned core systems, OFBiz provides:

- A data model shared across most businesses defining things like orders, invoices, general ledgers, customers and so on.
- A library of services that operate on the above mentioned data model such as "createBillingAccount" or "updateInvoice" and so on.
- A collection of applications that provide a user interface to allow users to interact with the system. These applications usually operate on the existing data model and service library. Examples include the "Accounting Manager" and "Order Manager".
- A collection of optional applications called "plugins" that extend basic functionality and is the main way to add custom logic to OFBiz.

1.2. Components

The basic unit in OFBiz is called "component". A component is at a minimum a folder with a file inside of it called "ofbiz-component.xml"

Every application in OFBiz is a component. For example, the order manager is a component, the accounting manager is also a component, and so on.

By convention, OFBiz components have the following main directory structure:
It is apparent from the above directory structure that each OFBiz component is in fact a full application as it contains entities, data, services, user interface, routing, tests, and business logic.

Both core OFBiz applications as well as plugins are nothing more than components. The only difference is that core applications reside in the "applications" folder whereas plugins reside in the "plugins" folder; also OFBiz does not ship with plugins by default.

### 1.3. Example workflow

Many basic concepts were explained so far. An example would help in putting all of these concepts together to understand the bigger picture. Let us take an example where a user opens a web browser and enters a certain URL and hits the enter key. What happens? It turns out answering this question is not quite simple because lots of things occur the moment the user hits "enter".

To try to explain what happens, take a look at the below diagram. Do not worry if it is not fully understandable, we will go through most of it in our example.
1.3.1. User enters URL

In the first step in our example, the user enters the following URL:

https://localhost:8443/accounting/control/findInvoices

If we break down this URL, we identify the following parts:

- localhost: Name of the server in which OFBiz is running
- 8443: Default https port for OFBiz
- accounting: web application name. A web application is something which is defined inside a component
- control: Tells OFBiz to transfer routing to the control servlet
- findInvoices: request name inside the control servlet

1.3.2. Control servlet takes over

The Java Servlet Container (tomcat) re-routes incoming requests through web.xml to a special OFBiz servlet called the control servlet. The control servlet for each OFBiz component is defined in controller.xml under the webapp folder.

The main configuration for routing happens in controller.xml. The purpose of this file is to map requests to responses.
Request Map

A request in the control servlet might contain the following information:

- Define communication protocol (http or https) as well as whether authentication is required.
- Fire up an event which could be either a piece of code (like a script) or a service.
- Define a response to the request. A response could either be another request or a view map.

So in this example, the findInvoices request is mapped to a findInvoices view.

View Map

A view map maps a view name to a certain view-type and a certain location.

View types can be one of:

- screen: A screen widget which translates to normal HTML.
- screenfop: A PDF screen designed with Apache FOP based constructs.
- screencsv: A comma separated value output report.
- simple-content; A special MIME content type (like binary files).
- ftl: An HTML document generated directly from a FreeMarker template.
- screenxls: An Excel spreadsheet.

In the findInvoices example, the view-map type is a normal screen which is mapped to the screen:
component://accounting/widget/InvoiceScreens.xml#FindInvoices

1.3.3. Widget rendered

Once the screen location is identified and retrieved from the previous step, the OFBiz widget system starts to translate the XML definition of the screen to actual HTML output.

A screen is a collection of many different things and can include:

- Other screens
- Decorator screens
- Conditional logic for hiding / showing parts of the screen
- data preparation directives in the <action> tag
- Forms
- Menus
- Trees
- Platform specific code (like FreeMarker for HTML output)
- Others (portals, images labels etc ...
Continuing the example, the FindInvoices screen contains many details including two forms. One form is for entering invoice search fields and the other form displays search results.
2. Web Framework
3. Web Applications

The OFBiz webapp is one of the core framework components. It is tightly integrated with other framework components.

3.1. Cross-domains Single Sign On (SSO)

In some cases you need to split the OFBiz applications on different servers, and possibly in production on different domains. This can happen for different reasons, most often for performance reason.

As it's annoying to give each time a credential when changing from an OFBiz application to another on the same server, the same applies when changing from an OFBiz application to another on another domain.

To prevent that on the same server, the ExternalLoginKey mechanism is used. The cross-domains SSO feature allows to navigate from a domain to another with automated SSO.

It based on 3 technologies:

JWT

JWT Official site - Wikipedia for JWT

CORS

CORS (Mozilla doc) - Wikipedia for CORS

Ajax

Ajax, now well known I guess, in OFBiz we use jQuery for that.

The mechanism is simple.

On the source side:

1. When an user log in in an application (webApp) a webappName.securedLoginId cookie is created. This cookie will be used by the mechanism to know the current logged in user. Note that all webappName.securedLoginId cookies are deleted when the user session is closed or time out. Hence (apart also using an intrinsically secured cookie) the mechanism is secured, even on shared machines. Of course if people are sharing a machine during their sessions, things could get complicated. This unlikely later case is not taken in account.

2. The user is given a JavaScript link which passes the URL to reach and the calling webapp name to the sendJWT() Ajax function.

3. The sendJWT() Ajax function calls the loadJWT() Ajax function which in turn calls the CommonEvents::loadJWT method through the common controller.

4. The CommonEvents::loadJWT method uses the calling webapp name to retrieve the userLoginId from the secured webappName.securedLoginId cookie, creates a JWT containing the userLoginId, and returns it to the loadJWT() Ajax function.

5. Then the sendJWT() Ajax function sends an Authorization header containing the JWT to the URL
to reach. At this stage, if all things are correct, the flow leaves the source side.

**On the server side:**

1. A CORS policy is needed. **Without it, the Authorization token containing the JWT will be rejected.** It's a simple policy but you need to strictly define the authorized domains. Never use the lazy "*" for domains (ie all domains), else the **preflight request** will not work. Here is an example for Apache HTTPD (domain value is "https://localhost:8443" for official OFBiz demo):

   ```
   Header set Access-Control-Allow-Origin domain
   Header set Access-Control-Allow-Headers "Authorization"
   Header set Access-Control-Allow-Credentials "true"
   ```

1. The checkJWTLogin preprocessor, similar to the checkExternalLoginKey, intercepts the JWT, checks it and if all is OK signs the user on. That’s it!

In the example component, the FormWidgetExamples screen contains 2 new fields in the LinksExampleForm which demonstrate the use from a local instance to the trunk demo instance.

If you are interested in more details you may refer to [https://issues.apache.org/jira/browse/OFBIZ-10307](https://issues.apache.org/jira/browse/OFBIZ-10307)

### 3.2. Control Servlet

#### 3.2.1. Requests

#### 3.2.2. Views
4. Entity Engine

4.1. Entities

4.1.1. Standard Entities

4.1.2. View Entities

4.1.3. Extended Entities

4.1.4. Dynamic View Entities

4.2. XML Data

4.3. Entity engine configuration

4.4. Supported databases

4.5. Data Model Changes

The Apache OFBiz® Project Release 17.12

Apache OFBiz follows The Universal Data Model by Len Silverston, with a grain of salt.

The following file contains information about the data model changes in the Apache OFBiz. The detailed description of migration scripts specified here can be found at Revisions Requiring Data Migration - upgrade ofbiz page.

4.5.1. Changes with OFBiz Trunk (Upcoming Branch)

Entity Changes

Added 1 new entity

1. ProdPromoCodeContactMech

Removed/Deprecate 1 entity

1. ProductPromoCodeEmail

Field Changes

<table>
<thead>
<tr>
<th>Entity</th>
<th>Field</th>
<th>Action</th>
<th>IsPK</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlXbrlClass</td>
<td>parentGlXbrlClassId</td>
<td>Added</td>
<td>No</td>
<td>36a123a</td>
</tr>
</tbody>
</table>
If in the list above fields are mentioned where 'IsPK' = yes, we advice to follow steps below for a successful upgrade:

1. Stop your OFBiz implementation
2. Go to the entity definition in the appropriate entitymodel.xml file
3. Disable temporarily the prim-key reference of the added field, and save the file
4. Restart the OFBiz implementation
5. Check with your RDBMS client that the new field is present in the table
6. Add values for the field in the table (if appropriate)
7. Stop the OFBiz implementation
8. Go back to the definition in the appropriate entitymodel.xml file
9. Enable the temporary disabled prim-key reference of the added field, and save the file
10. Check with your RDBMS client that the new field is part of the primary key definition of the table

Migration Scripts

1. Migration service migrateProductPromoCodeEmail is implemented to migrate the ProductPromoCodeEmail entity to ProductPromoCodeContactMech.
   (More detail at OFBIZ-5426)

4.5.2. Changes with OFBiz 17

Field types id-ne, id-long-ne & id-vlong-ne has been removed. Use id, id-long and id-vlong instead (detailed description at OFBIZ-9351).

Entity Changes

No changes

Field Changes

<table>
<thead>
<tr>
<th>Entity</th>
<th>Field</th>
<th>Action</th>
<th>IsPK</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarketingCampaign</td>
<td>fromDate</td>
<td>Added</td>
<td>Yes</td>
<td>R1805961</td>
</tr>
<tr>
<td>MarketingCampaign</td>
<td>thruDate</td>
<td>Added</td>
<td>No</td>
<td>R1805961</td>
</tr>
<tr>
<td>MarketingCampaign</td>
<td>fromDate</td>
<td>Added</td>
<td>Yes</td>
<td>R1805961</td>
</tr>
<tr>
<td>MarketingCampaign</td>
<td>thruDate</td>
<td>Added</td>
<td>No</td>
<td>R1805961</td>
</tr>
<tr>
<td>MarketingCampaign</td>
<td>fromDate</td>
<td>Added</td>
<td>Yes</td>
<td>R1805961</td>
</tr>
<tr>
<td>Entity</td>
<td>Field</td>
<td>Action</td>
<td>IsPK</td>
<td>Revision</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------</td>
<td>---------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>MarketingCampaignRole</td>
<td>thruDate</td>
<td>Added</td>
<td>No</td>
<td>R1805961</td>
</tr>
<tr>
<td>Product</td>
<td>manufacturerPartId</td>
<td>Removed</td>
<td>No</td>
<td>R1804408</td>
</tr>
<tr>
<td>SecurityGroupPermission</td>
<td>fromDate</td>
<td>Added</td>
<td>Yes</td>
<td>R1812383</td>
</tr>
<tr>
<td>SecurityGroupPermission</td>
<td>thruDate</td>
<td>Added</td>
<td>No</td>
<td>R1812383</td>
</tr>
</tbody>
</table>

**Migration Scripts**

1. Updated sql-type for date-time and time field in fieldtypemysql.xml file at commit R1793300
   
   *Update mysql sql-type for datetime field-type to support Fractional Seconds in Time Values Please upgrade mysql to at least 5.6.4 or higher.*
   
   After upgrade run `generateMySqlFileWithAlterTableForTimestamps` service, `groupName` is required field for this service.
   
   It will generate sql file with alter query statement for date-time and time field at location `${ofbiz.home}/runtime/tempfiles/.sql`
   
   You can use execute sql statement from any of the mysql batch command.

**4.5.3. Changes between OFBiz 9 to OFBiz 16**

**Entity Changes**

**Added 77 new entities**

1. JobRequisition
2. ProductAverageCostType
3. WorkEffortSurveyAppl
4. WorkEffortIcalData
5. WebSiteContactList
6. WebAnalyticsType
7. WebAnalyticsConfig
8. UserLoginSecurityQuestion
9. UomGroup
10. TrainingRequest
11. ThirdPartyLogin
12. TestFieldType
13. TestingSubtype
14. TestingStatus
15. TestingRemoveAll
16. TestingItem
17. TestingCrypto
18. SystemProperty
19. ShipmentGatewayUsps
20. ShipmentGatewayUps
21. ShipmentGatewayFedex
22. ShipmentGatewayDhl
23. ShipmentGatewayConfig
24. ShipmentGatewayConfigType
25. ReturnContactMech
26. QuoteNote
27. ProductPromoContent
28. ProductPromoContentType
29. ProductGroupOrder
30. ProductCostComponentCalc
31. CostComponentCalc
32. PayPalPaymentMethod
33. PaymentGroupType
34. PaymentGroup
35. PaymentGroupMember
36. PaymentGatewayConfig
37. PaymentGatewayConfigType
38. PaymentGatewayWorldPay
39. PaymentGatewaySecurePay
40. PaymentGatewaySagePay
41. PaymentGatewayOrbital
42. PaymentGatewayEway
43. PaymentGatewayCyberSource
44. PaymentGatewayAuthorizeNet
45. PaymentGatewayIDEAL
46. PaymentContentType
47. PaymentContent
48. OAuth2LinkedIn
49. OAuth2GitHub
50. JobManagerLock
51. JobInterviewType
52. JobInterview
53. JavaResource
54. InvoiceNote
55. InvoiceItemAssocType
56. InvoiceItemAssoc
57. InvoiceContentType
58. InvoiceContent
59. GlAccountCategoryType
60. GlAccountCategoryMember
61. GlAccountCategory
62. GitHubUser
63. FixedAssetTypeGlAccount
64. FacilityContent
65. ExcelImportHistory
66. EmplLeaveReasonType
67. EbayShippingMethod
68. EbayConfig
69. CountryAddressFormat
70. ContentSearchResult
71. ContentSearchConstraint
72. ContentKeyword
73. CheckAccount
74. AgreementFacilityAppl
75. AgreementContentType
76. AgreementContent

**Removed 8 entities**

1. DepreciationMethod
2. FixedAssetMaintMeter
3. OagisMessageErrorInfo
4. OagisMessageInfo
5. SalesOpportunityTrackingCode
6. SimpleSalesTaxLookup
### Field Changes

<table>
<thead>
<tr>
<th>Entity</th>
<th>Field</th>
<th>Action</th>
<th>IsPK</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcctgTransAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>AcctgTransEntry</td>
<td>inventoryItemId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>AcctgTransTypeAttribute</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>BenefitType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>BenefitType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>BudgetAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>BudgetItemAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>BudgetItemTypeAttribute</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>BudgetStatus</td>
<td>changeByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>BudgetTypeAttribute</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>CommunicationEventRole</td>
<td>statusId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>CommunicationEventType</td>
<td>contactMechTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ContactListCommunicationStatus</td>
<td>partyId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ContactListCommunicationStatus</td>
<td>messageId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ContactListCommunicationStatus</td>
<td>changeByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ContactMechAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ContactMechAttribute</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>DeductionType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>DeductionType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>DocumentAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Entity</td>
<td>Field</td>
<td>Action</td>
<td>IsPK</td>
<td>Revision</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>--------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>DocumentTypeAttr</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentApp</td>
<td>approverPartyId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentApp</td>
<td>jobRequisitionId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentAppSourceType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentAppSourceType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentAppSourceType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentAppSourceType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentType</td>
<td>partyId</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>EmploymentType</td>
<td>roleTypeId</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FinAccountAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FinAccountAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FinAccountTransaction</td>
<td>glReconciliationId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FinAccountTransaction</td>
<td>statusId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FinAccountTransactionTypeAttr</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FinAccountTransactionTypeAttr</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FinAccountStatus</td>
<td>changeByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FixedAsset</td>
<td>acquireOrderId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FixedAsset</td>
<td>acquireOrderItemSeqId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FixedAssetAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>FixedAssetTypeAttr</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>GlAccount</td>
<td>externalId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>GlAccount</td>
<td>openingBalance</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Entity</td>
<td>Field</td>
<td>Action</td>
<td>IsPK</td>
<td>Revision</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------</td>
<td>-----------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>GlReconciliation</td>
<td>createdDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>GlReconciliation</td>
<td>lastModifiedDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>GlReconciliation</td>
<td>statusId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>GlReconciliation</td>
<td>openingBalance</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>InventoryItemAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>InventoryItemStatus</td>
<td>changeByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>InvoiceAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>InvoiceItemAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>InvoiceItemTypeAttribute</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>InvoiceStatus</td>
<td>changeByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>InvoiceTypeAttribute</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>InvoiceTermAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>JobSandbox</td>
<td>currentRetryCount</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>JobSandbox</td>
<td>tempExprId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>JobSandbox</td>
<td>currentRecurrenceCount</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>JobSandbox</td>
<td>maxRecurrenceCount</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>JobSandbox</td>
<td>jobResult</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderAdjustment</td>
<td>amountAlreadyIncluded</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderAdjustment</td>
<td>isManual</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderAdjustment</td>
<td>oldPercentage</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderAdjustment</td>
<td>oldAmountPerQuantity</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderAdjustment</td>
<td>lastModifiedDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderAdjustment</td>
<td>lastModifiedByUserLogin</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Entity</td>
<td>Field</td>
<td>Action</td>
<td>IsPK</td>
<td>Revision</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------</td>
<td>---------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>OrderAdjustment Attribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderAdjustmentTypeAttr</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderItem</td>
<td>supplierProductId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderItem</td>
<td>cancelBackOrderDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderItem</td>
<td>changeByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderItemAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderItemShipGroup</td>
<td>facilityId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderItemShipGroup</td>
<td>estimatedShipDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderItemShipGroup</td>
<td>estimatedDeliveryDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderItemShipGrpInvRes</td>
<td>priority</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderItemShipGrpInvRes</td>
<td>oldPickStartDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderItemTypeAttr</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderTermAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderPaymentPreference</td>
<td>track2</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderPaymentPreference</td>
<td>swipedFlag</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderPaymentPreference</td>
<td>lastModifiedDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderPaymentPreference</td>
<td>lastModifiedByUserLogin</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderShipment</td>
<td>shipGroupSeqId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>OrderTypeAttr</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyAcctgPreference</td>
<td>orderSequenceEnumId</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Entity</td>
<td>Field</td>
<td>Action</td>
<td>IsPK</td>
<td>Revision</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>PartyAcctgPreference</td>
<td>quoteSequenceEnumId</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyAcctgPreference</td>
<td>invoiceSequenceEnumId</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyAcctgPreference</td>
<td>oldOrderSequenceEnumId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyAcctgPreference</td>
<td>oldQuoteSequenceEnumId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyAcctgPreference</td>
<td>oldInvoiceSequenceEnumId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyAcctgPreference</td>
<td>orderSeqCustMethId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyQual</td>
<td>infoString</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyQual</td>
<td>institutionInternalId</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyQual</td>
<td>institutionPartyId</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyQual</td>
<td>partyQualId</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyRate</td>
<td>percentageUsed</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyRate</td>
<td>rate</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PartyResume</td>
<td>contentId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PaymentAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PaymentGatewayResponse</td>
<td>gatewayCvResult</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PaymentMethod</td>
<td>finAccountId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PaymentTypeAttr</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PerfRatingType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PerfRatingType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PerfReview</td>
<td>payHistoryRoleTypeIdTo</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PerfReview</td>
<td>payHistoryRoleTypeIdFrom</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PerfReview</td>
<td>payHistoryPartyIdTo</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PerfReview</td>
<td>payHistoryPartyIdFrom</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PerfReview</td>
<td>payHistoryFromDate</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Entity</td>
<td>Field</td>
<td>Action</td>
<td>IsPK</td>
<td>Revision</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>PerfReviewItemType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PerfReviewItemType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PersonTraining</td>
<td>trainingRequestId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PersonTraining</td>
<td>workEffortId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PersonTraining</td>
<td>approverId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PersonTraining</td>
<td>approvalStatus</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PersonTraining</td>
<td>reason</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PostalAddress</td>
<td>houseNumber</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PostalAddress</td>
<td>houseNumberExt</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PostalAddress</td>
<td>cityGeoId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PostalAddress</td>
<td>municipalityGeoId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PostalAddress</td>
<td>geoPointId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PosTerminal</td>
<td>terminalName</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>PosTerminalInterTx</td>
<td>reasonEnumId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>releaseDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>originalImageUrl</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>inventoryItemTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>shippingWeight</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>productWeight</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>diameterUomId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>productDiameter</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>virtualVariantMethod</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>defaultShipmentBoxType</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>lotIdFilledIn</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>orderDecimalQuantity</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>weight</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>taxCategory</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Product</td>
<td>taxVatCode</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Entity</td>
<td>Field</td>
<td>Action</td>
<td>IsPK</td>
<td>Revision</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------</td>
<td>--------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>Product</td>
<td>taxDutyCode</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductAverageCost</td>
<td>productAverageCostTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductAverageCost</td>
<td>facilityId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductContent</td>
<td>sequenceNum</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductKeyword</td>
<td>keywordTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductKeyword</td>
<td>statusId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductRole</td>
<td>sequenceNum</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStore</td>
<td>balanceResOnOrderCreation</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStore</td>
<td>defaultTimeZoneString</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStore</td>
<td>oldStyleSheet</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStore</td>
<td>oldHeaderLogo</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStore</td>
<td>oldHeaderRightBackground</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStore</td>
<td>oldHeaderMiddleBackground</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStore</td>
<td>oldStyleSheet</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStore</td>
<td>headerLogo</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStore</td>
<td>headerRightBackground</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStore</td>
<td>headerMiddleBackground</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStorePaymentSetting</td>
<td>paymentCustomMethodId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStorePaymentSetting</td>
<td>paymentGatewayConfigId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStoreShipmentMethod</td>
<td>shipmentCustomMethodId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStoreShipmentMethod</td>
<td>shipmentGatewayConfigId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductStoreShipmentMethod</td>
<td>allowancePercent</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Entity</td>
<td>Field</td>
<td>Action</td>
<td>IsPK</td>
<td>Revision</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------</td>
<td>---------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>ProductStoreShipMentMeth</td>
<td>minimumPrice</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ProductTypeAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteAdjustment</td>
<td>lastModifiedDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteAdjustment</td>
<td>lastModifiedByUserLogin</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteItem</td>
<td>leadTimeDays</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteRole</td>
<td>fromDate</td>
<td>Added</td>
<td>Yes</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteRole</td>
<td>thruDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteTerm</td>
<td>termDays</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteTerm</td>
<td>textValue</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteTerm</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteTermAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>QuoteTypeAttribute</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>RequirementAttribute</td>
<td>changeByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>RequirementStatus</td>
<td>changeByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ResponsibilityType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ResponsibilityType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ReturnAdjustment</td>
<td>createdByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ReturnAdjustment</td>
<td>lastModifiedDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ReturnAdjustment</td>
<td>lastModifiedByUserLogin</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ReturnHeader</td>
<td>supplierRmaId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ReturnItemResponse</td>
<td>finAccountTransId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ReturnStatus</td>
<td>changeByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>SalaryStep</td>
<td>fromDate</td>
<td>Added</td>
<td>Yes</td>
<td>NA</td>
</tr>
<tr>
<td>SalaryStep</td>
<td>thruDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Entity</td>
<td>Field</td>
<td>Action</td>
<td>IsPK</td>
<td>Revision</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>SalaryStep</td>
<td>createdByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>SalaryStep</td>
<td>lastModifiedByUserLogin</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>SalesOpportunity</td>
<td>nextStepDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ServiceSemaphore</td>
<td>lockedByInstanceUserId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ShoppingListItem</td>
<td>modifiedPrice</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>SkillType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>SkillType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>SupplierProduct</td>
<td>shippingPrice</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>SupplierProduct</td>
<td>supplierCommissionPerc</td>
<td>Removed</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>TaxAuthorityRateProduct</td>
<td>isTaxInShippingPrice</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>TerminationType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>TerminationType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>TestingNodeMember</td>
<td>extendFromDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>TestingNodeMember</td>
<td>extendThruDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>TimeEntry</td>
<td>planHour</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Timesheet</td>
<td>approvedByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>TrainingClassType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>TrainingClassType</td>
<td>hasTable</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>UnemploymentClaim</td>
<td>thruDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>UserLogin</td>
<td>externalAuthId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>UserLogin</td>
<td>userLdapDn</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>UserLogin</td>
<td>disabledBy</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>ValueLinkKey</td>
<td>createdByUserLogin</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WebSite</td>
<td>visualThemeSetId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WebSite</td>
<td>hostedPathAlias</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WebSite</td>
<td>isDefault</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>Entity</td>
<td>Field</td>
<td>Action</td>
<td>IsPK</td>
<td>Revision</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------</td>
<td>---------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>WebSite</td>
<td>displayMaintenencePage</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WebSitePathAlias</td>
<td>fromDate</td>
<td>Added</td>
<td>Yes</td>
<td>R1738588</td>
</tr>
<tr>
<td>WebSitePathAlias</td>
<td>thruDate</td>
<td>Added</td>
<td>No</td>
<td>R1738588</td>
</tr>
<tr>
<td>WorkEffort</td>
<td>tempExprId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffort</td>
<td>sequenceNum</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffortAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffortAttribute</td>
<td>attrDescription</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffortTypeAttr</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffortContacMech</td>
<td>fromDate</td>
<td>Added</td>
<td>Yes</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffortContacMech</td>
<td>thruDate</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffortFixedAssetAssign</td>
<td>availabilityStatusId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffortPartyAssignment</td>
<td>assignedByUserLoginId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffortPurposeType</td>
<td>parentTypeId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffortStatus</td>
<td>reason</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkEffortTypeAttr</td>
<td>description</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
<tr>
<td>WorkOrderItemFulfillment</td>
<td>shipGroupSeqId</td>
<td>Added</td>
<td>No</td>
<td>NA</td>
</tr>
</tbody>
</table>
5. Service Engine

5.1. Declaration and Implementation

5.2. Supported languages

5.3. Transaction management

5.4. Web services
6. Widget System

6.1. Screen Widget

6.1.1. Decoration Pattern

6.2. Form Widget

6.3. Menu Widget

6.4. Tree Widget

6.5. Portal Widget

6.6. Platform Specific Code

6.7. Apache OFBiz Themes

The Apache OFBiz Project Release 17.12

6.7.1. What is a theme

A Theme is an ofbiz component that defines all elements necessary to render all information generated by the screen engine through an embedded technology. Currently themes presents in Apache OFBiz use html5/jquery/css to do that.

To realize that, a theme can define some properties, among them some can be necessary. It can define its own ftl macro to render the different modelScreen elements and can define its own screen decorator to prepare the final user screen structure for the technology used by the theme.

6.7.2. How to define it

A theme is a standard component, present in the directory themes or plugins with a file definition present on widget/Theme.xml and support the widget-theme.xsd

To offer the possibility for end users to select the theme through the "Select Theme Screen", the theme need to load the entity VisualTheme and one or more entries related to this theme.

6.7.3. Structure of Theme.xml

The theme definition file help OFBiz to know what specific rendering the Theme want to use. It's composed by two definition elements and four optional blocks
Main definition

The First mandatory element is one or several visualThemeId related to the theme Defined like that

```xml
<visual-themes>
  <visual-theme id="MY_THEME"/>
</visual-themes>
```

You need to define in the database an entry in VisualTheme Entity for each visual-theme id defined.

```xml
<entity-engine-xml>
  <VisualTheme visualThemeId="MY_THEME" visualThemeSetId="BACKOFFICE"
               description="My theme - Example (based on flatgrey)"/>
</entity-engine-xml>
```

A theme component can load one or more visual theme id. Usually only one is present.

The second important (but not mandatory) element is implements

```xml
<extends location="component://common-theme/widget/Theme.xml"/>
```

This element indicates that your theme copies from the extend theme all information not present in its file definition.

If this element isn't present in your theme, you will need to define all information present in common-theme to be sure that OFBiz misses nothing for a correct run. Otherwise some functionnalities can be broken...

It's highly recommended to extend the common-theme to be sure that your theme works correctly and to surcharge only what you need.

The four following blocks are optionnal if you define an extends theme

General properties

This block contains all properties that the screen engine can use to prepare the rendering and that the theme can implement
<widget-properties> <!--Transversal properties relative to ofbiz widget component--> 
  <default-view-size value="20"/>
  <autocompleter 
    default-view-size="10"
    default-min-lenght="2"
    default-delay="300"
    display-return-field="true"/>
  <lookup 
    position="topleft"
    width="640"
    height="500"/>
  <layered-modal 
    width="800"
    height="600"/>
</widget-properties>

**Theme's specific properties**

This block contains all properties specific to this theme. In general these are some properties present on ftl template, that are initialized by the theme and can be surchaged by another theme through the extends elements.

```
<property name="jgrowlPosition" value="center" type="String"/> <!--possible value: top-left, top-right, bottom-left, bottom-right, center-->
<property name="jgrowlWidth" value="800" type="Integer"/>
<property name="jgrowlHeight" value="" type="Integer"/>
<property name="jgrowlSpeed" value="100" type="Integer"/>
```

**Ftl macro library**

This block defines for each technology implemented by the screen engine where it can find the macro library for each model (Screen, Form, Tree, Menu)
If you want surcharge some macros, you can just create the desired macros and import the others from common-theme (at the top of file) like that:

```html
<#include "component://common-theme/template/macro/HtmlFormMacroLibrary.ftl"/>
```

**Screen library**

This block defines where OFBiz can find all official screens definitions in framework/common

Normally, you don’t need to change this file except if you need to define a default screen style that doesn’t exist OOTB. If you need to extend an existing one, you have to do it in the theme directory.

To define a new default screen style, you have to add it in this file, and point to the screen decorator in common-theme to define your default screen style as the default one.
Screens are separated in three types:

- structural-decorator: contains all decorators that organise the screens structures
- embed-decorator: decorator used only on sub screens
- general-screen: list all generic inter applications screens

6.7.4. The common-theme

This is the root theme that contains all information to ensure a good basic theme for OFBiz. Currently it keeps all old themes system for backward compatibility with ftl template managed by the entity VisualThemeResource

6.7.5. Create your own theme

As a theme is a component, you can create a new theme like a plugin.

After creating a component, you can add the two minimal information:

- Theme.xml file in plugins/my-theme/widget/ with minimal information:

```xml
<theme name="my-theme"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="http://ofbiz.apache.org/dtds/widget-theme.xsd">
   <visual-themes>
     <visual-theme id="MY_THEME" display-name="My Theme"/>
   </visual-themes>
</theme>
```

- your data file to add your visual theme in plugins/my-theme/data/

```xml
<entity-engine-xml>
   <VisualTheme visualThemeId="MY_THEME" visualThemeSetId="BACKOFFICE"/>
</entity-engine-xml>
```

The presence of VisualTheme entity helps to indicate which theme is available in your instance, specially helpful for tenant installations.

To display your theme in OFBiz theme library, you can complete the information on each visual theme like
display-name and description support the flexibleStringExpander syntax

extends common-theme

This is a first step to understand how the theme system works. With your new theme, you can try to surcharge different elements.
To start, extends the common-theme:

```xml
<theme name="my-theme"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="http://ofbiz.apache.org/dtds/widget-theme.xsd">
  <visual-themes>
    <visual-theme id="MY_THEME" display-name="My Theme">
      <description>My new funny theme under nice tecnno</description>
      <screenshot location="/mytheme/screenshot1.png"/>
      <screenshot location="/mytheme/screenshot2.png"/>
    </visual-theme>
  </visual-themes>
</theme>
```

Now your theme should be operational, but without particularity.

You can surcharge a ftl macro, to do this create your own ftl macro file in plugins/my-theme/templates/macro/HtmlFormMacroLibrary.ftl with
Now indicate to your theme that you want use this library

```xml
<theme name="my-theme"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="http://ofbiz.apache.org/dtds/widget-theme.xsd">
 <visual-themes>
  <visual-theme id="MY_THEME" display-name="My Theme">
   <description>My new funny theme under nice techno</description>
   <screenshot location="/mytheme/screenshot1.png"/>
  </visual-theme>
  <visual-theme/>
  <extends location="component://common-theme/widget/Theme.xml"/>
 <templates>
  <template name="screen" type="html" content-type="UTF-8" encoding="none"
   encoder="html" compress="false">
   <template-file widget="form" location="component://my-
theme/template/macro/HtmlFormMacroLibrary.ftl"/>
  </template>
 </templates>
 </theme>
```

and check the result when you select your theme. The result isn’t really interesting but it’s to understand how it works.

create from scratch

TODO...

**6.7.6. Backware compatibility with OFBiz 16.11 and above**

**How themes worked before**

Before the theme management by model definition, all configurations have been present in the database through entity `VisualTheme` and `VisualThemeRessource`. These ressources were loaded in a `layoutProperties` variable and used directly by decorator screens and ftl templates.
Now with the common-theme

All this logic is still present in the common-theme template to keep backward compatibility, but the VisualThemeResource is now useless and properties have been migrated to the Theme definition in the part theme-properties

Example with BlueLight

The blue light theme has been these properties in VisualThemeResource:

```xml
<VisualTheme visualThemeId="BLUELIGHT" visualThemeSetId="BACKOFFICE"
  description="BlueLight Theme: breadcrumbs, drop-down menus and rounded corners"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_NAME"
  resourceValue="BLUELIGHT" sequenceId="01"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_IMAGE_URL"
  resourceValue="/images/ofbiz_logo.png" sequenceId="01"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_SHORTCUT_ICON"
  resourceValue="/images/ofbiz.ico" sequenceId="01"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_SCREENSHOT"
  resourceValue="/bluelight/screenshot.jpg" sequenceId="01"/>

<!-- CSS references -->
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_STYLESHEET"
  resourceValue="/bluelight/style.css" sequenceId="01"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HELPSTYLESHEET"
  resourceValue="/bluelight/help.css" sequenceId="01"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_DOCBOOKSTYLESHEET"
  resourceValue="/bluelight/webapp/bluelight/docbook.css" sequenceId="01"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_STYLESHEET"
  resourceValue="/common/js/jquery/plugins/asmselect/jquery.asmselect-1.0.4a-beta.css"
  sequenceId="02"/>

<!-- Javascript references -->
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT"
  resourceValue="/common/js/jquery/jquery-1.11.0.min.js" sequenceId="01"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT"
  resourceValue="/common/js/jquery/jquery-migrate-1.2.1.js" sequenceId="02"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT"
  resourceValue="/common/js/jquery/ui/js/jquery-ui-1.10.3.min.js" sequenceId="03"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT"
  resourceValue="/common/js/jquery/plugins/asmselect/jquery.asmselect-1.0.4a-beta.js"
  sequenceId="05"/>
```
<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/jquery/plugins/datetimepicker/jquery-ui-timepicker-addon.min-1.4.3.js" sequenceId="07"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/jquery/plugins/fjTimer/jquerytimer-min.js" sequenceId="09"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/jquery/plugins/mask/jquery.mask-1.14.13.min.js" sequenceId="10"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/jquery/plugins/jeditable/jquery.jeditable-1.7.3.js" sequenceId="11"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/jquery/plugins/validate/jquery.validate.min.js" sequenceId="12"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/plugins/OpenLayers-2.13.1.js" sequenceId="13"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/util/OfbizUtil.js" sequenceId="15"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/util/fieldlookup.js" sequenceId="16"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/plugins/date/date.format-1.2.3-min.js" sequenceId="17"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/plugins/date/date.timezone-min.js" sequenceId="18"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/util/miscAjaxFunctions.js" sequenceId="19"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/common/js/plugins/date/FromThruDateCheck.js" sequenceId="22"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceTypeEnumId="VT_HDR_JAVASCRIPT" resourceValue="/bluelight/dropdown.js" sequenceId="30"/>

<!-- ftl references -->

<VisualThemeResource visualThemeId="BLUELIGHT"/>
<VisualThemeResource visualThemeId="BLUELIGHT" resourceValue="component://bluelight/template/Header.ftl" sequenceId="01"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceValue="component://bluelight/template/Footer.ftl" sequenceId="01"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceValue="component://bluelight/template/AppBarOpen.ftl" sequenceId="01"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceValue="component://bluelight/template/AppBarClose.ftl" sequenceId="01"/>

<VisualThemeResource visualThemeId="BLUELIGHT" resourceValue="component://bluelight/template/Messages.ftl" sequenceId="01"/>

Now it’s just

<VisualTheme visualThemeId="BLUELIGHT" visualThemeSetId="BACKOFFICE"/>

And on theme definition

<theme-properties>
  <!--javascript lib-->
  <property name="VT_HDR_JAVASCRIPT['add']" value="/bluelight/dropdown.js" sequenceId="30"/>
  <!--Css style-->
  <property name="VT_STYLESHEET['add']" value="/bluelight/style.css"/>
  <property name="VT_HELPSTYLESHEET['add']" value="/bluelight/help.css"/>
  <property name="VT_DOCBOOKSTYLESHEET['add']" value="/bluelight/webapp/bluelight/docbook.css"/>
  <!--template location-->
  <property name="VT_HDR_TMPLT_LOC" value="component://bluelight/template/Header.ftl"/>
  <property name="VT_FTR_TMPLT_LOC" value="component://bluelight/template/Footer.ftl"/>
  <property name="VT_NAV_OPEN_TMPLT" value="component://bluelight/template/AppBarOpen.ftl"/>
  <property name="VT_NAV_CLOSE_TMPLT" value="component://bluelight/template/AppBarClose.ftl"/>
  <property name="VT_MSG_TMPLT_LOC" value="component://bluelight/template/Messages.ftl"/>
</theme-properties>

Values with /images/... have been moved to the common-theme that bluelight extends, the theme definition keeps only what the theme adds to the extended theme.
property name supports the FlexibleMapAccessor syntax, so you can continue to populate a list (VT_STYLESHEET['add']), reset a list (VT_STYLESHEET[]) or add an element on the top list (VT_STYLESHEET[+0]) because some time the order libraries loading is important

Migrate you own theme

Easily, create you Theme.xml and move your VisualThemeResource in theme-properties like in the BlueLight example above. Maybe you will need to update your template because the modelTheme return ressources not always as list. So:

<property name="VT_HDR_TMPLT_LOC" value="component://bluelight/template/Header.ftl"/> → return a String with component://bluelight/template/Header.ftl
<property name="VT_STYLESHEET['add']" value="..." → return a List<String>
7. Core APIs
8. Development environment

8.1. Setup your environment

8.1.1. Java SE

8.1.2. IDE

Eclipse

Intellij Idea

8.1.3. Database

8.2. Web tools
9. Testing

9.1. Unit Tests

9.2. Integration Tests
10. Deployment
11. Security

11.1. Passwords and JWT (JSON Web Tokens) usage

11.1.1. How are set and used passwords and JWT in Apache OFBiz

The Apache OFBiz Project Release 17.12

Passwords

Demo and seed passwords are stored in files loaded through security ofbiz-component.xml. To know more about that be sure to read:

- The technical production setup guide notably "Initial Data Loading" and "Security Settings" sections
- How to secure your deployment

These configuration steps are not to be neglected for the security of a production environment

JWT usage

As says Wikipedia:

JSON Web Token (JWT) is an Internet standard for creating JSON-based access tokens that assert some number of claims.

We currently use JWT in 2 places:

1. To let users safely recreate passwords (in backend and frontend)
2. To allow SSO (Single Sign-on) jumpings from an OFBiz instance to another on another domain, by also using CORS (Cross-origin resource sharing) on the target server

How to secure JWT

When you use JWT, in order to sign your tokens, you have the choice of using a sole so called secret key or a pair of public/private keys: https://jwt.io/introduction/.

You might prefer to use pair of public/private keys, for now by default OFBiz uses a simple secret key. Remains the way how to store this secret key. This is an interesting introduction about this question.

1. The first idea which comes to mind is to use a property in the security.properties file. It’s safe as long as your file system is not compromised.
2. You may also pick a SystemProperty entity (overrides the file property). It’s safe as long as your DB is not compromised.
3. We recommend to not use an environment variable as those can be considered weak:
   - http://movingfast.io/articles/environment-variables-considered-harmful
   - https://security.stackexchange.com/questions/49725/is-it-really-secure-to-store-api-keys-in-environment-variables

4. You may want to tie the encryption key to the logged in user. This is used by the password recreation feature. The JWT secret key is salted with a combination of the current logged in user and her/his password. This is a simple and effective safe way.

5. Use a JTI (JWT ID). A JTI prevents a JWT from being replayed. This auth0 blog article get deeper in that. The same is kinda achieved with the password recreation feature. When the user log in after the new password creation, the password has already been changed. So the link (in the sent email) containing the JWT for the creation of the new password can’t be reused.

6. Tie the encryption key to the hardware. You can refer to this Wikipedia page for more information.

7. If you want to get deeper in this get to this OWASP documentation

Note: if you want to use a pair of public/private keys you might want to consider leveraging the Java Key Store that is also used by the "catalina" component to store certificates. Then don’t miss to read:

Also remember that like everything a JWT can be attacked and, though not used or tried in OFBiz yet, a good way is to mitigate an attack by using a KeyProvider. I have created OFBIZ-11187 for that.

Properties

The security.properties file contains five related properties:

```properties
# -- If false, then no externalLoginKey parameters will be added to cross-webapp urls
security.login.externalLoginKey.enabled=true

# -- Security key used to encrypt and decrypt the autogenerated password in forgot password functionality.
#    Read Passwords and JWT (JSON Web Tokens) usage documentation to choose the way you want to store this key
login.secret_key_string=login.secret_key_string

# -- Time To Live of the token send to the external server in seconds
security.jwt.token.expireTime=1800
```
# -- Enables the internal Single Sign On feature which allows a token based login between OFBiz instances
# -- To make this work you also have to configure a secret key with security.token.key
security.internal.sso.enabled=false

# -- The secret key for the JWT token signature. Read Passwords and JWT (JSON Web Tokens) usage documentation to choose the way you want to store this key
security.token.key=security.token.key

There are also SSO related SystemProperties in SSOJWTDemoData.xml:

```xml
<SystemProperty systemResourceId="security" systemPropertyId="security.internal.sso.enabled" systemPropertyValue="false"/>
<SystemProperty systemResourceId="security" systemPropertyId="security.token.key" systemPropertyValue="security.token.key"/>
<SystemProperty systemResourceId="security" systemPropertyId="SameSiteCookieAttribute" systemPropertyValue="strict"/>
```

**Internal SSO**

The introduction of the same-site attribute set to 'strict' for all cookies prevents the internal Single Sign On feature. Why is clearly explained [here](#).

So same-site attribute set to 'none' is necessary for the internal SSO to work, *[lax] is not enough*. So if someone wants to use the internal SSO feature s/he also needs to use the CSRF token defense. If s/he wants to be safe from CSRF attacks. Unfortunately, due backporting difficulties, this option is currently (2020-04-15) only available in trunk.

**Fetch API**

An alternative would be to use the Fetch Javascript API with the

```javascript
credentials: "include"
```

option to enable CORS. [Here is an example](#)

For those interested, there are more information in [https://issues.apache.org/jira/browse/OFBIZ-11594](https://issues.apache.org/jira/browse/OFBIZ-11594)

**Last but not least**

Be sure to read [Keeping OFBiz secure](#)
11.2. Impersonation

11.2.1. What is Impersonation in Apache OFBiz

The Apache OFBiz Project Release 17.12

Introduction to User impersonation

User Impersonation is a feature that offer a way to select a user login and impersonate it, i.e. see what the user could see navigating through the application in his name.

How do this work ?

An authorized user (see security and controls section for configuration), can select a user that will be impersonated.

The impersonation start, if everything is well configured, in current application (partymgr for the demo). Everything appears like if we were logged in with the userLoginId and the valid password (though we know nothing about it)

The only thing showing that we currently are impersonating a user is the little bottom-right image :

This icon indicates, when clicking on it, the user impersonated, and offer a way to depersonate.

The impersonate period is stored for audit purpose, and if the impersonator forgot to depersonate, the period is terminated one hour after impersonation start.

Security

This feature can draw some concerns about security aspect. This paragraph will introduce every controls and properties that have been implemented around the impersonation feature.

These configuration steps are not to be neglected for a production environment since this feature offer a way to act in place of another user.

Properties

The security.properties file introduce two properties that control impersonation feature :

```
security.disable.impersonation = true
```

This property, set by default to true, controls the activation of impersonation feature. If no configuration is done any user trying to use impersonation will face an error message, indicating that the feature is disabled.

To enable impersonation this property need to be set to false
This property controls the way impersonation occurred to the impersonated user:

In default configuration, the impersonated user see nothing and can use the application without knowing that he is currently impersonated. Several authorized user can impersonate a same login without any issue.

This configuration is intended for testing/QA environment allowing any authorized user to impersonate a login to validate its configuration, test the application etc.

Set to `true`, this configuration improve the control of the data generated by the impersonated user. Indeed, Only one authorized user can impersonate a login at the same time, and during the impersonation process, the impersonated user is unable to act within the application.

Since the impersonation period is stored in database, the actions done by the authorized user can be identified if there is the need to do so.

This configuration is intended for production environment

Controls

The permission

First, to be able to use impersonation, a user need to possess `IMPERSONATE_ADMIN` permissions. Demo data offer `IMPERSONATION` security group for this purpose.

In demo data, `FULLADMIN` security group also possess the permission.

Permission based user restriction

An authorized user cannot impersonate any user. There are two main controls that will restrict the impersonation feature.

Cannot impersonate Admin user

It is impossible to impersonate a user that is granted any of the admin permission:

```
"IMPERSONATE_ADMIN"
"ARTIFACT_INFO_VIEW"
"SERVICE_MAINT"
"ENTITY_MAINT"
"UTIL_CACHE_VIEW"
"UTIL_DEBUG_VIEW"
```

Cannot impersonate more privileged user

It is impossible to impersonate a user that has more permission than your user. Even if the missing permission is a minor one.
11.3. CSRF defense

11.3.1. How is done the CSRF defense in Apache OFBiz and how to adapt it if needed

The Apache OFBiz Project Release 17.12

The same-Site attribute

The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

— According to OWASP ZAP

By default OOTB the SameSiteFilter property sets the same-site attribute value to 'strict'. SameSiteFilter allows to change to 'lax' if needed. If you use 'lax' we recommend that you set the csrf.defense.strategy property to org.apache.ofbiz.security.CsrfDefenseStrategy in order to provide an effective defense against CSRF attacks.

Properties

The `security.properties` file contains related properties:

```java
# -- By default the SameSite value in SameSiteFilter is 'strict'.
# -- This property allows to change to 'lax' if needed.
# -- If you use 'lax' we recommend that you set
# -- org.apache.ofbiz.security.CsrfDefenseStrategy
# -- for csrf.defense.strategy (see below)
SameSiteCookieAttribute=
```

```java
# -- The cache size for the Tokens Maps that stores the CSRF tokens.
# -- RemoveEldestEntry is used when it's get above csrf.cache.size
# -- Default is 5000
# -- TODO: possibly separate tokenMap size from partyTokenMap size
csrf.cache.size=
```

```java
# -- Parameter name for CSRF token. Default is "csrf" if not specified
csrf.tokenName.nonAjax=
```
# -- The csrf.entity.request.limit is used to show how to avoid cluttering the Tokens Maps cache with URIs starting with "entity/
# -- It can be useful with large Database contents, ie with a large numbers of tuples,
like "entity/edit/Agreement/10000, etc.
# -- The same principle can be extended to other cases similar to "entity/" URIs
(harced or using similar properties).
# -- Default is 3
csrf.entity.request.limit=

# -- CSRF defense strategy.
# -- Because OFBiz OOTB also sets the SameSite attribute to 'strict' for all cookies,
# -- which is an effective CSRF defense,
# -- default is org.apache.ofbiz.security.NoCsrfDefenseStrategy if not specified.
# -- Use org.apache.ofbiz.security.CsrfDefenseStrategy
# -- if you need to use a 'lax' for SameSiteCookieAttribute
csf.defense.strategy=

There is also a SystemProperty in SSOJWTDemoData.xml:

```xml
<SystemProperty systemResourceId="security" systemPropertyId="SameSiteCookieAttribute" systemPropertyValue="strict"/>
```
12. Appendices
13. From Mini Language to Groovy

This is a small guide for everybody involved in converting the Mini Language into Groovy.

Why is this important?
This tutorial is directly linked to the efforts of converting all scripts in Mini Language to newer Groovy Scripts. All of this is done, because Groovy is much more readable and easier to review, more up to date and many other reasons, which can be found here: Proposal for deprecating Mini Language

To contribute, or just be up to date with the current process, you can look at the existing JIRA issue OFBIZ-9350 - Deprecate Mini Lang

13.1. Groovy DSL (dynamic scripting library)

13.1.1. How to get Groovy support in your IDE

The following paragraph is for Eclipse users.

It is possible to get Groovy support in Eclipse by converting the loaded project to a Groovy Project. The project itself will work as before.

To do this just follow these few steps:

1. Right-click on the project that has to be converted
2. Click on "Configure"
3. Click on "Convert to Groovy Project"

Eclipse will automatically load the file OfbizDslDescriptorForEclipse.dsld, in which the known fields and methods used in Groovy Scripts are defined.

13.1.2. Known Fields

```
property name: 'parameters'
type : 'java.util.Map'
```
These are the parameters given to the Groovy Script, when it is called as a service. It is equivalent to \texttt{Map}\textless String, Object\textgreater context in the Java-Service-Definition.

```
property name: 'context'
type: 'java.util.Map'
```
More parameters, which are, for example, given through a screen or another Groovy Script. This is important when the script is called through an action segment of a screen.

```
property name: 'delegator'
type: 'org.apache.ofbiz.entity.Delegator'
```
Normal instance of the Delegator, which is used for special database access.
property name: 'dispatcher'
type: 'org.apache.ofbiz.service.LocalDispatcher'
Normal instance of the LocalDispatcher, which is used to call services and other service-like operations.

property name: 'security'
type: 'org.apache.ofbiz.security.Security'
Normal instance of the Security-Interface with which permission checks are done.

13.2. Known Methods

method name: 'runService'
type: 'java.util.Map'
params: [serviceName: 'String', inputMap: 'java.util.Map']
Helping method to call services instead of dispatcher.runSync(serviceName, inputMap). Also possible: run service: serviceName, with: inputMap

method name: 'makeValue'
type: 'java.util.Map'
params: [entityName: 'String']
Helping method to make a GenericValue instead of delegator.makeValue(entityName). Creates an empty GenericValue of the specific entity.

method name: 'findOne'
type: 'java.util.Map'
params: [entityName: 'String', inputMap: 'java.util.Map']
Helping method to find one GenericValue in the database. Used instead of delegator.findOne(entityName, inputMap)

method name: 'findList'
type: 'java.util.List'
params: [entityName: 'String', inputMap: 'java.util.Map']
Helping method to find many GenericValue in the database. Used instead of delegator.findList(entityName, inputMap, null, null, null, false)

method name: 'select'
type: 'org.apache.ofbiz.entity.util.EntityQuery'
params: [entity: 'java.util.Set']
Helping method used instead of EntityQuery.use(delegator).select(...)

As above.

method name: 'from'
type: 'org.apache.ofbiz.entity.util.EntityQuery'
params: [entity: 'java.lang.Object']
Helping method used instead of EntityQuery.use(delegator).from(…)
method name: 'success'
type: 'def'
params: [message: 'String']
Helping method used instead of ServiceUtil.returnSuccess(message)

method name: 'failure'
type: 'java.util.Map'
params: [message: 'String']
Helping method used instead of ServiceUtil.returnFailure(message)

method name: 'error'
type: 'def'
params: [message: 'String']
Helping method used instead of ServiceUtil.returnError(message)

method name: 'logInfo'
type: 'void'
params: [message: 'String']
Helping method used instead of Debug.logInfo(message, fileName)

method name: 'logWarning'
type: 'void'
params: [message: 'String']
Helping method used instead of Debug.logWarning(message, fileName)

method name: 'logError'
type: 'void'
params: [message: 'String']
Helping method used instead of Debug.logError(message, fileName)

method name: 'logVerbose'
type: 'void'
params: [message: 'String']
Helping method used instead of Debug.logVerbose(message, fileName)

The actual definition of the methods can be found in `/framework/service/src/main/java/org/apache/ofbiz/service/engine/GroovyBaseScript.groovy`, the variables dctx, dispatcher and delegator are set in the file GroovyEngine.java which can be found in the same location.

13.3. Services

13.3.1. From MiniLang to Groovy

To see additional examples and finished conversions, which may help with occurring questions, click: [OFBiz-9350 - Deprecate Mini Lang](#) There is a chance that a similar case has already been converted.

When a simple-method ends, it will automatically at least return a success-map.
All the Groovy Services have to return success at least, too.

```groovy
return success()
```

### 13.3.2. Getting started

MiniLang files consist of services, which, in most cases, implement services.

The get converted to Groovy like the following:

```xml
<!-- This is MiniLang -->
<simple-method method-name="createProductCategory" short-description="Create an ProductCategory">
  <!-- Code -->
</simple-method>

// This is the converted Groovy equivalent
/**
 * Create an ProductCategory
 */
def createProductCategory() {
  // Code
}
```

It will be useful for future developers, and everybody who has to check something in the code, to put at least the short-description as the new Groovydoc. This will hopefully more or less explain, what the method should or shouldn't do. If the short-description isn't helpful enough, feel free complete it.

The structure of if and else in MiniLang is a little different than the one from Groovy or Java and can be a bit confusing when first seen, so here is an example:

```xml
<if-empty field="parameters.productCategoryId">
  <sequenced-id sequence-name="ProductCategory" field="newEntity.productCategoryId" />
</if-empty>
<else>
  <set field="newEntity.productCategoryId" from-field="parameters.productCategoryId" />
  <check-id field="newEntity.productCategoryId" />
  <check-errors />
</else>
</if-empty>
```

Notice, that the else always starts before the if-tag is closed, but sometimes isn’t indented as one would expect it.
When navigating through bigger if-phrases, the navigation itself will be much easier through just clicking in the opening or closing if-tag: Eclipse will automatically mark the matching opening or closing if-tag for you.

There are two possibilities to initialize a field/variable in Groovy.

1. To define a field/variable with its correct typing
   ```java
   String fieldName = "value"
   ```

2. To just "define" a field/variable. The IDE you are working with may not recognize the typing, but OFBiz can work with it:
   ```java
   def fieldName = "value"
   ```

### 13.4. Checking Fields

<table>
<thead>
<tr>
<th>Minilang</th>
<th>Groovy</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;if-empty field=&quot;fieldName&quot;&gt;&lt;/if-empty&gt;</code></td>
<td>// checks if fieldName is existent and/or empty</td>
</tr>
<tr>
<td></td>
<td><code>if (!fieldName) {}</code></td>
</tr>
<tr>
<td><code>&lt;if-empty field=&quot;fieldName.property&quot;&gt;&lt;/if-empty&gt;</code></td>
<td>// fieldName has to be existent, property doesn't need to</td>
</tr>
<tr>
<td></td>
<td>// if known, that property does exist, the ? can be left out</td>
</tr>
<tr>
<td></td>
<td><code>if (!fieldName?.property) {}</code></td>
</tr>
<tr>
<td></td>
<td>// CAUTION: every query like this in Groovy evaluates to a Boolean type</td>
</tr>
<tr>
<td></td>
<td>// everything that is empty or false will turn into false:</td>
</tr>
<tr>
<td></td>
<td>// null, [], [:], &quot;,&quot;, false -&gt; false</td>
</tr>
<tr>
<td></td>
<td><code>if (UtilValidate.isEmpty(fieldName)) {}</code></td>
</tr>
</tbody>
</table>
### 13.5. Setting Fields

<table>
<thead>
<tr>
<th>Minilang</th>
<th>Groovy</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;if&gt;</code>&lt;br&gt;<code>&lt;condition&gt;</code>&lt;br&gt;<code>&lt;or&gt;</code>&lt;br&gt;<code>&lt;if-empty field=&quot;field1&quot;/&gt;</code>&lt;br&gt;<code>&lt;if-empty field=&quot;field2&quot;/&gt;</code>&lt;br&gt;<code>&lt;/or&gt;</code>&lt;br&gt;<code>&lt;/condition&gt;</code>&lt;br&gt;<code>&lt;then&gt;</code>&lt;br&gt;<code>&lt;!-- code in if --&gt;</code>&lt;br&gt;<code>&lt;/then&gt;</code>&lt;br&gt;<code>&lt;else&gt;</code>&lt;br&gt;<code>&lt;!-- code in else --&gt;</code>&lt;br&gt;<code>&lt;/else&gt;</code>&lt;br&gt;<code>&lt;/if&gt;</code></td>
<td>`if (!field1</td>
</tr>
<tr>
<td><code>&lt;if-compare-field</code>&lt;br&gt;<code>field=&quot;product.primaryProductCategoryId&quot;</code>&lt;br&gt;<code>to-field=&quot;parameters.productCategoryId&quot;</code>&lt;br&gt;<code>operator=&quot;equals&quot;</code>&lt;br&gt;<code> &lt;!-- code --&gt;</code>&lt;br&gt;<code>&lt;/if-compare-field&gt;</code></td>
<td><code>// this will even work, if product is not existent or null</code>&lt;br&gt;<code>if (UtilValidate.areEqual(product?.primaryProductCategoryId, parameters?.productCategoryId)) {</code>&lt;br&gt;<code>// code</code>&lt;br&gt;`}</td>
</tr>
<tr>
<td><code>&lt;if-instance-of</code>&lt;br&gt;<code>field=&quot;parameters.categories&quot;</code>&lt;br&gt;<code>class=&quot;java.util.List&quot;</code>&lt;br&gt;<code>&lt;/if-instance-of&gt;</code></td>
<td><code>if (parameters.categories instanceof java.util.List) {}</code></td>
</tr>
</tbody>
</table>

```minilang
<set field="fieldName" value="value"/>
```

```groovy
// if fieldName is not initialized
String fieldName = "value"
// if fieldName is initialized
fieldName = "value"
```
### Minilang

```
<set field="otherFieldName.property" value="value"/>
<set field="otherFieldName.otherProperty" value="true" type="Boolean"/>
<set field="otherFieldName.otherProperty" from-field="parameters.property"/>
```

// if otherFieldName is not yet initialized, you have to do it first
// MiniLang does that automatically
Map otherFieldName = [:]; // empty Map
// now put the values in
otherFieldName = [
    property: "value",
    otherProperty: true
];
// or the less efficient way
otherFieldName.property = "value"
otherFieldName.otherProperty = true

// it is possible to put different values in later:
otherFieldName.property = parameters.property
```

### Groovy

```
<set field="thisFieldName" value="${groovy: []}" type="List"/>
```

// this is easier in Groovy
List thisFieldName = []

```
<property-to-field resource="CommonUiLabels" property="CommonGenericPermissionError" field="failMessage"/>
<!-- there are different cases of this, which are not distinguished in MiniLang -->
<property-to-field resource="general.properties" property="currency.uom.id.default" field="parameters.rateCurrencyUomId"/>
```

String failMessage = UtilProperties.getMessage("CommonUiLabels", "CommonGenericPermissionError", parameters.locale)
// in Groovy there can be a difference for the second case
parameters.rateCurrencyUomId = UtilProperties.getPropertyValue('general.properties', 'currency.uom.id.default')

```
<clear-field field="product.primaryProductCategoryId"/>
```

```
product.primaryProductCategoryId = null
```

### 13.6. Starting Services
### 13.7. Preparing Service Results

**Minilang**

```minilang
<set
    field="relatedCategoryContext.parentProductCategoryId" from-field="defaultTopCategoryId"/>
<call-service service-name="getRelatedCategories" in-map-name="relatedCategoryContext">
    <result-to-field result-name="categories" field="resCategories"/>
</call-service>
```

**Groovy**

```groovy
def relatedCategoryContext = [parentProductCategoryId: defaultTopCategoryId]
def serviceResult = run service: "getRelatedCategories", with: relatedCategoryContext
def resCategories = serviceResult.categories
// if it is not too confusing to read you can leave out the extra variable
run service: "getRelatedCategories", with: [parentProductCategoryId: defaultTopCategoryId]
```

```minilang
<set-service-fields service-name="productCategoryGenericPermission" map="parameters" to-map="productCategoryGenericPermissionMap"/>
<call-service service-name="productCategoryGenericPermission" in-map-name="productCategoryGenericPermissionMap">
    <results-to-map map-name="genericResult"/>
</call-service>
```

**Groovy**

```groovy
// instead of setting the service fields from parameters, it is possible to run the service with the parameters map
Map genericResult = run service: "productCategoryGenericPermission", with: parameters
```

### 13.8. Database Communication

**Minilang**

```minilang
<field-to-result field="fieldBudgetId" result-name="budgetId"/>
```

**Groovy**

```groovy
// MiniLang knows this implicitly
def result = success()
result.budgetId = fieldBudgetId
return result
```
Minilang

```xml
<make-value entity-name="FinAccountTrans" value-field="newEntity"/>
<set-nonpk-fields map="parameters" value-field="newEntity"/>
<set-pk-fields map="parameters" value-field="newEntity"/>
```

```groovy```
GenericValue newEntity = makeValue("FinAccountTrans", parameters)
GenericValue newEntity = makeValue("FinAccountTrans")
newEntity.setPKFields(parameters)
newEntity.setNonPKFields(parameters)
```

```
// this is the easy way
GenericValue newEntity = makeValue("FinAccountTrans", parameters)
// this is also possible
GenericValue newEntity = makeValue("FinAccountTrans")
newEntity.setPKFields(parameters)
newEntity.setNonPKFields(parameters)
```

```
<entity-and entity-name="BudgetStatus" list="budgetStatuses">
  <field-map field-name="budgetId" from-field="parameters.budgetId"/>
  <order-by field-name="-statusDate"/>
</entity-and>
```

```groovy```
// this can also be done in one line, but it can easily become unreadable
def budgetStatuses = from("BudgetStatus")
    .where("budgetId", parameters.budgetId)
    .orderBy("-statusDate")
    .queryList()
```

```
<entity-one entity-name="StatusValidChange" value-field="statusValidChange">
  <field-map field-name="statusId" from-field="budgetStatus.statusId"/>
  <field-map field-name="statusIdTo" from-field="parameters.statusId"/>
</entity-one>
```

```groovy```
// this is also possible
statusValidChange = from("StatusValidChange")
    .where("statusId", budgetStatus.statusId, parameters.statusIdTo, false)
```

```
<!-- entity-one can be called without child elements, too -->
<entity-one entity-name="Product" value-field="product" auto-field-map="true"/>
```

```groovy```
// if there are no child elements, this can be used
GenericValue product = from("Product")
    .where(parameters).queryOne()
```

```
// MiniLang has false set for useCache as the default value
statusValidChange = findOne("StatusValidChange", [statusId:
    budgetStatus.statusId, statusIdTo:
    parameters.statusId], false)
```
### Minilang

```xml
<find-by-primary-key entity-name="ProductCategoryMember"
  map="lookupPKMap" value-field="lookedUpValue"/>
```

### Groovy

```groovy
GenericValue lookedUpValue = findOne
( "ProductCategoryMember", lookupPKMap, false)
// this is also possible
lookedUpValue = from
( "ProductCategoryRole"
  .where(lookupPKMap)
  .queryOne())
```

```xml
<entity-condition entity-name="ProductCategoryContentAndInfo"
  list="productCategoryContentAndInfoList"
  filter-by-date="true" use-cache="true">
  <condition-list combine="and">
    <condition-expr field-name="productCategoryId" from-field="productCategoryList.productCategoryId"/>
    <condition-expr field-name="prodCatContentTypeId" value="ALTERNATIVE_URL"/>
  </condition-list>
  <order-by field-name="-fromDate"/>
</entity-condition>

<!-- entity-condition can also be used with the "or" operator -->

```groovy
// the Groovy methods use the "and" and "equals" operator as default values
List productCategoryContentAndInfoList = from("ProductCategoryContentAndInfo")
  .where("productCategoryId", productCategoryList.productCategoryId, "prodCatContentTypeId", "ALTERNATIVE_URL")
  .cache().orderBy("-fromDate")
  .filterByDate()
  .queryList()
```

```groovy
// with the use of the "or" operator you have to build your condition like this
EntityCondition condition = EntityCondition.makeCondition([
  EntityCondition.makeCondition([
    EntityCondition.makeCondition(
      "prodCatalogCategoryTypeId",
      "PCCT_VIEW_ALLW"),
    EntityCondition.makeCondition(
      "prodCatalogCategoryTypeId",
      "PCCT_PURCH_ALLW")
  ], EntityOperator.OR),
  EntityCondition.makeCondition(
    "productCategoryId",
    parameters.productCategoryId)
])
```

```groovy
List prodCatalogCategoryList = from("ProdCatalogCategory")
  .where(condition)
  .filterByDate().queryList()
```
<table>
<thead>
<tr>
<th>Minilang</th>
<th>Groovy</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;make-value entity-name=&quot;FinAccountTrans&quot; value-field=&quot;newEntity&quot;/&gt;</code></td>
<td><code>def newEntity = makeValue (&quot;FinAccountTrans&quot;, parameters)</code></td>
</tr>
<tr>
<td><code>&lt;set-nonpk-fields map=&quot;parameters&quot; value-field=&quot;newEntity&quot;/&gt;</code></td>
<td>// you can set multiple fields of a GenericValue like this</td>
</tr>
<tr>
<td>&lt;!-- In this case multiple fields of the GenericValue are set --&gt;</td>
<td>def newLimitRollup = makeValue (&quot;ProductCategoryRollup&quot;, [</td>
</tr>
<tr>
<td><code>&lt;make-value entity-name=&quot;ProductCategoryRollup&quot; value-field=&quot;newLimitRollup&quot;/&gt;</code></td>
<td><code>    productCategoryId: newEntity.productCategoryId,</code></td>
</tr>
<tr>
<td><code>&lt;set field=&quot;newLimitRollup.productCategoryId&quot; from-field=&quot;newEntity.productCategoryId&quot;/&gt;</code></td>
<td><code>    parentProductCategoryId: productCategoryRole.productCategoryId,</code></td>
</tr>
<tr>
<td><code>&lt;set field=&quot;newLimitRollup.parentProductCategoryId&quot; from-field=&quot;productCategoryRole.productCategoryId&quot;/&gt;</code></td>
<td><code>    fromDate: nowTimestamp</code></td>
</tr>
<tr>
<td><code>&lt;set field=&quot;newLimitRollup.fromDate&quot; from-field=&quot;nowTimestamp&quot;/&gt;</code></td>
<td>})</td>
</tr>
<tr>
<td><code>&lt;set field=&quot;statusValidChange.prop&quot; value=&quot;value&quot;/&gt;</code></td>
<td><code>statusValidChange.prop = &quot;value&quot;</code></td>
</tr>
<tr>
<td><code>&lt;create-value value-field=&quot;newEntity&quot;/&gt;</code></td>
<td><code>newEntity.create()</code></td>
</tr>
<tr>
<td><code>&lt;store-value value-field=&quot;newEntity&quot;/&gt;</code></td>
<td><code>newEntity.store()</code></td>
</tr>
<tr>
<td><code>&lt;store-list list=&quot;listToStore&quot;/&gt;</code></td>
<td><code>delegator.storeAll(listToStore)</code></td>
</tr>
<tr>
<td><code>&lt;clone-value value-field=&quot;productCategoryMember&quot; new-value-field=&quot;newProductCategoryMember&quot;/&gt;</code></td>
<td><code>def newProductCategoryMember = productCategoryMember.clone()</code></td>
</tr>
<tr>
<td><code>&lt;remove-value value-field=&quot;lookedUpValue&quot;/&gt;</code></td>
<td><code>lookedUpValue.remove()</code></td>
</tr>
</tbody>
</table>
### 13.9. Permissions

To also check for admin-permissions, this method has to be used:

```groovy
hasEntityPermission(permission, action, userLogin)
```

If the method is used with wildcards, it is important to **not forget the underscore**, which comes before the parameter action!
13.10. Timestamp And System Time

The first two simple-method are deprecated; the third method should have been used instead.

```
Minilang

<set field="hasCreatePermission" value="false" type="Boolean"/>
<if-has-permission permission="${primaryPermission}" action="${mainAction}"
  <set field="hasCreatePermission" value="true" type="Boolean"/>
</if-has-permission>

// this will automatically be set to false if the user doesn't have the permission
def hasCreatePermission = security .hasEntityPermission(primaryPermission,
"_${mainAction}" , parameters.userLogin)

Groovy

Timestamp nowTimestamp = UtilDateTime
.nowTimestamp()

Timestamp nowDate = UtilDateTime
.nowTimestamp()

Timestamp fooNow = UtilDateTime
.nowTimestamp()

Timestamp thruDate = productCategoryMember.thruDate
if (thruDate && thruDate.before(expireTimestamp)) {
  // code
}
```

13.11. Logging

Since all of the log methods are know to the Groovy Language, it is possible to just nearly use them as they are in MiniLang.

For further explanation, here are some examples:
### 13.12. General

**Minilang**

```xml
<call-simple-method method-name="checkCategoryRelatedPermission"/>
<check-errors/>
<iterate list="subCategories" entry="subCategory">
  <!-- code -->
</iterate>
```

**Groovy**

```groovy
// simple-methods inside of classes, as long as they are not services, will be called like normal methods
Map res =
  checkCategoryRelatedPermission("updateProductCategory", "UPDATE", null, null)
if (!ServiceUtil.isSuccess(res)) {
  return res
}
```

```groovy
for (def subCategory : subCategories) {
  // code
}
subCategories.each { subCategory ->
  // code
}
```
<iterate-map
  map="parameters.productFeatureIdByType"
  key="productFeatureTypeId"
  value="productFeatureId">
  <!-- in here something should happen with value and key -->
</iterate-map>

for (Map entry : parameters.productFeatureIdByType.entrySet()) {
  def productFeatureTypeId = entry.getKey()
  def productFeatureId = entry.getValue()
  // in here something should happen with value and key
}

if (!security.hasEntityPermission("CATALOG", "_${checkAction}", parameters.userLogin)) {
  if (!security.hasEntityPermission("CATALOG_ROLE", "_${checkAction}", parameters.userLogin)) {
    // code
  }
}

def query = from("ProductCategoryMember").where("productCategoryId", parameters.productCategoryId)
if (parameters.validDate) {
  query.filterByDate()
}
List productCategoryMembers = query.queryList()
13.13. Where to find MiniLang implementation

If you find yourself in a position, where you don’t know how to convert a certain tag from MiniLang to Groovy, you can always check the Java implementation of the MiniLang method. All of the methods have an existing Java implementation and you can find all of them in this folder: /ofbiz/trunk/framework/minilang/src/main/java/org/apache/ofbiz/minilang/method

The interesting part of this implementation is the method `exec()`, which actually runs the MiniLang tag. The tag `<remove-by-and>` for example is realized using this part of code here:

```java
@Override
public boolean exec(MethodContext methodContext) throws MiniLangException {
    @Deprecated
    String entityName = entityNameFse.expandString(methodContext.getEnvMap());
    if (entityName.isEmpty()) {
        throw new MiniLangRuntimeException("Entity name not found.", this);
    }
    try {
        Delegator delegator = getDelegator(methodContext);
        delegator.removeByAnd(entityName, mapFma.get(methodContext.getEnvMap()));
    } catch (GenericEntityException e) {
        String errMsg = "Exception thrown while removing entities: " + e.getMessage();
        Debug.logWarning(e, errMsg, module);
        simpleMethod.addErrorMessage(methodContext, errMsg);
        return false;
    }
    return true;
}
```

In this you can find one important part of code, which is:

```java
deleagator.removeByAnd(entityName, mapFma.get(methodContext.getEnvMap()));
```

This tells you, that, if you’re trying to convert the tag `<remove-by-and>`, you can use `delegator.removeByAnd()` in Groovy.