**VMS-OOB Migration Codify 1.2-SNAPSHOT**

**Contents:**

1. Migration steps
2. New files Added:
   1. applicationContext-csv.xml
   2. ValidationMessagesMetaInfo.properties
3. Files modified:
   1. applicationContext.xml
   2. pom.xml
   3. AttributeChoiceLookupProvider.java
   4. AttributeChoiceLookup.java
4. Issues /Resolutions during migration phase:
   1. navigation.xml
5. Version Change Verification Results
   1. Saved Filter
   2. Copy/Replicate Feature
   3. Workqueue Requirements
   4. Dialog window on User actions
   5. Dirty detection and confirmation
   6. Change Log functionality
   7. Web based permissions
   8. Validation Messages Grouping
   9. Effective Date Range Support
   10. Secret Masking

**Migration steps:**

1. **Update the pom.xml with new codify version from 1.1 Snapshot to 1.2 Snapshot as given below:**

<parent>

<groupId>com.thirdpillar.codify</groupId>

<artifactId>codify-parent-pom</artifactId>

<version>1.2-SNAPSHOT</version>

</parent>

<properties>

<filterdir>${basedir}/src/main/filters</filterdir>

<filter>dev</filter>

<codify.version>1.2-SNAPSHOT</codify.version>

. . . . . . . . . .

</properties>

1. **Add applicationContext-csv.xml and ValidationMessagesMetaInfo.properties file under ‘………\vrms\service\src\main\resources’ .**
2. **Update applicationContext.xml with code given in the next sections under heading ‘Modification in applicationContext.xml’ in the same document.**
3. **Run command : mvn -U clean install.**
4. Don’t do a full db refresh and only below script need to be executed without seeding full data:

Saved Filter-/trunk/database/src/main/resources/update/update-5034.sql

Change Log functionality-/trunk/database/src/main/resources/update/update-5984.sql

Web based permissions-/trunk/database/src/main/resources/update/update-4894.sql

TUSF–1104-/trunk/database/src/main/resources/update/update-5140.sql

**Modification in applicationContext.xml :**

**Following lines of code removed from applicationContext.xml:**

1. **PermissionHandler:**

<!-- <bean id="PermissionHandler" class="com.thirdpillar.foundation.rules.PermissionHandler" init-method="initialize">

<property name="files" value="/permissions/sample.xml"/>

</bean> -->

1. **uiPermissionHandler:**

<!-- <bean id="uiPermissionHandler" class="com.thirdpillar.foundation.rules.UiPermissionHandler" init-method="initialize">

<property name="files">

<list>

<value>/permissions/user-uiPermissions.xml</value>

<value>/permissions/role-uiPermissions.xml</value>

<value>/permissions/team-uiPermissions.xml</value>

<value>/permissions/attribute-uiPermissions.xml</value>

<value>/permissions/vendor-uiPermissions.xml</value>

<value>/permissions/engagement-uiPermissions.xml</value>

<value>/permissions/incident-uiPermissions.xml</value>

<value>/permissions/surveyGroup-uiPermissions.xml</value>

<value>/permissions/contract-uiPermissions.xml</value>

<value>/permissions/dashboard-uiPermissions.xml</value>

<value>/permissions/calendar-uiPermissions.xml</value>

<value>/permissions/report-uiPermissions.xml</value>

<value>/permissions/documentLibrary-uiPermissions.xml</value>

<value>/permissions/tasks-uiPermissions.xml</value>

<value>/permissions/configuration-uiPermissions.xml</value>

<value>/permissions/contractReviewDetail-uiPermissions.xml</value>

</list>

</property>

</bean> -->

1. **permissionFactory:**

<!-- <bean id="permissionFactory" class="com.thirdpillar.foundation.permissions.PermissionFactory">

<property name="files">

<list>

<value>permissions/BusinessPermissions.xml</value>

<value>permissions/RoleCategoryVendor.xml</value>

</list>

</property>

</bean> -->

1. **permissionManager**:

<!-- <bean id="permissionManager" class="com.thirdpillar.foundation.permissions.PermissionManager" lazy-init="false">

<property name="businessPermission" value="BusinessRules" />

</bean> -->

1. **auditProcessingJmsContainer :**

<!-- <bean id="auditProcessingJmsContainer" parent="abstractAuditProcessingJmsContainer">

<property name="autoStartup" value="false"/>

</bean> -->

**Following lines of code added into applicationContext.xml:**

1. **webPermissionManager and webXmlPermissionFactory :**

<bean id=*"webPermissionManager"* class=*"com.thirdpillar.foundation.permissions.WebPermissionManager"* lazy-init=*"false"*>

<property name=*"businessPermission"* value=*"BusinessRules"* />

<property name=*"permissionFactories"*>

<list>

<!-- <ref bean="dbPermissionFactory"/> -->

<ref bean=*"webXmlPermissionFactory"*/>

</list>

</property>

</bean>

<bean id=*"webXmlPermissionFactory"* class=*"com.thirdpillar.foundation.permissions.WebXmlPermissionFactory"*>

<property name=*"files"*>

<list>

<value>/permissions/user-uiPermissions.xml</value>

<value>/permissions/role-uiPermissions.xml</value>

<value>/permissions/team-uiPermissions.xml</value>

<value>/permissions/attribute-uiPermissions.xml</value>

<value>/permissions/vendor-uiPermissions.xml</value>

<value>/permissions/engagement-uiPermissions.xml</value>

<value>/permissions/incident-uiPermissions.xml</value>

<value>/permissions/surveyGroup-uiPermissions.xml</value>

<value>/permissions/contract-uiPermissions.xml</value>

<value>/permissions/dashboard-uiPermissions.xml</value>

<value>/permissions/calendar-uiPermissions.xml</value>

<value>/permissions/report-uiPermissions.xml</value>

<value>/permissions/documentLibrary-uiPermissions.xml</value>

<value>/permissions/tasks-uiPermissions.xml</value>

<value>/permissions/configuration-uiPermissions.xml</value>

<value>/permissions/contractReviewDetail-uiPermissions.xml</value>

</list>

</property>

</bean>

1. **entityPermissionManager and entityXmlPermissionFactory :**

<bean id=*"entityPermissionManager"* class=*"com.thirdpillar.foundation.permissions.EntityPermissionManager"* lazy-init=*"false"*>

<property name=*"businessPermission"* value=*"BusinessRules"* />

<property name=*"permissionFactories"*>

<list>

<!-- <ref bean="dbPermissionFactory"/> -->

<ref bean=*"entityXmlPermissionFactory"*/>

</list>

</property>

</bean>

<bean id=*"entityXmlPermissionFactory"* class=*"com.thirdpillar.foundation.permissions.EntityXMLPermissionFactory"*>

<property name=*"files"*>

<list>

<value>permissions/BusinessPermissions.xml</value>

</list>

</property>

</bean>

1. **dbPermissionFactory and dbPermissionSetFinder :**

<!-- <bean id="dbPermissionFactory" class="com.thirdpillar.foundation.permissions.DBPermissionFactory">

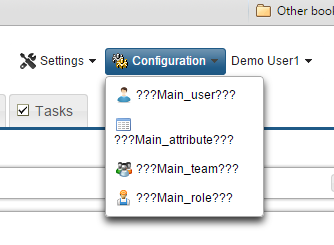
<property name="permissionSetFinder" ref="dbPermissionSetFinder" />

</bean> -->

<!-- <bean id="dbPermissionSetFinder" class="com.thirdpillar.codify.loanpath.service.PermissionSetFinderService"/> -->

**Issues /Resolutions during migration phase:**

After migration from Codify 1.1-SNAPSHOT to Codify 1.2-SNAPSHOT below know issue under Configuration list as given below:



**For the resolution, add below code in navigation.xml :**

<tabs>

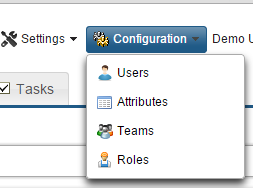
<tab id=*"User"* label=*"Users"* />

<tab id=*"attribute"* label=*"Attributes"* />

<tab id=*"Team"* label=*"Teams"* />

<tab id=*"Role"* label=*"Roles"* />

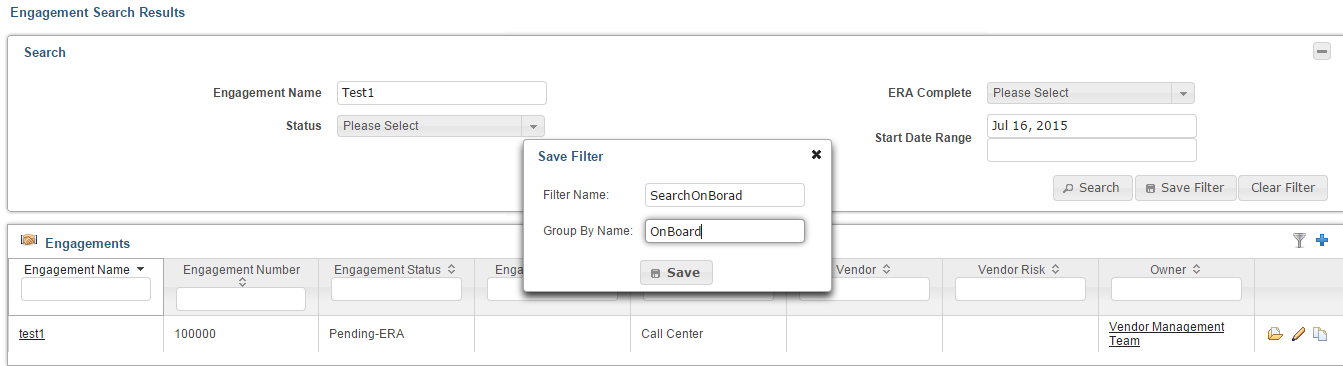
</tabs>



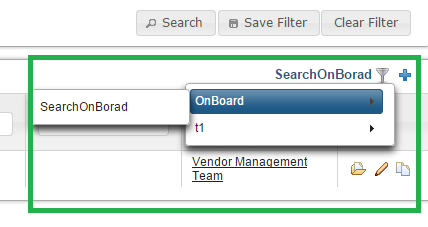
**Version Change Verification Results:**

**Saved Filter:**

1. Run the update script required for saved search filters to work “/trunk/database/src/main/resources/update/update-5034.sql”
2. Click on Search panel. ‘Save Filter’ and ‘Clear Filter’ buttons getting displayed.
3. Click on ‘Save Filter’ button, then a pop dialog box will open as shown in below screenshot.
4. Fill ‘Filter Name’ (ex: ‘SearchOnBorad’) and ‘Group By Name (ex: ‘Onboard’) and click on ‘Save’ button.



1. Click on button  and filter ‘OnBoard’ created previously start getting displayed in filter list as shown below. Click on OnBoard filter then related all Engagement listed



# Copy/Replicate Feature:

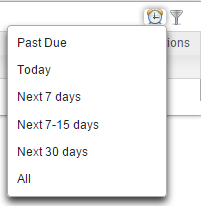
Copy/Replicate feature provides us the ability to create a copy of any entity. Like View/Edit we’ll have a Copy Icon automatically available on List page. System will create “Fnd-Copy” Profile by default which is an extension of “Full ” and lists all fields except id/auto generated fields (like refNumber) and metainfo.

**Workqueue Requirements:**

The workqueue requirements included following

1. Highlight based on days in a certain Status. ex. Pending > 5 days
2. Task list filters by Due date
3. View various transactions in different statuses - Grouped search filters
4. Support use of dynamic criteria like Now, ctxUser
5. Highlighting in Tasks

Task list filters by Due date as given below :

****

**Dialog window on User actions:**

This feature provides capability to user to be able to specify a dialog on any command button. For eg, a specific custom dialog on remove of an entity, a custom dialog which quickly takes in user inputs for specific fields before generating a document or saving an entity etc

Eg:

<commands>

<command id="attribute\_attributeChoices\_deleteAction" name="Delete Attribute" confirmation-key="attribute\_attributeChoices\_deleteAction.confirm">

<dialog-canvas label="Delete Attributes">

<section id="AtrribDialog">

<field name="attributeChoice.name" />

<field name="attributeChoice.key" />

</section>

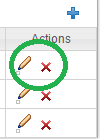
</dialog-canvas>

</command>

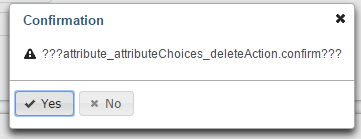
</commands>

</data-table>

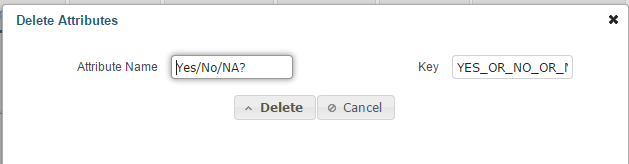
As soon as, User clicks on delete button from a list data (eg. **Attribute Choices list)**



After click on delete(X) button, confirm Dialog window opened as shown below:

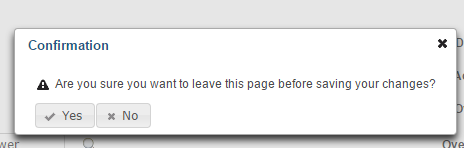


If user click on Yes then another confirm Dialog window opened as given below :

****

# 

# Dirty detection and confirmation:

The requirement for Dirty Detect and confirm is for the user to be notified by codify when he tries to move out of current page/tab while having some unsaved changes. There can be many scenarios here where the user should be prompt and should not be prompted as shown below:****

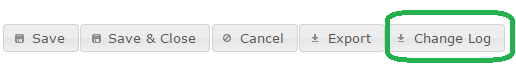
# Change Log functionality:

The requirement is for user to see all the changes made by her as soon as an entity is created. The functionality can be enabled by adding following tag in metamodel

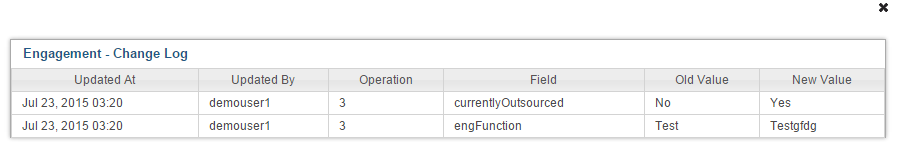
1. <entity name=*"Engagement"* stereotype=*"business-entity"* label=*"Engagement"*>

<change-log compare-profile=*"Full"* />

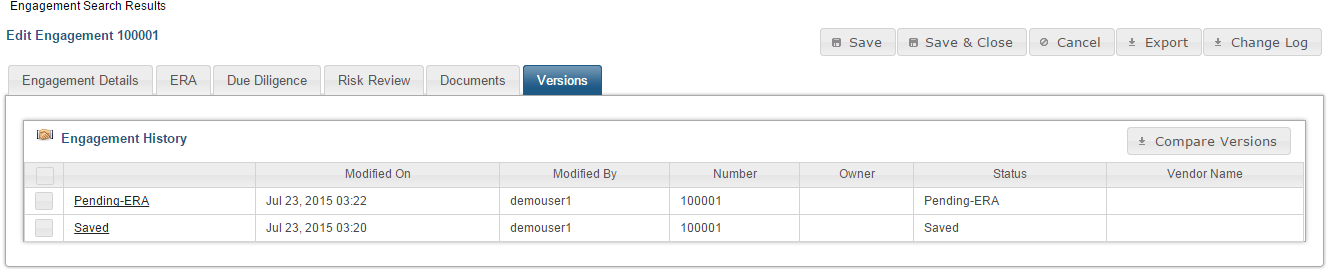
1. Run updatescript “: /trunk/database/src/main/resources/update/update-5984.sql”



1. After Click on  button below dialog box opened:

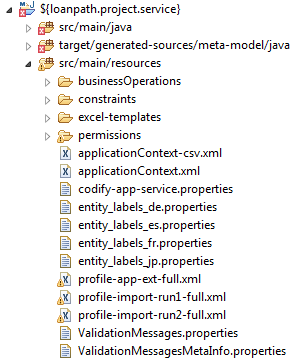


1. User should create their own profile and feed it to the system instead of using Full as using Full is extremely memory intensive and costly. On enabling change logs for an entity, user will be able to see Change Log button on main crud page and Compare Version button on Versions page. Versions whether tagged or not, both will have Compare Versions functionality automatically enabled.



**Validation Messages Grouping:**

* New feature added in Codify 1.2, we can group the validation messages or order them.
* For this feature to work, every project will have to include in file V***alidationMessagesMetaInfo.properties*** in service module.
* Contents of file will contain key for message which should be picked up from ValidationMessages.properties and value would be a json structure for representing priority.

****

# 

# Effective Date Range Support:

**Changes in 'AttributeChoiceLookupProvider.java' given below :**

* lookupList() method -modified
* List<AttributeChoice> applyEffectiveDateRange(List<AttributeChoice> values)-new method added
* List<AttributeChoice> applyGroupMvel(List<AttributeChoice> values)-new method added

**Changes in file 'AttributeChoiceLookup.java' given below:**

* New method ‘boolean effective() default true;’ added

# 

# Secret Masking:

**Reg Ex Examples**

Fixed length. Format like SSN

/\d{5}(?=.{4})/xxx-xx-

**Variable length. Show last 4, mask the rest**

/\d(?=\d{4})/x

**Variable length. Mask last 4, show the rest**

/\d{4}$/xxxx

For masking the replacer is responsible, hence if the replacer is having '\*' it will be used, if 'x' it will be used.

Dev is in control to use '\*' or 'x' or any other char