LECTRA FASHION PLM
PRODUCT DEVELOPER

Screens Configuration Use Cases

Date of last update: December 2016

Contents

Introduction **3**

1. Product **3**

1.1 New Custom Field 3
1.2 New custom field with associated length unit 8
1.3 New attribute (variation axis) 10
1.4 New classification attribute 13

2. SKU **14**

2.1 SKU form and SKU Grid in the product 14
2.2 SKU global Search 19

3. SPEC PACKAGE **24**

3.1 Instruction Custom Spec package - New Configuration 24

4. BOM BOL COST **26**

5. CALENDAR **29**

5.1 Steps 29
5.2 Planning 41

6. SUBSETS **50**

6.1 Subset instance 50
6.1 Subsets explorer 54

7. COLLECTION PLAN **57**

7.1 Detail Collection Plan screen 57
7.2 List Collection Plan screen 62

Modifications made to the document since its last publication are highlighted in blue.

INTRODUCTION

The objective of this document is to show how to implement some screens configuration use cases to help implementation team to solve some frequent customer's requests. This document does not explain how to configure the system or how to use PLM Manager. Please see appropriate Documents about V4R2 Configuration and PLM Manager for more details.

1. PRODUCT

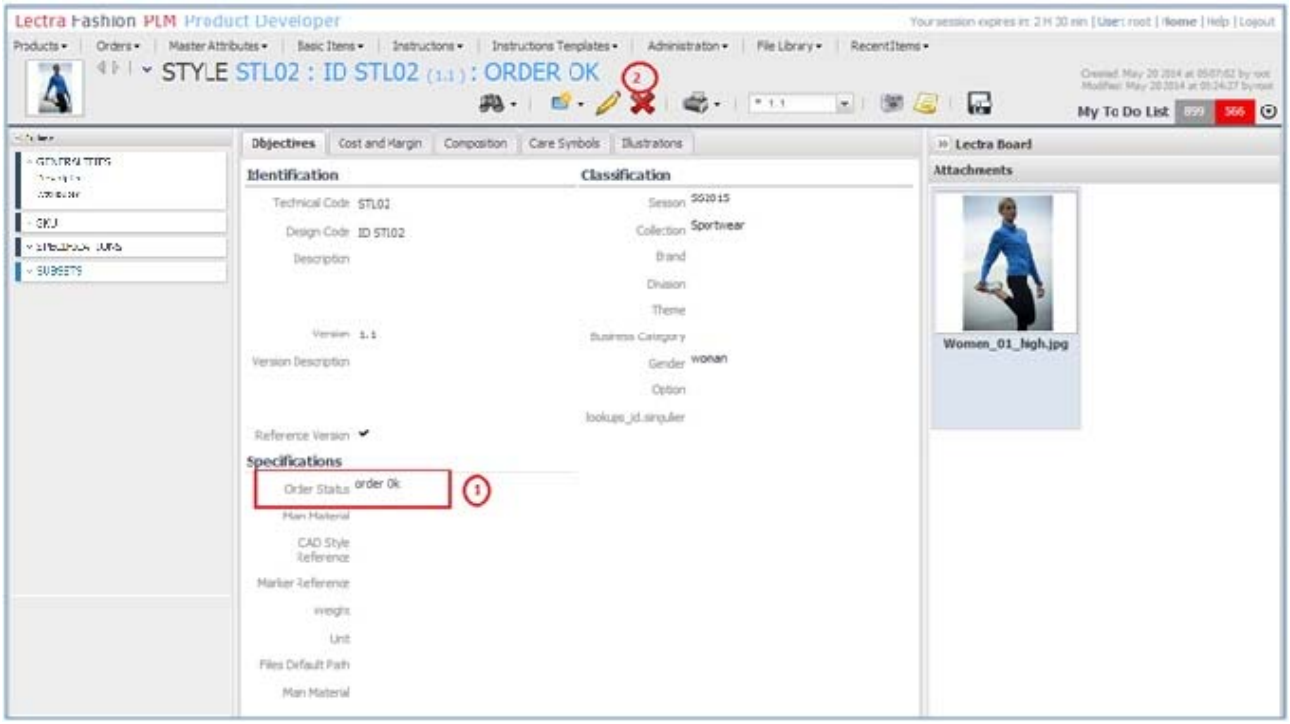
1.1 New Custom Field

1.1.1 Case

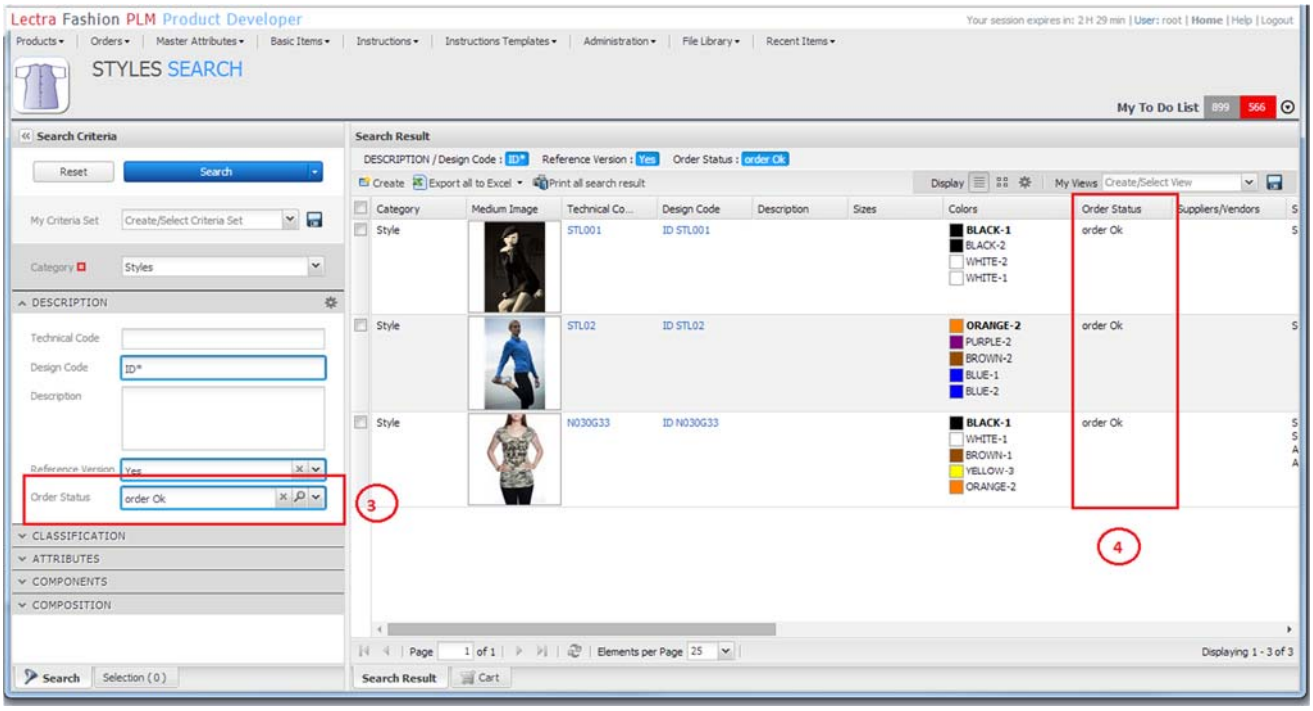
Add a new field for Style Category named "Order Status". This field takes its value from a pick list.

Use this field as search criteria.

Then show this field in Style Header.



Order Status in Style Description and Header



Order Status as Search Criteria in Style Explorer

1.1.2 Configuration

PLM Manager Steps

1. Menu: Data->Values List.
2. Add pick list named "OrderStatus" if it doesn't exist
3. Menu: Data Configuration->Extend Data Model.
4. Branch Style: Add Custom role named "orderStatus".
5. Define "orderStatus" custom role as a search criteria in .xls file to import in PLM Manager.

Screens

(1) Declare your custom role for Style Description Objectives:

a. On your PLM server, open **namespaces/custom/Lectra.PDM.ProductGO.xml** under LectraPLMParam.

b. Duplicate the following code from namespaces/Lectra.PDM.ProductGO.xml to namespaces/custom/Lectra.PDM.ProductGO.xml.

c. Add the highligted line.

```
<panel name="description_Style" layout="column" autoScroll="true" padding="4">
  <panel columnWidth="0.5" border="false">
     <panel name="identification" titleI18n="Screen.identification" inherits="looklikefieldset">
       <textfield id="codeAlpha1" name="codeAlpha1" fieldLabelI18n="technicalCode"
exclusiveField="codeAlpha2" />
       <textfield id="codeAlpha2" name="codeAlpha2" fieldLabelI18n="studyCode"
exclusiveField="codeAlpha1" />
       <textarea name="description" fieldLabelI18n="description" />
```

```
        <textfield name="version.number" fieldLabelI18n="version" readOnly="true"
condition="${#newStatut neq true}" conditionCtrl="com.lectra.lpf.uimodel.FeatureFilter" />
        <textarea name="version.description" fieldLabelI18n="label.version.description"
condition="${#newStatut neq true}" conditionCtrl="com.lectra.lpf.uimodel.FeatureFilter" />
        <xcheckbox xtype="refversionfield" name="version.refVersion" fieldLabelI18n="RefVersion"
condition="${#newStatut neq true}" conditionCtrl="com.lectra.lpf.uimodel.FeatureFilter" />
    </panel>
    <panel name="detail" titleI18n="Screen.specification" inherits="looklikefieldset">
        <nodecombo name="orderStatus" fieldLabelI18n="orderStatus" hiddenName="orderStatus"
editable="true" />
        <nodecombo fieldLabelI18n="Style.mainFabric" hiddenName="mainFabric" />
        <textfield  name="cad_style_reference" fieldLabelI18n="cad_style_reference"
conditionCtrl="com.lectra.lpf.uimodel.FeatureFilter" anchor="100%"/>
        <textfield  name="marker_consumption" fieldLabelI18n="marker_consumption"
conditionCtrl="com.lectra.lpf.uimodel.FeatureFilter" />
        <numberfield fieldLabelI18n="WeightDTO.value" name="weight.value" />
        <nodecombo fieldLabelI18n="unit" hiddenName="weight.unit" />
        <nodecombo xtype="filepathnodecombo" forceSelection="false"
propertyNameText="${#i18n['path']}" name="fileAttachementDefaultPath"
fieldLabelI18n="ProductGODTO.fileAttachementDefaultPath"
conditionCtrl="com.lectra.lpf.uimodel.FeatureFilter" >
            <listener on="filePathUpdated" fn="Lectra.PDM.ProductGO.updateFileDefaultPath" />
        </nodecombo>
    </panel>
  </panel>

  <panel name="collection" titleI18n="Screen.collection" columnWidth="0.5"
inherits="looklikefieldset">
    <nodecombo fieldLabelI18n="${#axeRoleName}.singulier" hiddenName="${#classifRoleName}"
              multiSelect="true" propertyName="${#propertyName}" genProductClassif="prodgo" />

     <!-- Private field(s) : do not edit -->
     <customSerializer
type="com.lectra.pdm.lpf.ext.serializers.ProductClassifGeneratorCustomSerializer" />
     <!-- End of private field(s)          -->
  </panel>

        <!-- Private field(s) : do not edit -->
        <customSerializer type="com.lectra.lpf.ext.serializer.ReadOnlyCustomSerializer"
condition="${#PAGE_TYPE.isDisplaying()}" />
        <!-- End of private field(s) -->

</panel>
```

2 Declare your custom role for Style Header:

a. On your PLM server, open **/template/default/newHeaderAndActions.html** under LectraPLMParam.

b. Add the highlighted lines in "titrebleu" description part.

```
<span class="titrebleu">
        #if($BLOCK)
                #if($BLOCK.getField("code1") && !
$TOOLS.isNull($SCREEN_HELPER.getField($SCREEN_DTO, $BLOCK.getField("code1"))))
                $DTO_HELPER.link($SCREEN_HELPER.getField($SCREEN_DTO, $BLOCK.getField("code1")))
                #end
                #if($BLOCK.getField("code2")&& ! $TOOLS.isNull($SCREEN_HELPER.getField($SCREEN_DTO,
$BLOCK.getField("code2")).getValue()  )  )
                 : $SCREEN_HELPER.getField($SCREEN_DTO, $BLOCK.getField("code2")).getValue()
                #end
```

```
                #if($TOOLS.isNotNull($SCREEN_HELPER.getField($SCREEN_DTO
,$BLOCK.getField("version"))) )
                <span style="font-size: 60%">
                #set($VERSION = $SCREEN_HELPER.getField($SCREEN_DTO ,$BLOCK.getField("version")))
                #if($VERSION.getValue())
                        ( $VERSION.getValue() )
                #end
                </span>
        #end
        #if($BLOCK.getField("order")&& ! $TOOLS.isNull($SCREEN_HELPER.getField($SCREEN_DTO,
$BLOCK.getField("order")).getValue()  )  )
          : $SCREEN_HELPER.getField($SCREEN_DTO, $BLOCK.getField("order")).getValue()
        #end
#else
        $I18N.getString($part2)
#end
</span>
```

3 Declare your custom role as search criteria for Style Category:

a. On your PLM server, open **namespaces/custom/Lectra.PDM.Search.Product.xml** under LectraPLMParam.

b. Copy the following code in **namespaces/custom/Lectra.PDM.Search.Product.xml** if it doesn't exist yet.

c. Add the highligted line.

```
<!-- ********************************************** -->
<!--                Description block definition    -->
<!-- ********************************************** -->
<panel name="description_Style" titleI18n="ProductGO.tree.generalObjectives" collapsible="true">
        <textfield name="codeAlpha1.values" fieldLabelI18n="codeAlpha1" chkVisible="true" />
        <textfield name="codeAlpha2.values" fieldLabelI18n="codeAlpha2" chkVisible="true" />
        <textarea name="description.values" fieldLabelI18n="description" chkVisible="true" />
        <twincombo name="RefVersion" fieldLabelI18n="RefVersion" hiddenName="refVersion.values"
        ignoreFormClear="false" inherits="http://lectra.com/pdm/common#yesno"
        conditionCtrl="com.lectra.lpf.uimodel.FeatureFilter" />
        <nodecombo fieldLabelI18n="orderStatus" propertyFamily="process"
        propertyName="OrderStatus" hiddenName="orderStatus.values" chkVisible="false" />
</panel>
```

4 Display the Column result for the Custom Role in Style Explorer:

a. On your PLM server, open **namespaces/custom/Lectra.PDM.Search.Product.xml** under LectraPLMParam.

b. Copy the following code in **namespaces/custom/Lectra.PDM.Search.Product.xml** if it doesn't exist yet.

c. Add the highligted line.

```
<explorereditablegrid name="resultGrid_Style" inherits="resultGrid">
<!--  XXX : categoryName (Style, Fabric, etc ....) -->
<!-- Columns description -->
<!-- securityCells : define the tagNames tagName1, tagName1 etc -->
<!-- tagName : column Name wisht can be controlled before render explorer -->
      <colModel inherits="resultGrid/colModel"
      securityColumns="colorTag,supplierTag,descTag"
      securityBranch="Screen" securityType="${#categoryName}" colorTag="colors,sizes"
      supplierTag="suppliers,seasonsClassif" descTag="description,codeAlpha1" >
```

```
            <column xsi:type="nodecolumn" headerI18n="orderStatus"
                    dataIndex="orderStatus"editable="true" propertyName="OrderStatus"
                    propertyFamily="process"/>
        </colModel>
        <store xsi:type="lpfdirectstore" inherits="resultGrid/store" >
               <dataField name="orderStatus" />
        </store>
</explorereditablegrid>
```

1.2 New custom field with associated length unit

1.2.1 Case

Add a new field for Style Category named "**CustomDap**". A length unit must be associated to this field.

Use this field as a research criteria associated to a second criteria of "**nodeCombo**" type that will allow search parameters to be added to the first field.

This field will then appear in the Style header as a value to be entered and a list to select the unit.

1.2.2 Configuration

PLM Manager Steps

1. Menu: Data Configuration->Extend Data Model.
2. Branch Style: Add Custom field named "CustomDap".
3. Enter the type of field to Double.
4. Select the process unit family and the unit name at "Unit Length".
5. Save and apply.
6. Define the "CustomDap" custom field as well as the length unit property as search criteria in the .xls file to port into the PLM Manager.

Screens

1 Declare your custom role for Style Description Objectives:

a. On your PLM server, open **namespaces/custom/Lectra.PDM.ProductGO.xml** under LectraPLMParam.

b. Duplicate the following code from **namespaces/Lectra.PDM.ProductGO.xml** to **namespaces/custom/Lectra.PDM.ProductGO.xml**.

c. Add the highligted line.

```
<panel name="description_Style" layout="column" autoScroll="true" padding="4"><panel
columnWidth="0.5" border="false">
<panel name="identification" titleI18n="Screen.identification"
inherits="looklikefieldset">
<numberfield fieldLabelI18n="customDap" name="customDap" />
<nodecombo fieldLabelI18n="customDap_CustomFieldUnit"
hiddenName="customDap_CustomFieldUnit"
propertyName="UnitLength" multiSelect="false" />
```

2 Declare your custom role as search criteria for Style Category:

d. On your PLM server, open **namespaces/custom/Lectra.PDM.Search.Product.xml** under LectraPLMParam.

e. Copy the following code in **namespaces/custom/Lectra.PDM.Search.Product.xml** if it doesn't exist yet.

f. Add the highligted line.

```
<panel name="description_Style" titleI18n="ProductGO.tree.generalObjectives"
collapsible="true">
<textfield name="codeAlpha1.values" fieldLabelI18n="codeAlpha1" chkVisible="true" />
<textfield name="codeAlpha2.values" fieldLabelI18n="codeAlpha2" chkVisible="true" />
<textarea name="description.values" fieldLabelI18n="description" chkVisible="true" />
<twincombo name="RefVersion" fieldLabelI18n="RefVersion" hiddenName="refVersion.values"
ignoreFormClear="false" inherits="http://lectra.com/pdm/common#yesno"/>
      <numberfield fieldLabelI18n="customDap"
name="customFieldWithUnit|||customDap|||double" />
      <nodecombo fieldLabelI18n="customDap_CustomFieldUnit"
                  hiddenName="customFieldWithUnit.arg_unitId"
                  propertyName="UnitLength"
                  multiSelect="false" />
</panel>
```

Note:

The "`customFieldWithUnit|||customDap|||double`" character string format must be followed. The `customFieldWithUnit` value is mandatory, as well as the field name (`customDap`) and its type (`double`), seperated by the "`|||`".

The types can be: `double`, `float`, `long`, `int`.

3 To display the two columns for a custom field in the styles explorer, add highlighted lines:

a. On your PLM server, open **namespaces/custom/Lectra.PDM.Search.Product.xml** under LectraPLMParam.

b. Copy the following code in **namespaces/custom/Lectra.PDM.Search.Product.xml** if it doesn't exist yet.

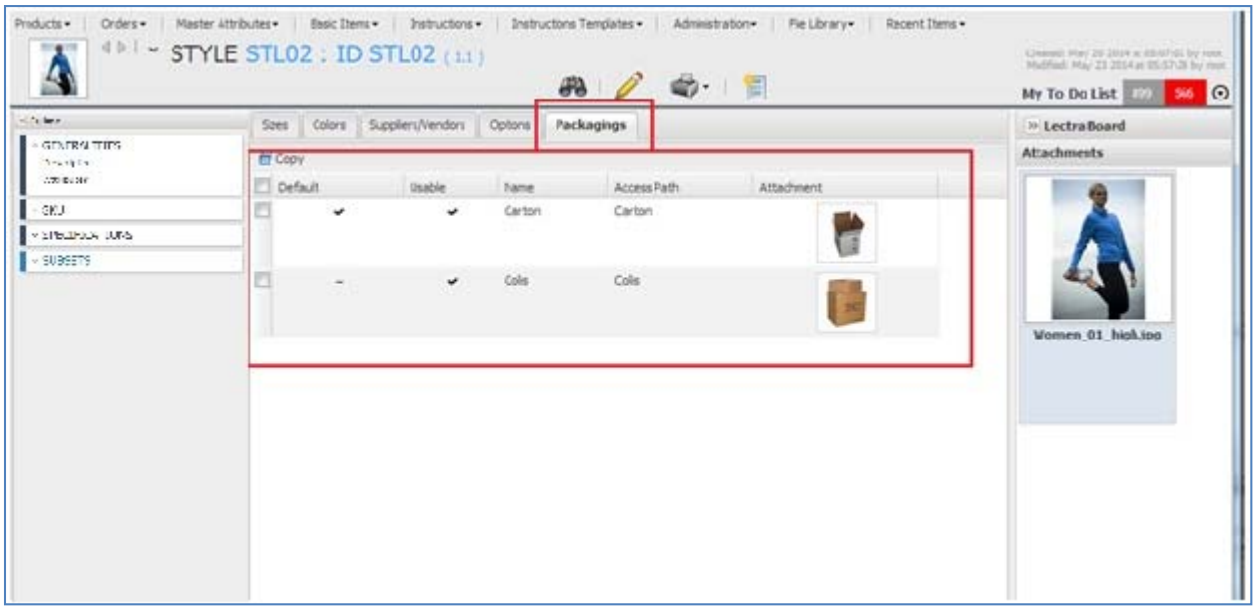
c. Add the highligted lines.

```
<explorereditablegrid name="resultGrid" managerRef="explorerStyleGrid"
explorerSubType="Product" explorerType="${#categoryName}"  lockColumn="true"
forceValidation="true" multiCell="true" stateId="ExplorerProduct_grid" stateful="true"
tempCreationId="codeNum2" singleSelectExpr="${#singleSelect}" quickCreateMode="false" >
<!-- Columns description -->
<colModel>
<column xsi:type="numbercolumn" headerI18n="customDap" dataIndex="customDap"
editable="true"/>
<column xsi:type="nodecolumn" headerI18n="customDap_CustomFieldUnit"
dataIndex="customDap_CustomFieldUnit" propertyFamily="process"
propertyName="UnitLength" editable="true"/>
...
<!-- Data description -->
<!-- Advice : Do not edit to keep all data definition -->
<store xsi:type="lpfdirectstore" remoteSort="true" forceUpdateRecordsOnFail="true">
<dataField name="customDap" />
<dataField name="customDap_CustomFieldUnit" />
...
```

1.3 New attribute (variation axis)

1.3.1 Case

Add a new attribute named "Packaging" for Style Category.



Packaging tab added in Style Attributes

1.3.2 Configuration

PLM Manager Steps

1. Menu: Classification/Define lookup values.
2. Add a new entry named "Packaging".
3. Menu: Data Configuration->Extend Data Model.
4. Branch Style->attributes: add the new attribute Packaging, Save and apply the configuration.

Screens

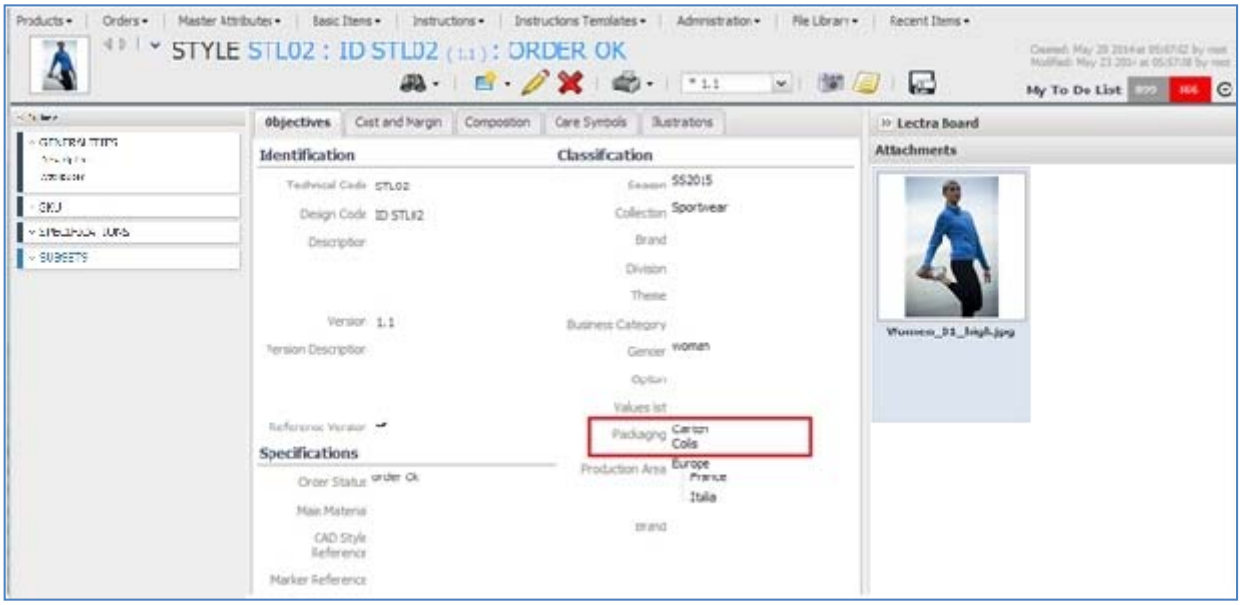
To add show the new attribute tab on Attributes explorer:

a. On your PLM server, open **screensCusto.xml** under LectraPLMParam.

b. Duplicate the following code from **screens.xml** to **screensCusto.xml**.

c. Add the highligted line.

```
<!-- Attributes screen definition of Style category -->
<screen name="Axis.Style" extends="Axis">
<tabs name="axisTabs" placement="top" prefKey="ProductGO.axis.show"
       tabManager="com.lectra.pdm.webapp.tabs.AxisTabManager"
       tabSelectionStrategy="com.lectra.pdm.webtool.config.tab.ProductGOTabStrategy">
       <tab name="sizesTab" i18nKey="ProductGO.description.sizeValues"
              icon="cmtype_icons/FinalSize_16.gif">
       <table name="sizes"
              collapsible="false"
              defaultColumn="default"
              customizable="true"
              sortable="true"
              thumbView="false"
              displayed="true"
              displayActionsLabels="true"
              enableColId="true"
              forceFit="true"
              border="false"
              resizable="false"
              resizerCt="tabPanel.Axis.Style.axisTabs">
       <column name="linePosition" i18n="linePosition" displayed="false" editable="false"
              sortableBy="linePosition"/>
       <column name="default" i18n="default" editable="true" sortable="false" minwidth="50"/>
       <column name="usable" i18n="usable" editable="true" minwidth="50"/>
       <column name="base" i18n="base" editable="true" minwidth="40"/>
<!--    <column name="envItem" i18n="FinalSize.envItem" editable="false"/> -->
       <column name="axifyable.name" i18n="name" editable="false" autoComplete="true"
              canCreate="false" pick="false" editableIf="local"/>
       <column name="envItem.qualifier" i18n="FinalSize.qualifier" editable="false"/>
       <column name="envItem.size.size.name" i18n="size" editable="false"/>
       <column name="envItem.option1.size.name" i18n="finalSizes.option1.size.name"
              editable="false"/>
       <column name="envItem.option2.size.name" i18n="finalSizes.option2.size.name"
              editable="false"/>
       <column name="envItem.sizeRun.name" i18n="sizeRun" editable="false"/>
       <column name="fileLinkSmall" i18n="fileLinkSmall" editable="true"/>
       <column name="sizeAlias" i18n="label.alias" editable="true"/>
       </table>
       </tab>
       <tab name="packagingsTab" i18nKey="packagings" directChild="true"
              associatedDtoName="packagingsClassif">
       <table name="packagings"
              collapsible="false"
              defaultColumn="default"
              customizable="true"
              sortable="true"
              displayed="true"
              forceFit="false"
              border="false">
       <column name="linePosition" i18n="linePosition" displayed="false" editable="false"
              sortableBy="linePosition"/>
       <column name="default" i18n="default" editable="true" sortable="false" minwidth="50"/>
       <column name="usable" i18n="usable" editable="true" minwidth="50"/>
       <column name="axifyable.name" i18n="name" editable="false" autoComplete="true"
              canCreate="false" pick="false" editableIf="local"/>
       <column name="axifyable.path" i18n="path" editable="false" autoComplete="true"
              canCreate="false" pick="false" editableIf="local"/>
       <column name="fileLinkSmall" i18n="fileLinkSmall" editable="true"/>
       </table>
       </tab>
</tabs>
</screen>
```

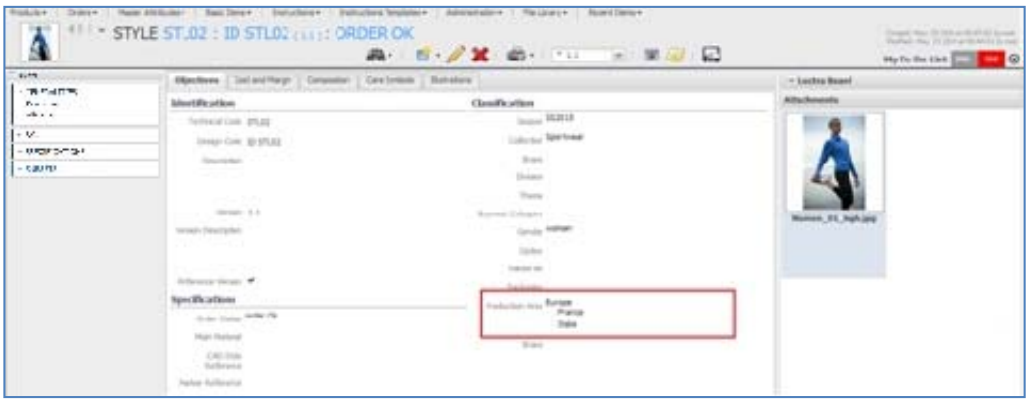


New attribute will appear also in Style Description

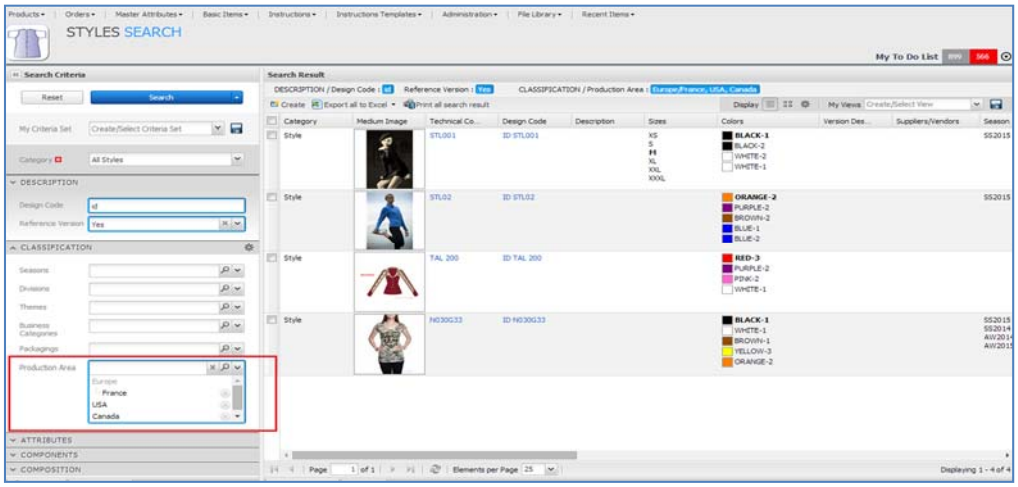
1.4 New classification attribute

1.4.1 Case

Add a new classification attribute named "ManufacturingCountry" to be able to classify Styles. The attribute is displayed in Description bloc of Styles, and will be displayed in Style search criteria.



Classification attribute added in Style Description



Classification attribute added as search criteria in Style Explorer

1.4.2 Configuration

PLM Manager Steps

1. Menu: Classification/Define Lookup Values.
2. Add a new entry named "ManufacturingCountry".
3. Menu: Classification/Affect Lookup Values.
4. Set the new attribute as classification attribute for Style Category.

Screens

Nothing to do: the new attribute will appear in both Style Description and Style Explorer.

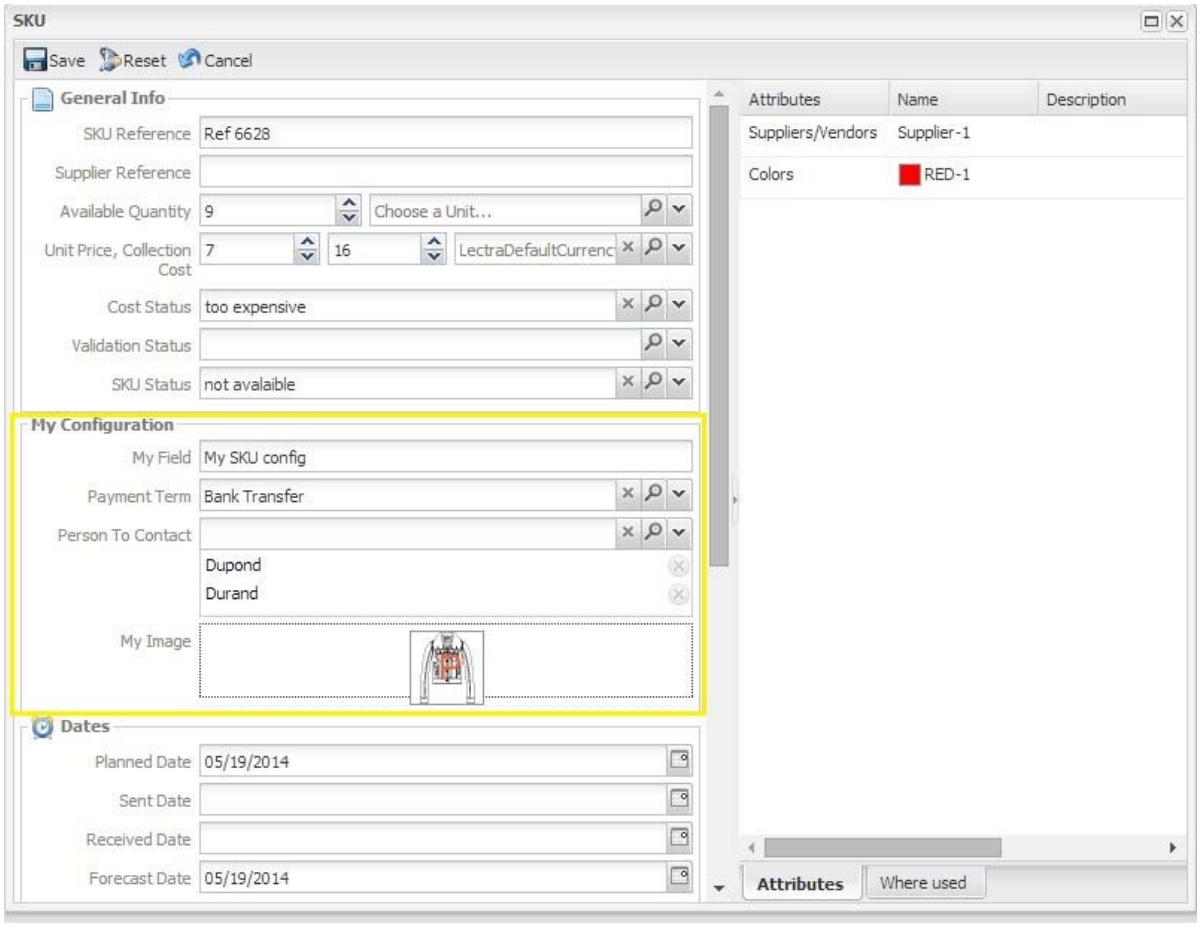
2. SKU

2.1 SKU form and SKU Grid in the product

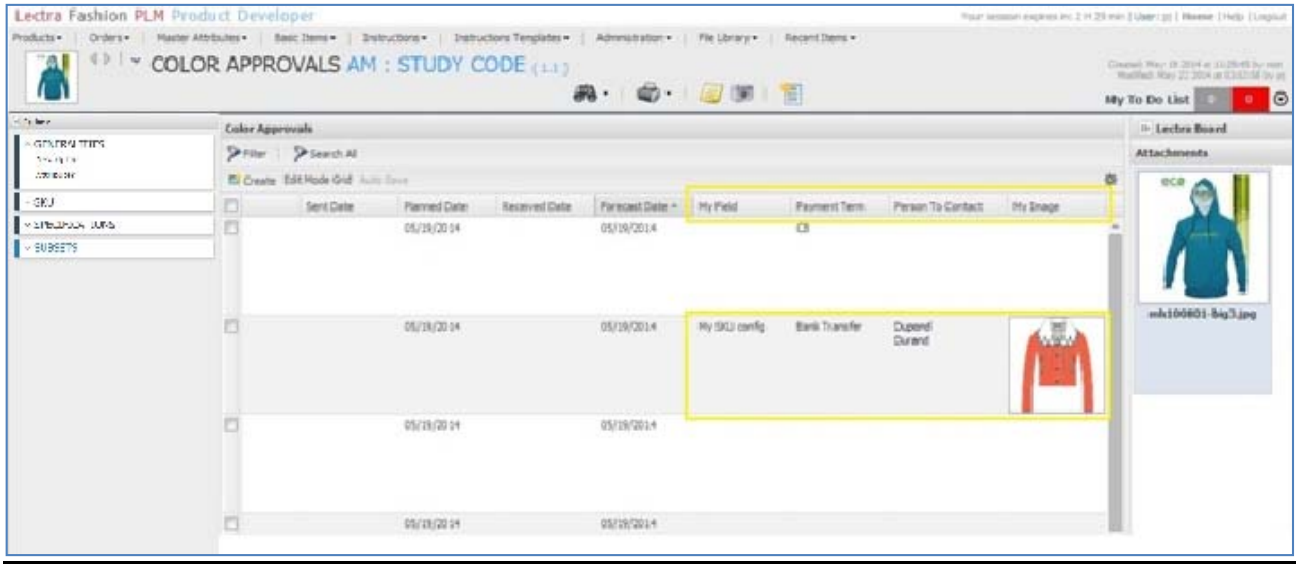
2.1.1 Case

Configure a standard SKU named "ColorApproval" with Custom Fields and Custom Roles.

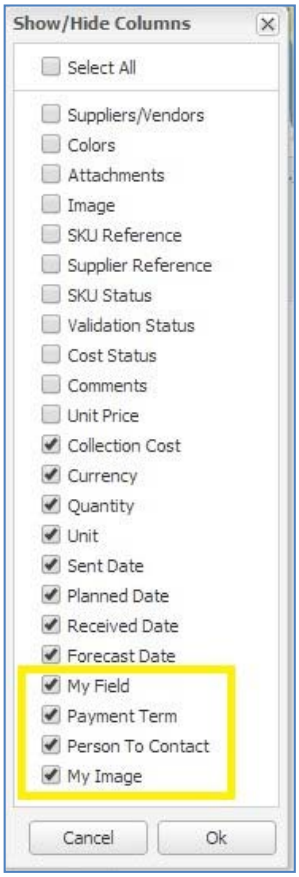
SKU form: "My Configuration" block in My SKU ColorApproval.



SKU Grid in a Product: See here the four custom columns: My Image, My Field, Payment Term, Person To Contact.



Custom fields and roles in the column configurator: See here the four custom columns: My Image, My Field, Payment Term, Person To Contact.



2.1.2 Configuration

PLM Manager Steps

1. Go to Data Configuration - Extend Data Model
2. Extend SKU item
3. Right click on ColorApproval - Add custom fields and custom roles :
   - Add a String CustomField named MyField
   - Add a Pick list CustomRole named PaymentTerm
   - Add a Pick list CustomRole named PersonToContact and checked "Allow multiple"
   - Add a Document Target named MyImage

Configure the Screens SKU Form and SKU Grid in the product

To be able to use the Custom Field and Custom Roles in the SKU form and in the SKU grid of the product, on your PLM server, open and edit the file **Lectra.PDM.SKU.xml** under **LectraPLMParam\namespaces\custom** and add the following XML code.

Step1: declare the "MyConfiguration" block in the SKU form:

Open a panel on Sku "ColorApproval" form and add a fieldset "MyConfiguration" , and declare each custom fields and custom roles:

```
<panel name="mainPanel_ColorApproval" inherits="mainPanel_default">
	<fieldset titleI18n="MyConfiguration" position="1">
		<textfield fieldLabelI18n="MonChamp" name="MyField" />
		<nodecombo fieldLabelI18n=" PaymentTerm " hiddenName="PaymentTerm"
	propertyName="PaymentTerm" propertyFamily="process" multiSelect="false" />
	<!-- change to multiSelect to true for a multiple role -->
		<nodecombo fieldLabelI18n=" PersonToContact " hiddenName="PersonToContact"
	propertyName="PersonToContact" propertyFamily="process" multiSelect="true" />
		<imagefield name="MyImage" fieldLabelI18n=" MyImage " imageSize="small"
	inherits="http://lectra.com/pdm/common#defaultImageField" managerRef="smallImageField">
			<managerInstance name="Lectra.PDM.Mgrs.imageFieldMgr" />
		</imagefield>
	</fieldset>
</panel>
```

Green color represents translations. See step 3.

Step 2: declare the columns in the SKU grid of the product:

Open the explorereditablegrid of the ColorApproval SKU:

```
<explorereditablegrid name="skuSearchResultsGrid_ColorApproval"
inherits="skuSearchResultsGrid_default">
```

Open the colModel (columns) of the ColorApproval SKU and declare the custom fields as columns:

```
<colModel inherits="skuSearchResultsGrid_default/colModel">
	<column xsi:type="textcolumn" headerI18n=" MyField " dataIndex="MyField"
	sortIndex="customFields.MyField" editable="true" />
	<column xsi:type="nodecolumn" headerI18n=" PaymentTerm " dataIndex="PaymentTerm"
	sortIndex="customRoles.PaymentTerm.name" editable="true" />
	<column xsi:type="nodecolumn" headerI18n=" PersonToContact " propertyFamily="process"
	propertyName="PersonToContact" dataIndex="PersonToContact" editable="true"
	multiselect="true" />
	<column xsi:type="imagecolumn" headerI18n=" MyImage " dataIndex="MyImage" editable="true"
	imageSize="medium">
	<imageEditor inherits="http://lectra.com/pdm/common#defaultImageField"
	managerRef="productImageField">
	<managerInstance name="Lectra.PDM.Mgrs.imageFieldMgr" />
	</imageEditor>
	</column>
</colModel>
```

The string custom field MyField is sortable.

*The **non multiselect** list custom role PaymentTerm is sortable.*

*The **multiselect** list custom role PersonToContact is **not** sortable.*

Declare all the dataIndexes of custom fields and the custom roles:

```
<store xsi:type="lpfgroupingstore" inherits="skuSearchResultsGrid_default/store">
	<dataField name="MyField" />
	<dataField name="PaymentTerm" />
	<dataField name="PersonToContact" />
	<dataField name="MyImage" />
</store>
```

Close the SKU ColorApproval ExplorerGrid

```
</explorereditablegrid>
```

Here the view of the consolidate xml **Lectra.PDM.SKU.xml** under **LectraPLMParam\namespaces\custom**:

```
<namespace uri="http://lectra.com/pdm/sku/custom" prefix="sku"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

<panel name="mainPanel_ColorApproval" inherits="mainPanel_default">
	<fieldset titleI18n="MyConfiguration" position="1">
		<textfield fieldLabelI18n="MonChamp" name="MyField" />
		<nodecombo fieldLabelI18n=" PaymentTerm " hiddenName="PaymentTerm"
	propertyName="PaymentTerm" propertyFamily="process" multiSelect="false" />
	<!-- change to multiSelect to true for a multiple role -->
		<nodecombo fieldLabelI18n=" PersonToContact " hiddenName="PersonToContact"
	propertyName="PersonToContact" propertyFamily="process" multiSelect="true" />
		<imagefield name="MyImage" fieldLabelI18n=" MyImage " imageSize="small"
	inherits="http://lectra.com/pdm/common#defaultImageField" managerRef="smallImageField">
			<managerInstance name="Lectra.PDM.Mgrs.imageFieldMgr" />
		</imagefield>
	</fieldset>
</panel>
```

```
<explorereditablegrid name="skuSearchResultsGrid_ColorApproval"
inherits="skuSearchResultsGrid_default">
<colModel inherits="skuSearchResultsGrid_default/colModel">
       <column xsi:type="textcolumn" headerI18n="MonChamp" dataIndex="MyField"
       sortIndex="customFields.MyField" editable="true" />
       <column xsi:type="nodecolumn" headerI18n=" PaymentTerm " dataIndex="PaymentTerm"
       sortIndex="customRoles.PaymentTerm.name" editable="true" />
       <column xsi:type="nodecolumn" headerI18n="P PersonToContact " propertyFamily="process"
       propertyName="PersonToContact" dataIndex=" PersonToContact " editable="true" />
       <column xsi:type="imagecolumn" headerI18n=" MyImage " dataIndex="MyImage" editable="true"
       imageSize="medium">
       <imageEditor inherits="http://lectra.com/pdm/common#defaultImageField"
       managerRef="productImageField">
       <managerInstance name="Lectra.PDM.Mgrs.imageFieldMgr" />
       </imageEditor>
       </column>
</colModel>
<store xsi:type="lpfgroupingstore" inherits="skuSearchResultsGrid_default/store">
       <dataField name=" MyField " />
       <dataField name=" PaymentTerm " />
       <dataField name="PersonToContact" />
       <dataField name=" MyImage " />
</store>
</explorereditablegrid>
</namespace>
```

Step 3: declare the translations :

On the server edit the file: **PLM-Fashion\PDM\LectraPLMParam\i18n\xls\custom\messagesCusto_en.properties.xml**.

Copy/Paste the following content at the end of the file:

```
MyConfiguration =My Configuration
MyField =My Field
PaymentTerm =Payment Term
PersonToContact =Person To Contact
MyImage =My Image
```

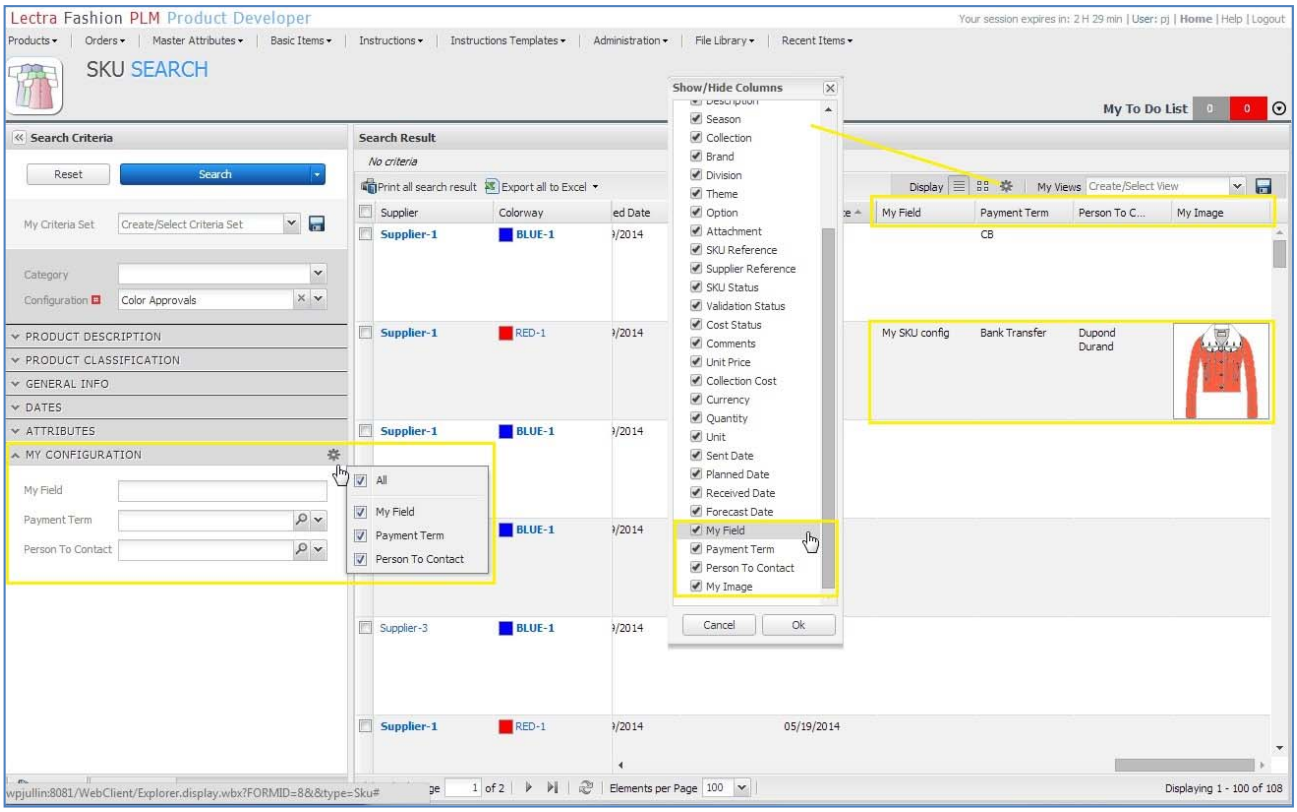
Save the file.

2.2 SKU global Search

2.2.1 Case

Configure the global search SKU menu to be able to search on the custom fields and the custom roles that you have declared in the PLM Manager.

PLM Illustration SKU Global Search: Here we see the search criteria My Field, Payment Term, Person To Contact and in the result the columns: My Image, My Field, Payment Term, Person To Contact. Configurators are also impacted.



2.2.2 Configuration

PLM Manager Steps

1. Prerequisite: you have configured as above your customFields and CutomRoles in the menu Data Configuration->Extend Data Model.
2. Import your Search Criteria on MyField, PaymentTerm and PersonToContact :
   - Create a new .xls file, and paste (in A1 cell) the table below

Entity Type	Entity Configur	Field Name	Field Path	Field Type	Field Type Configuration ((	Field
productmana	ColorApproval	PaymentTerm	customRoles.PaymentTerm	core.Property	process/PaymentTerm	false
productmana	ColorApproval	MyField	customFields.MyField	core.customdefinitions.StringDef		
productmana	ColorApproval	PersonToContact	customRoles.PersonToContact	core.Property	process/PersonToContact	true

- In PLM Manager, select Data Configuration / Define Search Criteria
- Select SKU/SKU_ColorApproval item and click on Import
- Search and import .xls created file

Configure the Screens SKU Form and SKU Grid in the product

To be able to use the Custom Field and Custom Roles as Search Criteria and in the result Skus grid, on your PLM server, open and edit the file **Lectra.PDM.Search.SKU.xml** under **LectraPLMParam\namespaces\custom** and add the following XML code:

Step1: declare Search criteria in the bloc SKUGeneralInfo

Open the Panel on SKU ColorApproval, declare the panel MyConfiguration with custom fields and custom roles:

```
<panel name="criteria_ColorApproval">
	<panel region="north" name="ConfigInstance" inherits="Configuration" />
	<panel region="center" name="accordionInstance" inherits="accordionPanel">

	  <panel titleI18n="MyConfiguration" name="MyConfiguration" collapsible="true"
		layout="form">
		<defaults anchor="100%" />

		<textfield name="MyField.values" fieldLabelI18n=" MyField " value=""
		chkVisible="true" />
		<nodecombo fieldLabelI18n=" PaymentTerm " propertyFamily="process"
		propertyName="PaymentTerm" hiddenName="PaymentTerm.values" chkVisible="false"
		multiSelect="false" />
		<nodecombo fieldLabelI18n=" PersonToContact " propertyFamily="process"
		propertyName="PersonToContact" hiddenName="PersonToContact.values"
		chkVisible="false" multiSelect="true" />
		<!-- change to multiSelect to true for a multiple role -->

	  </panel>
	</panel>
</panel>
```

Step2: declare the columns in the SKU grid of General Search

Open the explorereditablegrid of the ColorApproval SKU:

```
<explorereditablegrid name="resultGrid_ColorApproval" inherits="resultGrid">
```

Open the colModel (columns) of the ColorApproval SKU and declare the custom fields as columns :

```
<colModel inherits="resultGrid/colModel">
	<column xsi:type="textcolumn" headerI18n=" MyField " dataIndex="MyField"
	sortIndex="customFields.MyField" editable="true" />
	<column xsi:type="nodecolumn" headerI18n=" PaymentTerm " dataIndex="PaymentTerm"
	sortIndex="customRoles.PaymentTerm.name" editable="true" propertyName="PaymentTerm"
	propertyFamily="process" />
	<column xsi:type="nodecolumn" headerI18n=" PersonToContact " dataIndex="PersonToContact"
	editable="true" propertyName="PersonToContact" propertyFamily="process"
	multiSelect="true" />
	<column xsi:type="imagecolumn" headerI18n=" MyImage " dataIndex="MyImage" editable="true"
	locked="false" imageSize="medium">
		<imageEditor inherits="http://lectra.com/pdm/common#defaultImageField"
		managerRef="productImageField">
		<managerInstance name="Lectra.PDM.Mgrs.imageFieldMgr" />
		</imageEditor>
	</column>
</colModel>
```

The string custom field MyField is sortable.

*The **non multiselect** list custom role PaymentTerm is sortable.*

*The **multiselect** list custom role PersonToContact is **not** sortable.*

Declare all the dataIndexes of custom fields and the custom roles:

```
<store xsi:type="lpfdirectstore" inherits="resultGrid/store">
	<dataField name="MyField" />
	<dataField name="PaymentTerm" />
	<dataField name="PersonToContact" />
	<dataField name="MyImage" />
</store>
```

Close the SKU ColorApproval explorereditableGrid

```
	</explorereditablegrid>
```

Note: to declare the Columns below My Image, My Field, Payment Term, Person To Contact, the grammar is exactly the same as in Step2 of the §2.1 but the columns are not declared in the same file

Here is the view of the consolidate xml **Lectra.PDM.Search.SKU.xml** under **LectraPLMParam\namespaces\custom**:

```
<namespace uri="http://lectra.com/pdm/search/sku/custom" prefix="searchsku"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:common="http://lectra.com/pdm/common">

<panel name="criteria_ColorApproval">
	<panel region="north" name="ConfigInstance" inherits="Configuration" />
	<panel region="center" name="accordionInstance" inherits="accordionPanel">

		<panel titleI18n=" MyConfiguration " name="MyConfiguration" collapsible="true"
			layout="form">
		<defaults anchor="100%" />
		<textfield name="MyField.values" fieldLabelI18n="MyField" value=""
		chkVisible="true" />
		<nodecombo fieldLabelI18n=" PaymentTerm " propertyFamily="process"
		propertyName="PaymentTerm" hiddenName="PaymentTerm.values" chkVisible="false"
		multiSelect="false" />
		<nodecombo fieldLabelI18n=" PersonToContact " propertyFamily="process"
		propertyName="PersonToContact" hiddenName="PersonToContact.values"
		chkVisible="false" multiSelect="true" />
		<!-- change to multiSelect to true for a multiple role -->
		</panel>

	</panel>
</panel>

<explorereditablegrid name="resultGrid_ColorApproval" inherits="resultGrid">
<colModel inherits="resultGrid/colModel">
	<column xsi:type="textcolumn" headerI18n=" MyField " dataIndex="MyField" sortIndex=
	"customFields.MyField" editable="true" />
	<column xsi:type="nodecolumn" headerI18n=" PaymentTerm " dataIndex="PaymentTerm"
	sortIndex="customRoles.PaymentTerm.name" editable="true" propertyName="PaymentTerm"
	propertyFamily="process" />
	<column xsi:type="nodecolumn" headerI18n=" PersonToContact " dataIndex="PersonToContact"
	editable="true" propertyName="PersonToContact" propertyFamily="process"
	multiSelect="true" />
	<column xsi:type="imagecolumn" headerI18n=" MyImage " dataIndex="MyImage" editable="true"
	locked="false" imageSize="medium">
	<imageEditor inherits="http://lectra.com/pdm/common#defaultImageField"
	managerRef="productImageField">
	<managerInstance name="Lectra.PDM.Mgrs.imageFieldMgr" />
	</imageEditor>
```

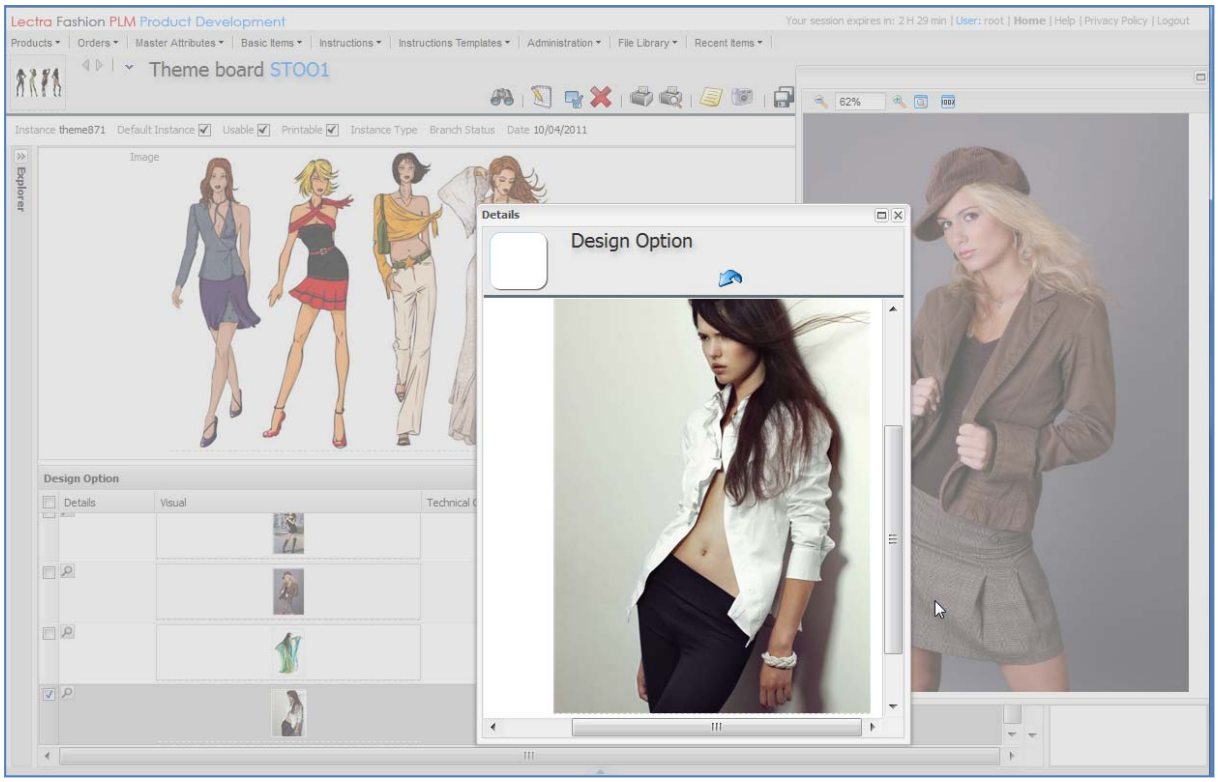
```
        </column>
</colModel>

<store xsi:type="lpfdirectstore" inherits="resultGrid/store">
        <dataField name="MyField" />
        <dataField name="PaymentTerm" />
        <dataField name="PersonToContact" />
        <dataField name="MyImage" />
</store>

</explorereditablegrid>

</namespace>
```

3. SPEC PACKAGE

3.1 Instruction Custom Spec package - New Configuration

3.1.1 Case

Configure a new spec package named "Theme Board" for Style Category. The specification item is an instruction named "Design Option".



3.1.2 Configuration

3.1.2.1 Design Options Instruction:

PLM Manager Steps

1. Menu: Data Configuration->Extend Data Model.
2. Branch Instruction: create a configuration named "Design Option".

Screens

To be able use Designs Option instructions Explorer: on your PLM server, open explorer-specitem.xml under LectraPLMParam/ screens and add the following XML code.

```
<screen name="Explorer_DesignOption" extends="Explorer_SpecItem" extendsMode="Lazy">
        <parameter name="type" value="DesignOption" />
        <parameter name="tableTitle" value="model.DesingOption" />

        <group name="generalGroup">
                <combobox name="group"
                controller="{class:com.lectra.lpf.uimodel.controller.LPFWebUIController,
    params:{lpfExtPath:'http//lectra.com/pdm/search/specitem#Embellichment.group',
```

```
      withValues: true}}" />
        </group>
</screen>
```

3.1.2.2 Theme Board Spec Package

PLM Manager Steps

1. Menu: Data Configuration->Extend Data Model.
2. Branch Style->specifications: create a specification named "Theme Board".

Screens

On your PLM server, open **ScreensCusto.xml** under **LectraPLMParam** and add the following XML code.

```
<screen name="ProductSpecInstance.PlancheTheme" extends="ProductSpecInstance">
        <block name="description" columns="2">
                <field name="defaultSPImageField" />
        </block>
        <table name="links" customizable="true" title="model.DesignOption"
                thumbView="true"
     thumbField="component.defaultImageField.file.imageSet.mediumImage"
                thumbDescription="component.description"
                thumbTitles="component.codeAlpha1, component.codeAlpha2" sortable="true">
                <column name="linePosition" displayed="false" editable="false" />
                <column name="component.defaultImageField" editable="false" />
                <column name="component.defaultImageField.file.imageSet.mediumImage"
                        displayed="false" editable="false" />
                <column name="component.codeAlpha1" editable="false" />
                <column name="component.codeAlpha2" editable="false" />
                <column name="component.description" editable="false" minwidth="150" />
                <column name="component.comment" editable="false" minwidth="150" />
                <column name="comment" editable="true" minwidth="150" />
        </table>
</screen>
```

To be able to pick a Design Option in Theme spec package: on your PLM server, open **explorer-specitem.xml** under **LectraPLMParam/** screens and add the following XML code.

```
<screen name="Explorer_DesignOption_Picker" extends="Explorer_DesignOption"
        extendsMode="lazy">
        <toolbar name="explorerToolBar" extends="pickerToolBar"
                extendsMode="copy" />
</screen>
```

To be able to set default attachment image in the form, add

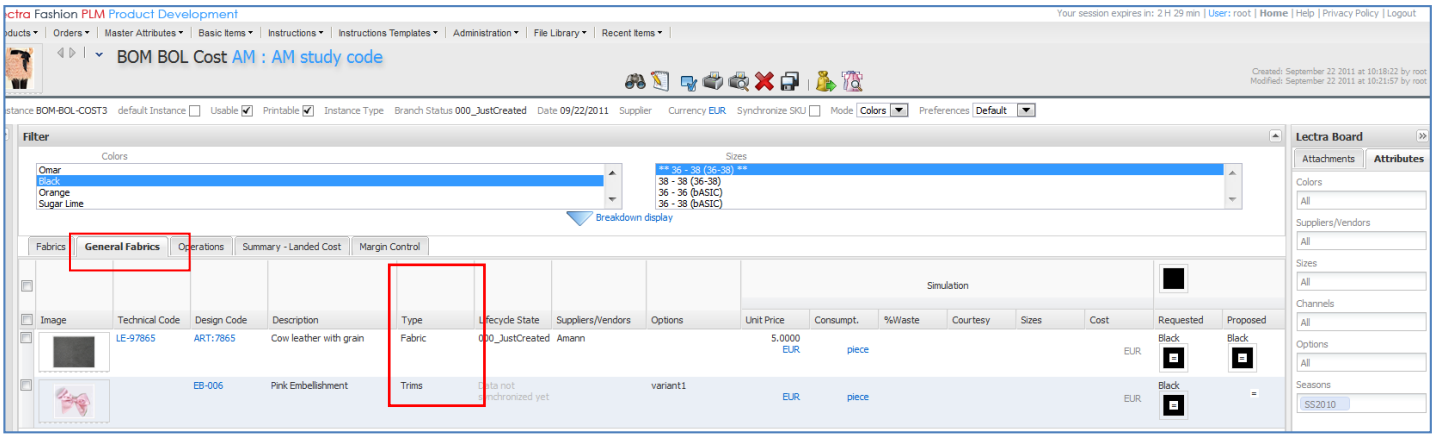
```
<block name="description" columns="2">
        <field name="defaultSPImageField" />
</block>
```

4. BOM BOL COST

No change with the V3R3 customization.

4.1.1 Case

Configure a new BOM Table named GeneralFabrics for Style Category to use Trims and Fabrics components in the same table.



4.1.2 Configuration

PLM Manager Steps

1. Menu: Data Configuration->Extend Data Model.
2. Branch Style/Style/specifications/Style - CostingNG:
   a. Add a Table named: GeneralFabrics.
   b. Choose the categories: Fabric and Trims.
   c. Select Save button
   d. Select Apply button

Configure the screens to declare the new spectable

On your PLM server, open **ScreensCusto.xml** under **PDM\LectraPLMParam** and add inside <screens> xml tag the following XML code:

```
<!--          GeneralFabrics Bom Table                                          -->
<screen name="specPackages.ProductAdvancedCostingInstance.Style"
extends="specPackages.ProductAdvancedCostingInstance" >
        <block name="costingContainer" >
        <tabs name="costingTabs">
        <!-- Declare your new spectable name GeneralFabrics -->
           <tab name="GeneralFabricsTab" i18nKey="GeneralFabrics"
        associatedDtoName="costBOMSpLinks" autoLoad="true" lockable="true"
        lockAction="Lectra.PDM.SpecPackage.Costing.tableLock" autoHeight="true">
            <table name="GeneralFabrics">
                   sortable ="true"
                   customizable="true"
                   scrollable="true"
                   collapsible="true"
                   lockColumn="true"
                   displayBbar="true"
```

```
                    maxHeightForFixedCols="380"
                    defaultFixedColumnsNumber="3"
                    displayActionsLabels="false"
                    thumbView="true"
        thumbField="componentProduct.multiMediaBoard.default.fileLink.file.imageSet.mediumImage"
                    thumbDescription="componentProduct.codeAlpha1"
                    enableColId="false"
                    displayTitle="false"
                    xtype="Lectra.PDM.Quirks.BreakdownGridPanel">
                <column name="linePosition" displayed="false" editable="false" sortableBy="linePosition"
showable="false" />
                <column editable="false" name="componentProduct.defaultImageField" minwidth="90"/>
                <column editable="false"
name="componentProduct.multiMediaBoard.default.fileLink.file.imageSet.mediumImage" displayed="false"/>
<!-- only for thumb mode -->

                <column editableIf="componentProduct.local" editable="false"
name="componentProduct.codeAlpha1" minwidth="90" autoComplete="true" canCreate="false"
displayKey="codeAlpha1" differentialKeys="codeAlpha2" pick="false"/>
                <column editableIf="componentProduct.local" editable="false"
name="componentProduct.codeAlpha2" minwidth="90" autoComplete="true" canCreate="false"
displayKey="codeAlpha2" differentialKeys="codeAlpha1" pick="false"/>
                <column editableIf="componentProduct.local" editable="false"
name="componentProduct.description" minwidth="150"/>

                <column editable="false" name="componentProduct.categoryName" minwidth="90"/>
                <column editable="false" name="componentProduct.productStatus" minwidth="90"/>
                <column editable="true" name="classification" minwidth="90"/>
                <column editable="true" name="position" minwidth="90"/>
                <column editable="true" name="role" minwidth="90"/>
                <column editable="true" name="comment" minwidth="150"/>
                <column name="colorInvariant" editable="true" minwidth="90"/>
                <column name="purchaseUnit" editable="false" minwidth="110"/>
                <column name="buyingUnit" editable="false" minwidth="110"/>
                <column name="defaultPurchasePrice" editable="false" minwidth="110"/>

                <column name="suppliers" editable="true" minwidth="110"/>
                <column name="options" editable="true" minwidth="110"/>
                <column editable="true" name="simulation" displayed="true" sortable="false">
                        <column name="BOMUnitPrice" editable="true" sortable="false" minwidth="80"/>
                        <column name="consumption" editable="true" sortable="false" minwidth="80"/>
                        <column name="wastage" editable="true" sortable="false" minwidth="80"/>
                        <column name="courtesy" editable="true" sortable="false" minwidth="80"/>
                        <column name="size" editable="true" sortable="false" minwidth="80"/>
                        <column name="simulationBOMCost" editable="true" sortable="false" minwidth="80"/>
                </column>
                <column editable="true" name="breakdownColor" displayed="false" sortable="false">
                        <column editable="true" name="requestedColor" minwidth="80"/>
                        <!-- can not use displayed="false" for componentAv : delete the line -->
                        <column editable="true" name="componentAv" minwidth="80"/>
                        <column editable="true" name="comment" minwidth="80"/>
                        <column editable="false" name="unitCost" minwidth="80"/>
                        <column editable="true" name="negociatedCost" minwidth="80"/>
                        <column editable="false" name="costBOM" minwidth="80"/>
                </column>
                <column editable="true" name="breakdownNOColor" displayed="false" sortable="false">
                        <column editable="true" name="requestedColor" minwidth="80"/>
                        <!-- can not use displayed="false" for componentAv : delete the line -->
                        <column editable="true" name="componentAv" minwidth="80"/>
                </column>
                <column editable="true" name="breakdownSize" displayed="false" sortable="false">
                        <!-- can not use displayed="false" for componentAv : delete the line -->
                        <column editable="true" name="componentAv" minwidth="80"/>
                        <column editableIf="componentProduct.local" editable="false" name="proposedSizeUnit"
minwidth="80"/>
                        <column editable="true" name="consumptionBOM" minwidth="80"/>
                        <column editable="false" name="unitCost" minwidth="80"/>
                        <column editable="true" name="negociatedCost" minwidth="80"/>
                        <column editable="false" name="costBOM" minwidth="80"/>
                </column>
                <column editable="true" name="breakdownNOSize" displayed="false" sortable="false">
                        <!-- can not use displayed="false" for componentAv : delete the line -->
```

```
                        <column editable="true" name="componentAv" minwidth="80"/>
                        <column editableIf="componentProduct.local" editable="false" name="proposedSizeUnit"
minwidth="80"/>
                        <column editable="true" name="consumptionBOM" minwidth="80"/>
                    </column>
                </table>
                </tab>
            </tabs>
            </block>
</screen>
<!-- End GeneralFabrics Bom Table -->
```

Some observations:

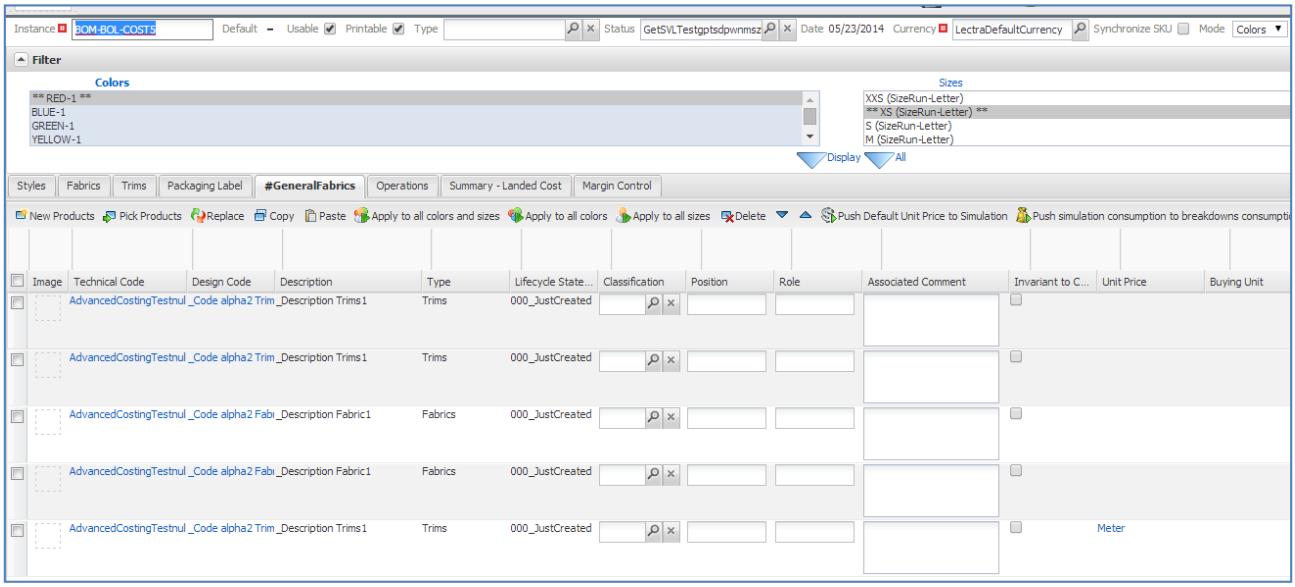
- The Columns of GeneralFabrics tab are a merge of Trims and Fabrics columns.
- Columns no editable (editable="false") could be removed from the GeneralFabrics tab, permitting to reduce the set of columns in this tab.

Next steps:

- restart the PLM.
- After login, reload the PLM configuration

  <host>/pdm/admin.SessionMonitor.reloadConfig.wbx

- Create a BOM BOL COST.
- Pick Products (Trims and Fabrics)
- You must obtain a screen like the following :



For more information on BOM BOL COST customization, you can consult the document Lectra_Fashion_PLM_V4R1_Configuration_Exercises.pptx (page 29).

5. CALENDAR

5.1 Steps

5.1.1 How customize steps in Time & Actions?

5.1.1.1 How to add a custom field "Task Type" applied to tasks and use it in Time & Actions?

PLM Manager Steps

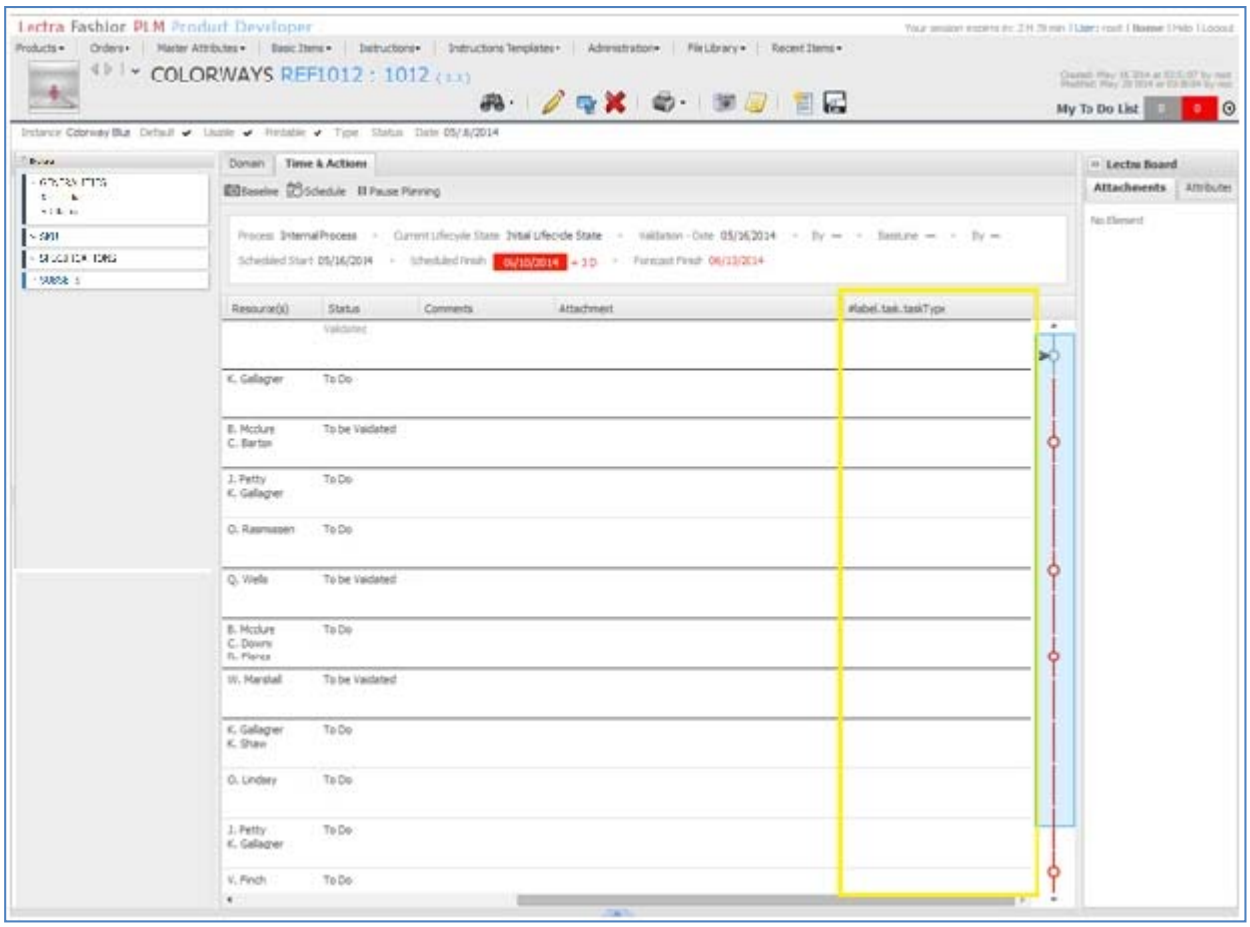
1. Go to Data Configuration - Extend Data Model
2. Extend Calendar Management item
3. Right click on Task - Add custom field
4. Add field named **taskType** as String field

Screen custo

On file **namespaces/custom/Lectra.PDM.TimeAndAction.xml**, between <namespace> tags, add the following XML code:

```
<pdmbasicgrid name="planningGrid_custom" inherits="planningGrid_default">
   <colModel inherits="planningGrid_default/colModel">
      <column xsi:type="textcolumn" dataIndex="taskType" editable="true"
       headerI18n="label.task.taskType" sortable="true" />
   </colModel>
   <store xsi:type="lpfdirectstore" inherits="planningGrid_default/store">
      <dataField name="taskType" />
   </store>
</pdmbasicgrid>
```

Screen Result



5.1.1.2 How to add a new field "Deliverable document" to Lifecycle States?

PLM Manager

1. Go to Data Configuration - Extend Data Model
2. Extend Calendar Management item
3. Right click on LifeCycleState - Add custom role
4. Add a field named **deliverableDoc** as Document target field

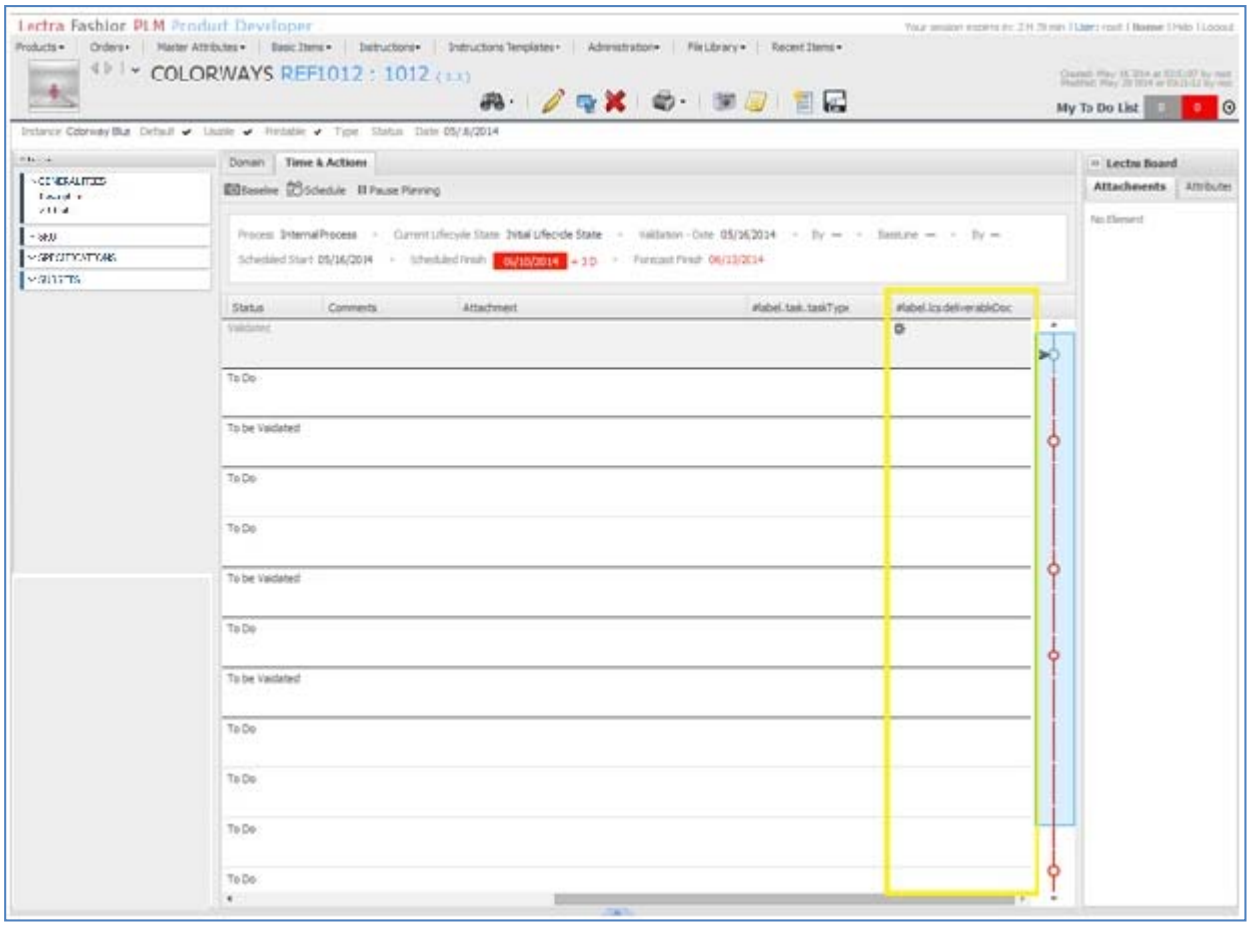
Screen custo

On file **namespaces/custom/Lectra.PDM.TimeAndAction.xml**, to add new column Deliverable document, update XML with adding a new column and a new dataField:

```
<pdmbasicgrid name="planningGrid_custom" inherits="planningGrid_default">
   <colModel inherits="planningGrid_default/colModel">
      <column xsi:type="textcolumn" dataIndex="taskType" editable="true"
       headerI18n="label.task.taskType" sortable="true" />
      <column xsi:type="imagecolumn" headerI18n="label.lcs.deliverableDoc"
       dataIndex="deliverableDoc" editable="true" hideThumbnail="true"
       showLabel="true" cls="pdm-planning-step-skipped-ignore">
         <imageEditor inherits=http://lectra.com/pdm/common#defaultImageField
          managerRef="mediumImageField">
            <managerInstance name="Lectra.PDM.Mgrs.imageFieldMgr"/>
         </imageEditor>
```

```
      </column>
   </colModel>
   <store xsi:type="lpfdirectstore" inherits="planningGrid_default/store">
  <dataField name="taskType" />
      <dataField name="deliverableDoc" />
   </store>
</pdmbasicgrid>
```

Screen Result



5.1.1.3 How to add a new field "Time Spent" to both Lifecycle States and tasks?

PLM Manager

1. Go to Data Configuration - Extend Data Model
2. Extend Calendar Management item
3. Right click on LifeCycleState - Add custom field
4. Add field named **timeSpent** as Integer field
5. Right click on Tasks - Add custom field
6. Add field named **timeSpent** as Integer field

Screen custo

On file **namespaces/custom/Lectra.PDM.TimeAndAction.xml**, between colModel tags, below previous columns added, add the following XML code:

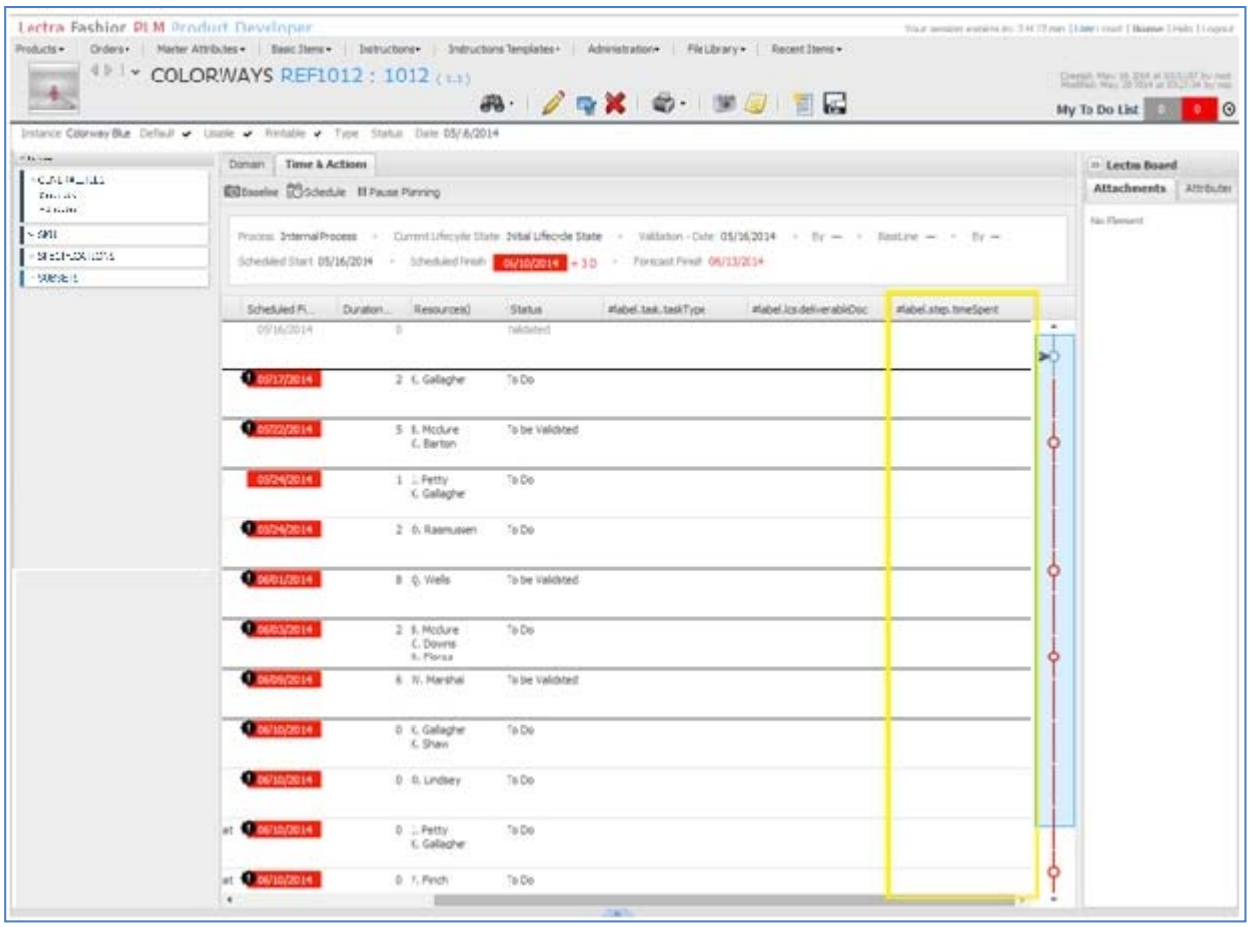
```
<column xsi:type="numbercolumn" sortable="true" dataIndex="timeSpent" format="0"
  headerI18n="label.step.timeSpent" editable="true">
    <editorField xsi:type="spinnerfield" allowDecimals="false" />
</column>
```

NB: <*editorField*> tag is optional, it changes field into spinner field and prohibits decimals.

And between store tags, below previous column added, add the following XML code :

```
<dataField name="timeSpent" />
```

Screen Result



5.1.2 How to implement this customization to the Subset Explorer?

5.1.2.1 In subset explorer, with Calendar configuration, how to display new custom fields on tasks?

Screen custo

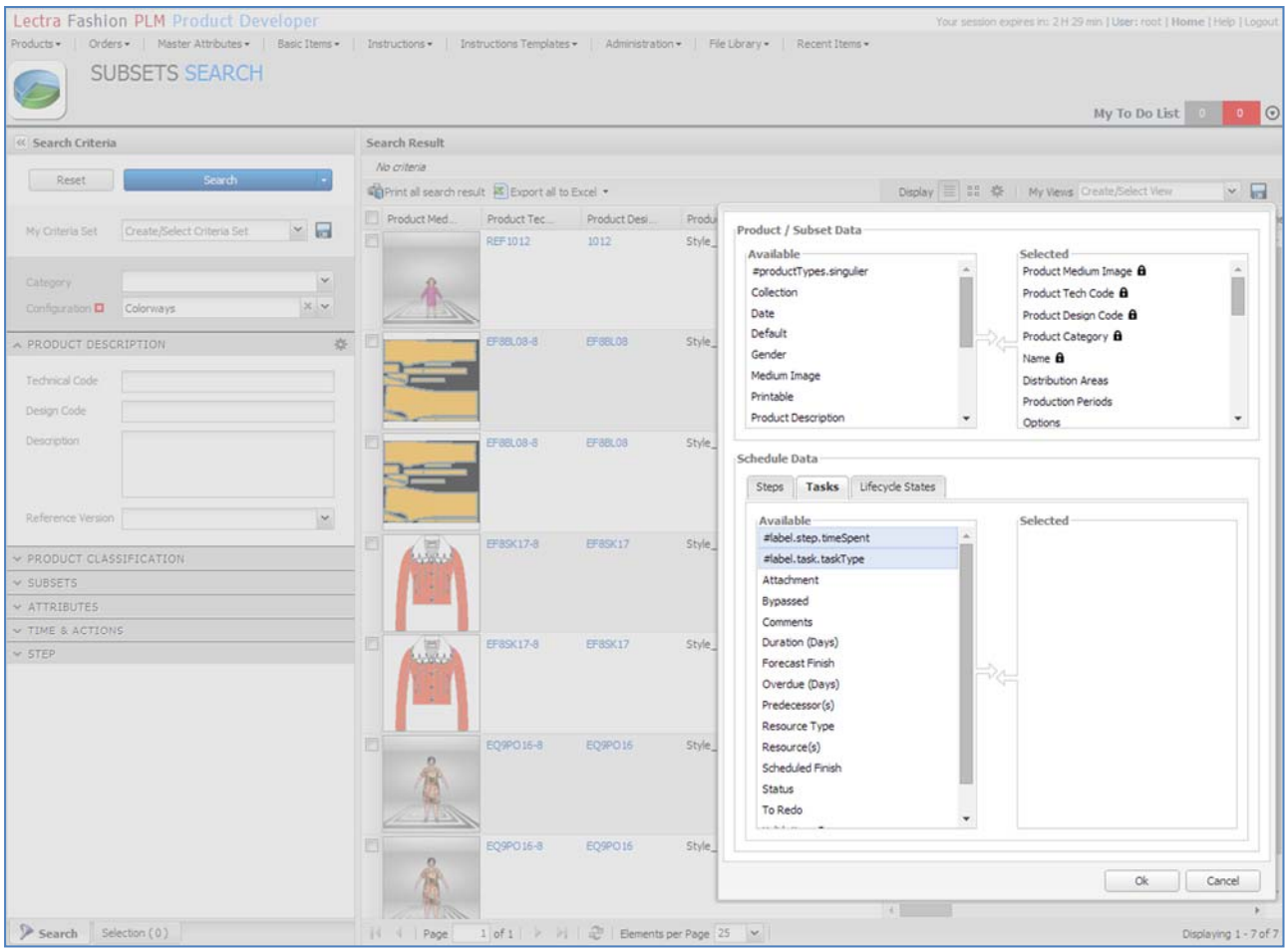
On file **namespaces/custom/Lectra.PDM.Search.Assortment.xml**, between namespace tags, add the following XML code

```
<explorereditablegrid name="resultGrid_with_workflow_custo"
 inherits="resultGrid_with_workflow_default">
   <colModel inherits="resultGrid_with_workflow_default/colModel">
   </colModel>
   <store xsi:type="lpfdirectstore"
    inherits="resultGrid_with_workflow_default/store" >
     <dataField name="${#stepPath}.deliverableDoc" genStepField="deliverableDoc"
      arrayIndex="true" />
     <dataField name="${#stepPath}.timeSpent" genStepField="timeSpent"
      arrayIndex="true" />
   </store>
</explorereditablegrid>

<groupcolumn name="taskColumns_custo" inherits="taskColumns_default" >
   <column xsi:type="textcolumn" dataIndex="${#stepPath}.taskType"
    headerI18n="label.task.taskType" name="taskType" id="${#stepId}_taskType"
    width="100" editable="true" sortable="true" wfGroupable="true" />
   <column xsi:type="numbercolumn" sortable="true"
    dataIndex="${#stepPath}.timeSpent" format="0"
    headerI18n="label.step.timeSpent" editable="true" name="timeSpent"
    id="${#stepId}_timeSpent" wfGroupable="true" >
     <editorField xsi:type="spinnerfield" allowDecimals="false" />
   </column>
</groupcolumn>
```

NB: Make sure *<column>* tags have to be mutually identical in two custom files (**Lectra.PDM.Search.Assortment.xml** and **Lectra.PDM.TimeAndAction.xml**), except for some differences:

1. *<column>* tags have to include attributes in Lectra.PDM.Search.Assortment.xml (XXX is custom field name) :
   - dataIndex="${#stepPath}.**XXX**"
   - name="**XXX**"
   - id="${#stepId}_**XXX**"
   - wfGroupable="true"
2. *<dataField>* tags have to include attributes in **Lectra.PDM.Search.Assortment.xml** :
   - genStepField ="**XXX**"
   - name="${#stepPath}.**XXX**"
   - arrayIndex="true"

Screen Result



5.1.2.2 How to display new Lifecycle states custom fields too?

Screen custo

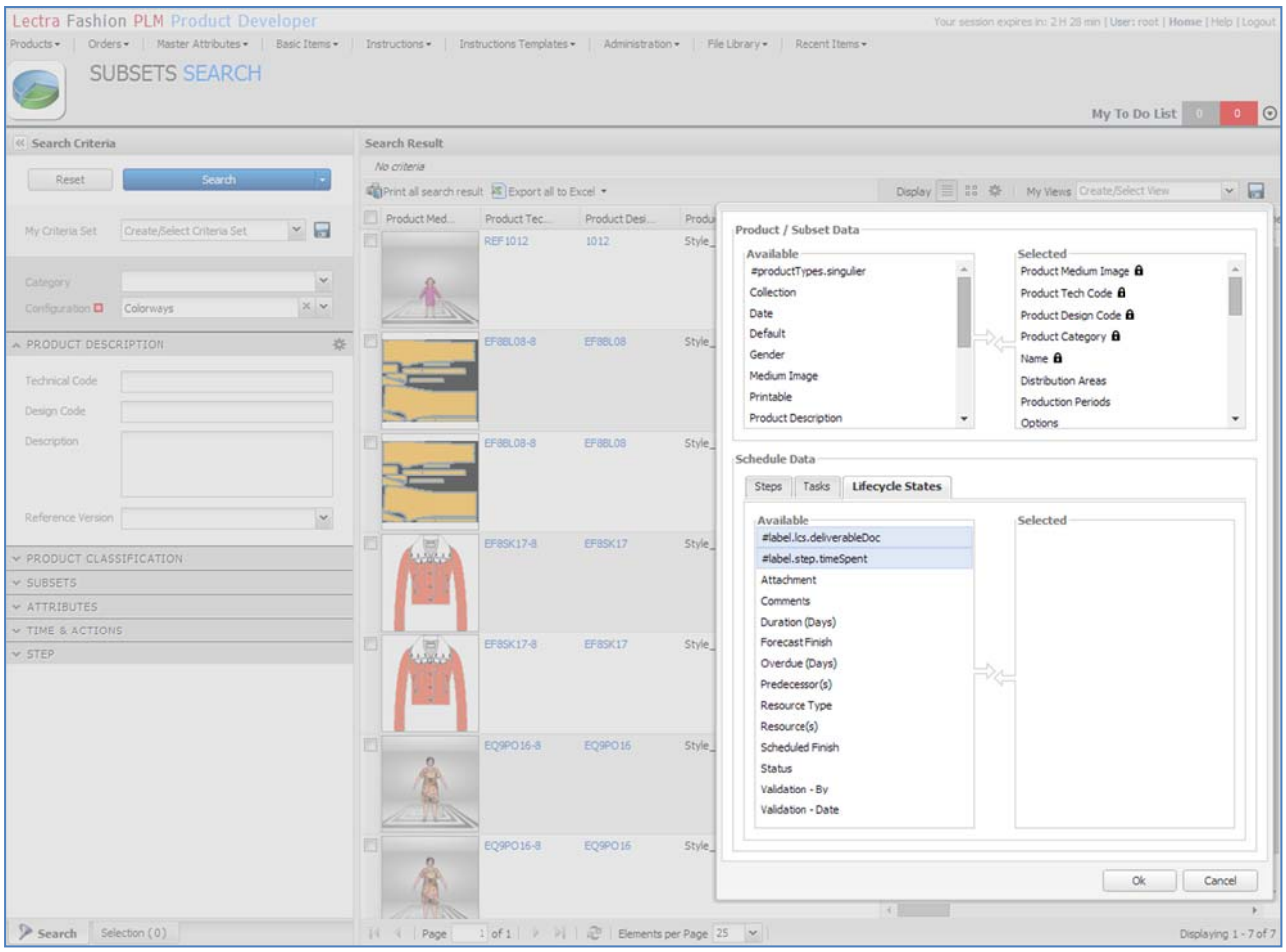
On file **namespaces/custom/Lectra.PDM.Search.Assortment.xml**, add the following XML code:

```
<groupcolumn name="lcsColumns_custo" inherits="lcsColumns_default" >
   <column xsi:type="imagecolumn" dataIndex="${#stepPath}.deliverableDoc"
    imageSize="medium" hideThumbnail="true" showLabel="true"
    headerI18n="label.lcs.deliverableDoc" name="deliverableDoc"
    id="${#stepId}_deliverableDoc" wfGroupable="true" editable="true">
      <imageEditor inherits="”http://lectra.com/pdm/common#defaultImageField”
       managerRef="mediumImageField">
          <managerInstance name="Lectra.PDM.Mgrs.imageFieldMgr"/>
      </imageEditor>
   </column>
   <column xsi:type="numbercolumn" sortable="true"
    dataIndex="${#stepPath}.timeSpent" format="0"
    headerI18n="label.step.timeSpent" editable="true" name="timeSpent"
    id="${#stepId}_timeSpent" wfGroupable="true" >
     <editorField xsi:type="spinnerfield" allowDecimals="false" />
   </column>
</groupcolumn>
```

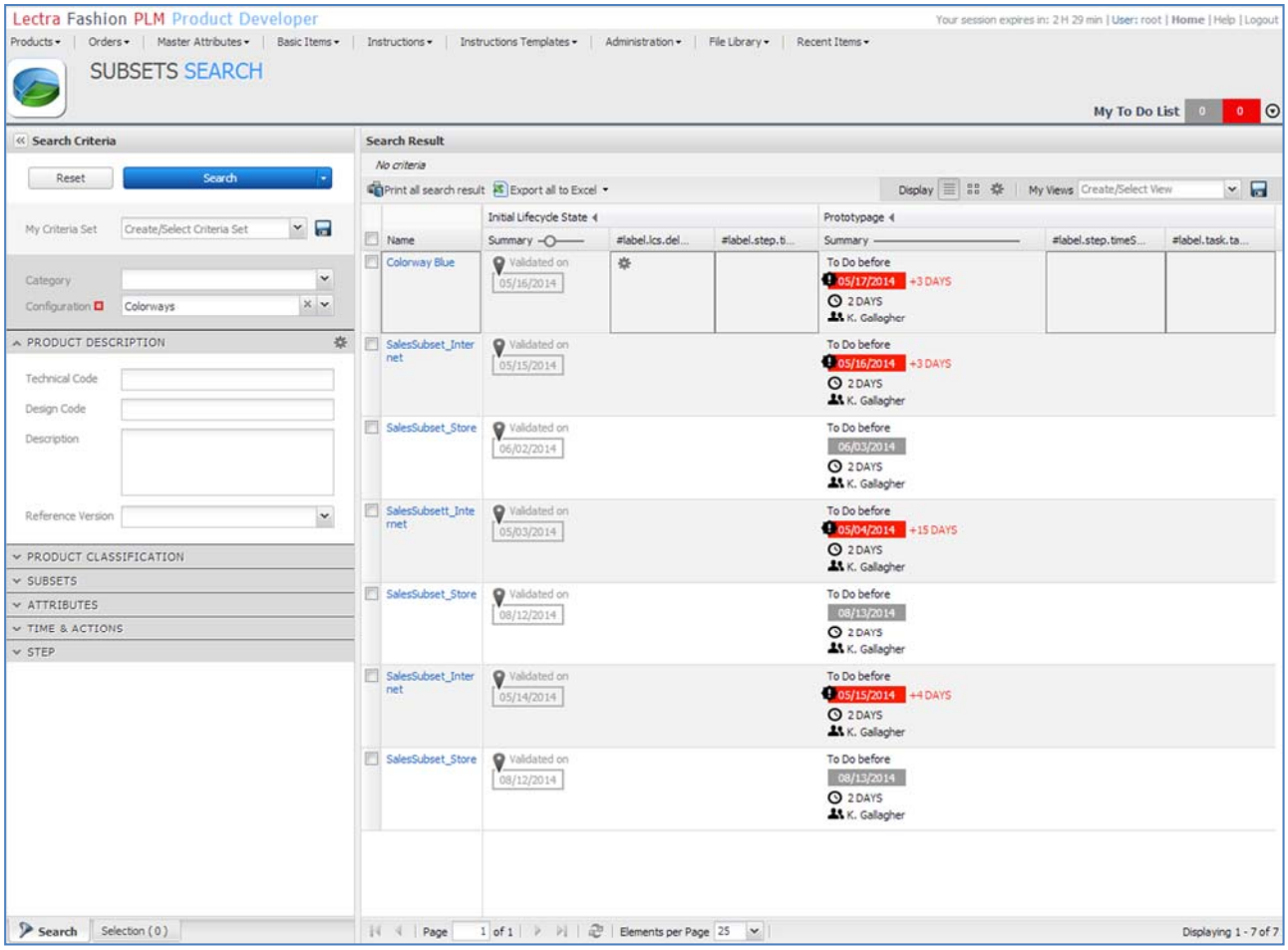
And between explorereditablegrid / store tags, below previous dataField added, add the following XML code:

```
<dataField name="${#stepPath}.taskType" genStepField="taskType" arrayIndex="true" />
```

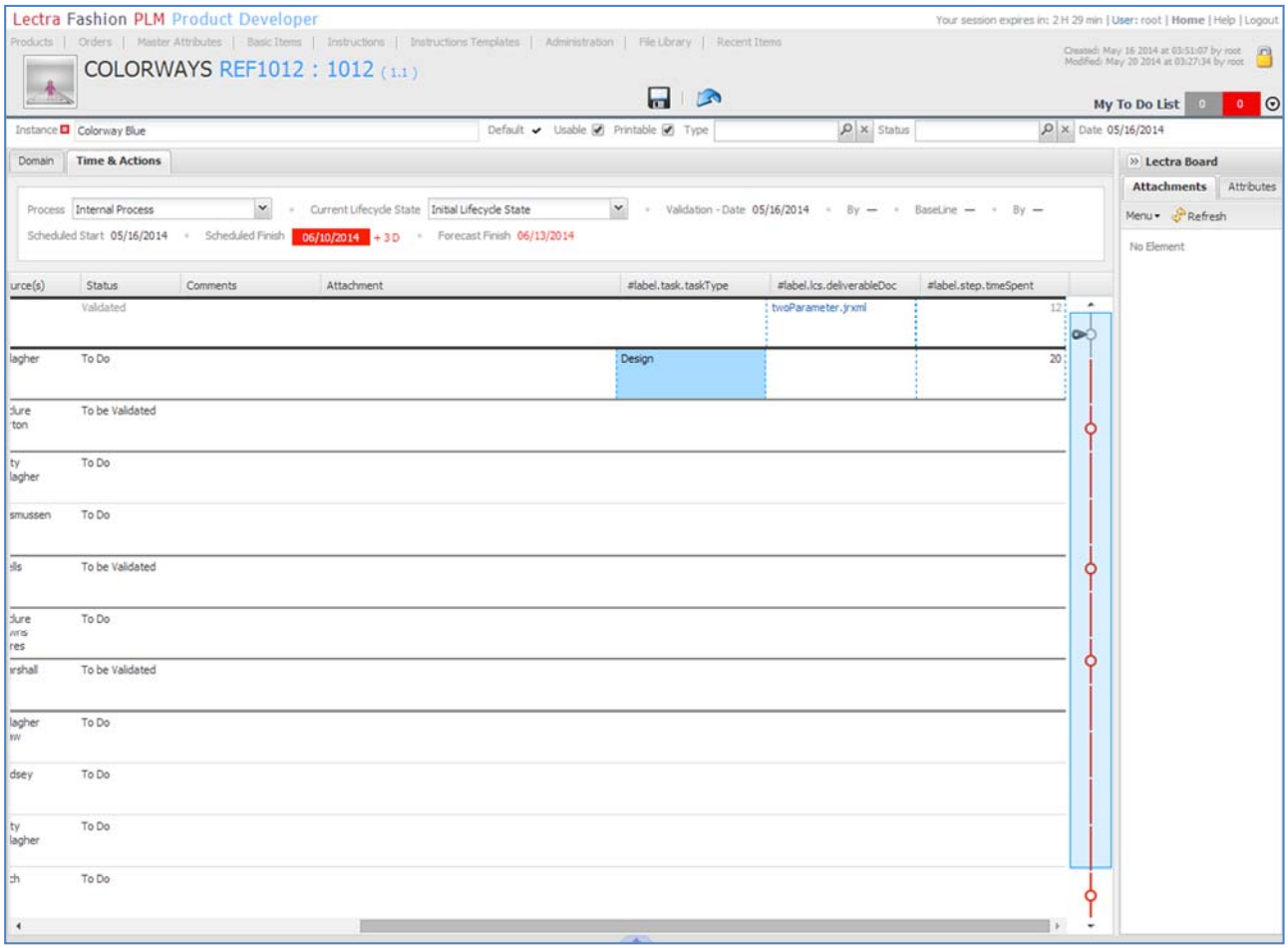
Screen result (columns configurator)



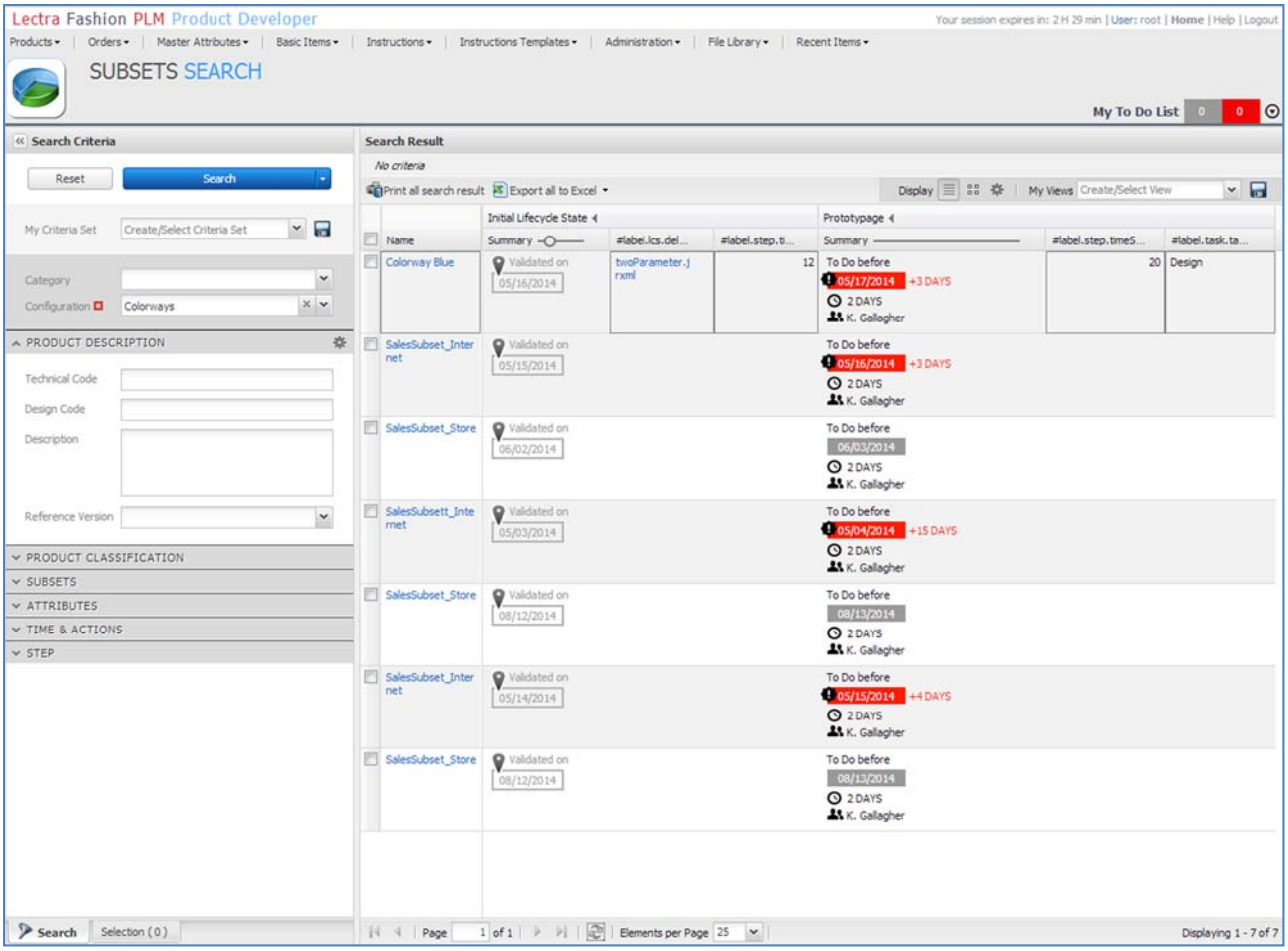
5.1.2.3 Result display when steps expanded



5.1.2.4 Update new custom fields values in Time&Action screen of first displayed subset



5.1.2.5 Compare values in the subset explorer



5.2 Planning

5.2.1 How to customize schedules in Time & Actions?

5.2.1.1 How to add a custom long text field "Notes" applied to schedules in the Time & Actions, below standard schedule fields?

PLM Manager

3. Go to Data Configuration - Extend Data Model
4. Extend Calendar Management item
5. Right click on Schedule - Add custom field
6. Add field named **notes** as Long string field

Screen custo

On file **namespaces/custom/Lectra.PDM.TimeAndAction.xml**, between *namespace* tags, add the following XML code

```
<fieldset name="fieldsetform_custom" inherits="fieldsetform_default">
   <textarea name="notes" fieldLabelI18n="label.planning.notes"
    dataIndex="notes"/>
</fieldset>
```

5.2.1.2 How to add a custom date field "Deadline" applied to planning in Time & Actions, beside standard planning dates?

PLM Manager

1. Go to Data Configuration - Extend Data Model
2. Extend Calendar Management item
3. Right click on Schedule - Add custom field
4. Add field named **deadlineDate** as Date field

Screen custo

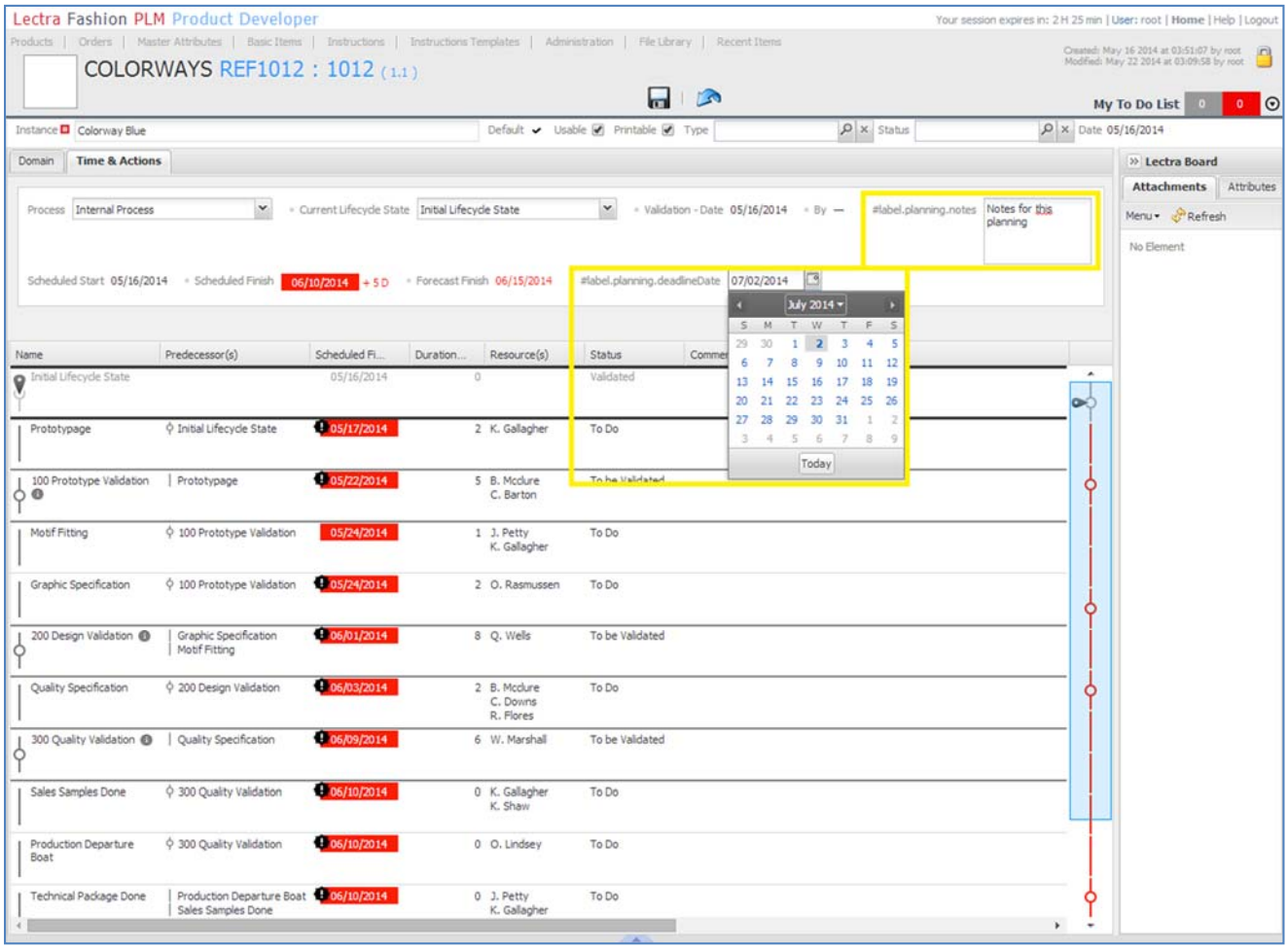
On file **namespaces/custom/Lectra.PDM.TimeAndAction.xml**, between *namespace* tags, add the following XML code

```
<fieldset name="fieldsetform_date_custom" inherits="fieldsetform_date_default" >
   <datefield name="deadlineDate" fieldLabelI18n="label.planning.deadlineDate"
    dataIndex="deadlineDate"/>
</fieldset>
```

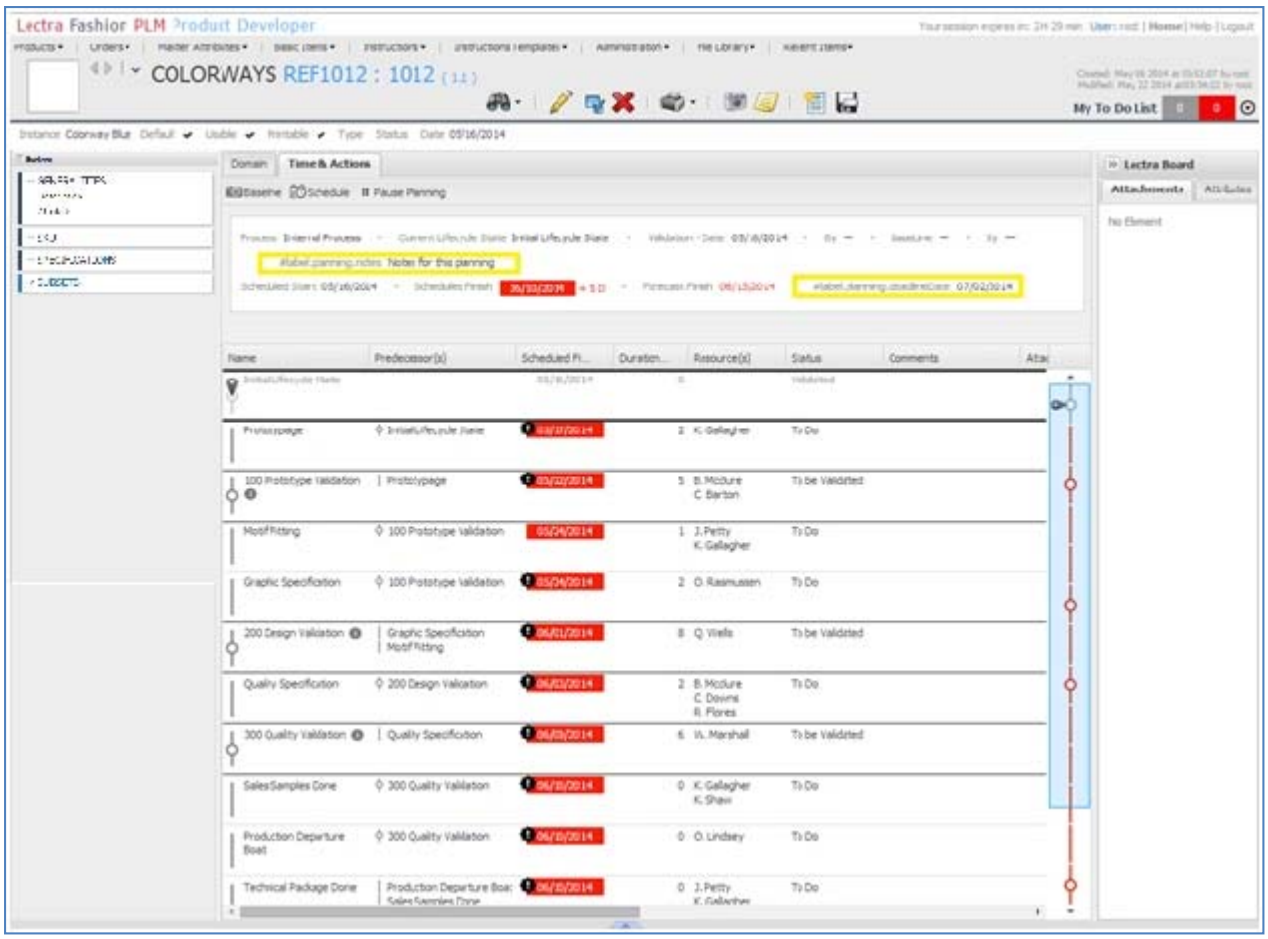
NB : fieldset named "`fieldsetform_custom`" is used for first line of planning attributes (with Process, Current Lifecycle State etc.), while fieldset named "`fieldsetform_date_custom`" is used for second line of schedule attributes (with dates : Scheduled Start, Scheduled finish etc.)

Screen Result

In edit mode:



In display mode, after save:



5.2.1.3 How to find custom field "Notes" on Subset Explorer, whatever calendar configuration used?

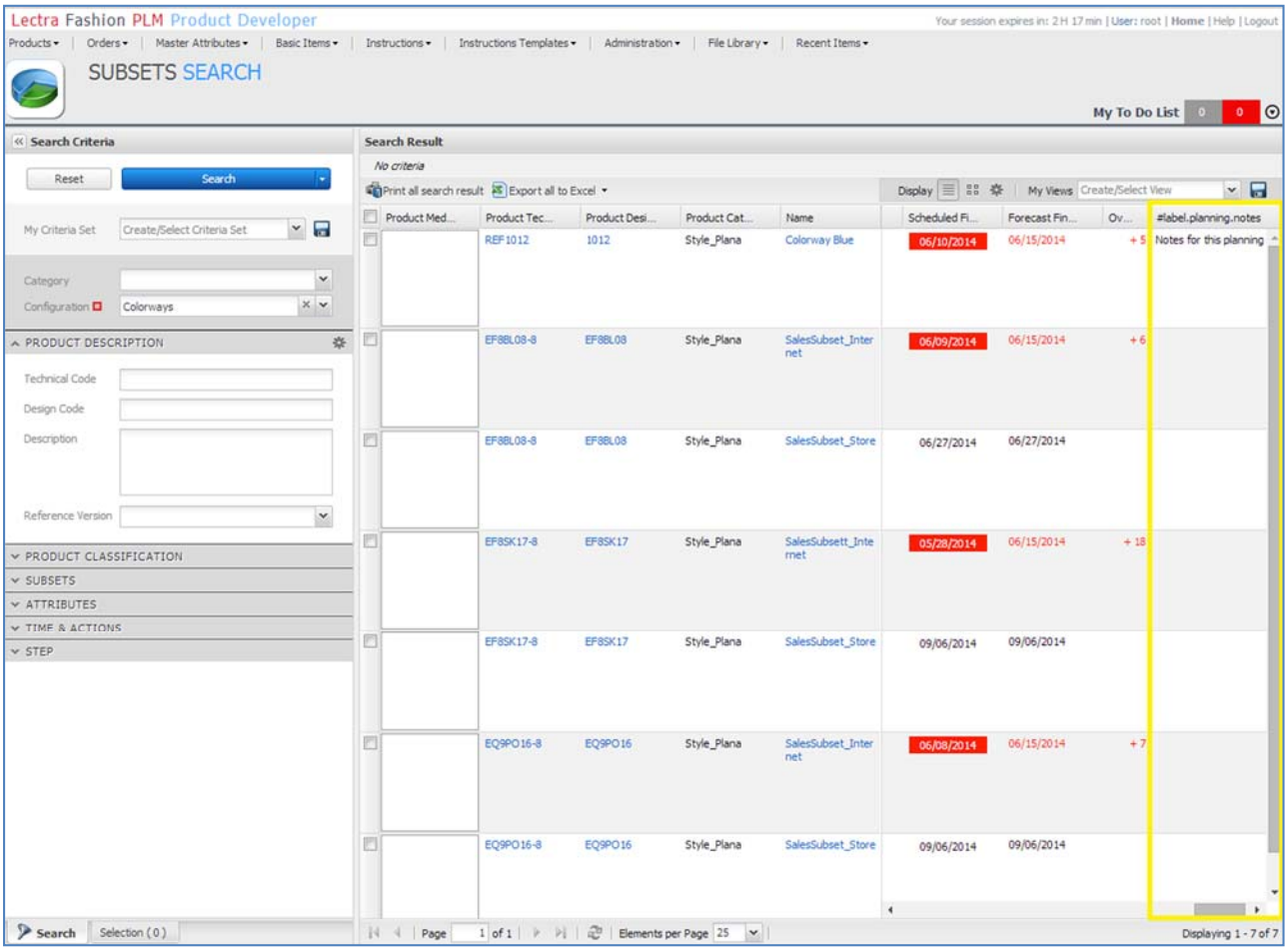
Screen custo

On file **namespaces/custom/Lectra.PDM.Search.Assortment.xml**, between namespace tags, add the following XML code

```
<explorereditablegrid name="resultGrid_with_workflow_custo"
inherits="resultGrid_with_workflow_default">
   <colModel inherits="resultGrid_with_workflow_default/colModel"  >
      <column xsi:type="longtextcolumn" dataIndex="workflowData.planning.notes"
       headerI18n="label.planning.notes" editable="true" />
   </colModel>
   <store xsi:type="lpfdirectstore"
    inherits="resultGrid_with_workflow_default/store" >
      <dataField name="workflowData.planning.notes" />
   </store>
</explorereditablegrid>
```

NB: In Subset explorer, custom schedule fields have to be prefixed by "**workflowData.planning**" in dataIndex of column and in dataField of store.

Screen Result



5.2.1.4 How to add custom field "Deadline" on Subset Explorer, only with "Colorways with deadline" configuration?

Screen custo

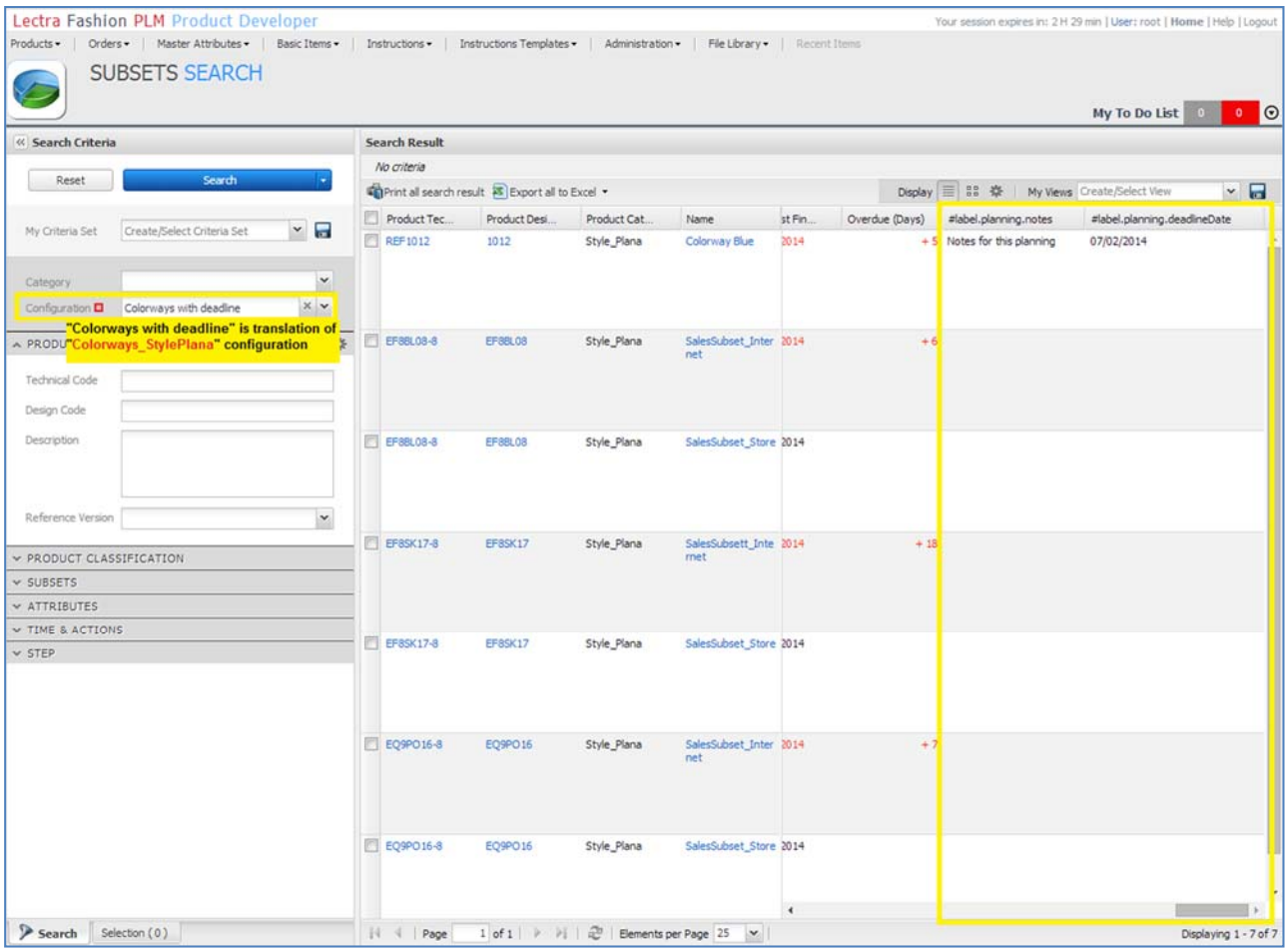
On file **namespaces/custom/Lectra.PDM.Search.Assortment.xml**, after first *<explorereditablegrid>* added, add the following XML code:

```
<explorereditablegrid name="resultGrid_Colorways_StylePlana"
  inherits="resultGrid_with_workflow_custo">
    <colModel inherits="resultGrid_with_workflow_custo/colModel"  >
       <column xsi:type="datecolumn"
        dataIndex="workflowData.planning.deadlineDate"
        headerI18n="label.planning.deadlineDate" editable="true"/>
    </colModel>
    <store xsi:type="lpfdirectstore"
     inherits="resultGrid_with_workflow_custo/store">
       <dataField name="workflowData.planning.deadlineDate" />
    </store>
</explorereditablegrid>
```

NB: To keep previous custo and inherits "resultGrid_Colorways_StylePlana" of "resultGrid_with_workflow_custo", all **resultGrid_with_workflow_default** references have to be replaced by **resultGrid_with_workflow_custo**.

NB2: Configuration name suffix "**Colorways_StylePlana**" is the Configuration key, defined in PLM Manager and translated in **Administration > Internationalization** on PDM.

Screen Result



5.2.1.5 How to create the associated search criteria?

PLM Manager

1. Create a new .xls file, and paste (in A1 cell) the table below

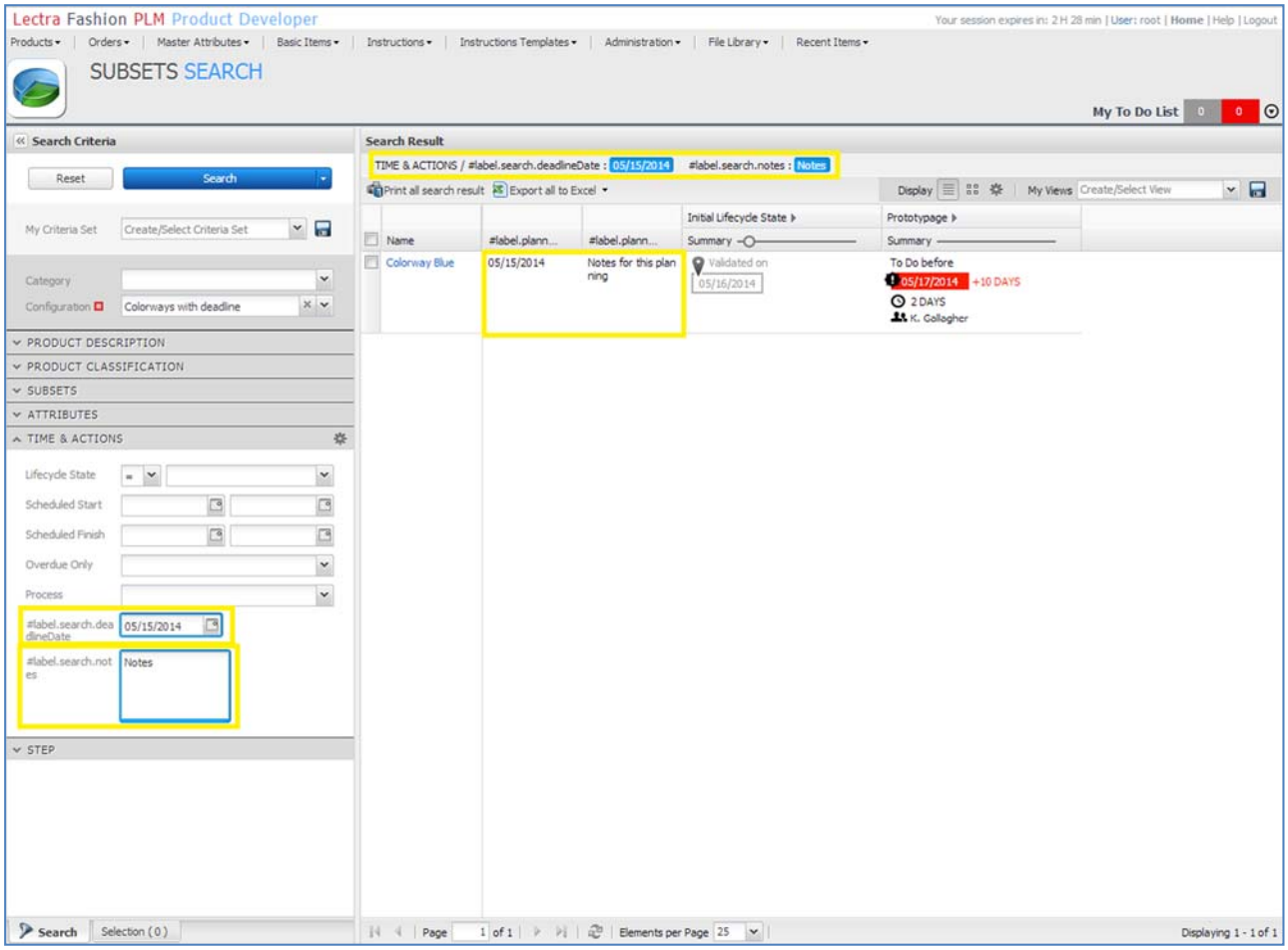
{core	{core.	{core.qualifica	{core.qualification.SearchableBinding}.conditionDefs	{c	{core.qualifi	{core.qualifi	{core.qual
Entit	Entity	Field Name	Field Path	Fi	Field Type	Field Type C	Field Type
prod	Color	taskType	workflowData.planning.tasks.customFields.taskType		core.customdefinitions.StringDef		
prod	Color	taskTimeSpen	workflowData.planning.tasks.customFields.timeSpent		core.customdefinitions.Intdef		
prod	Color	lcsTimeSpent	workflowData.planning.states.customFields.timeSpent		core.customdefinitions.Intdef		
prod	Color	deadlineDate	workflowData.planning.customFields.deadlineDate		core.customdefinitions.DateDef		
prod	Color	notes	workflowData.planning.customFields.notes		core.customdefinitions.LongStringD		

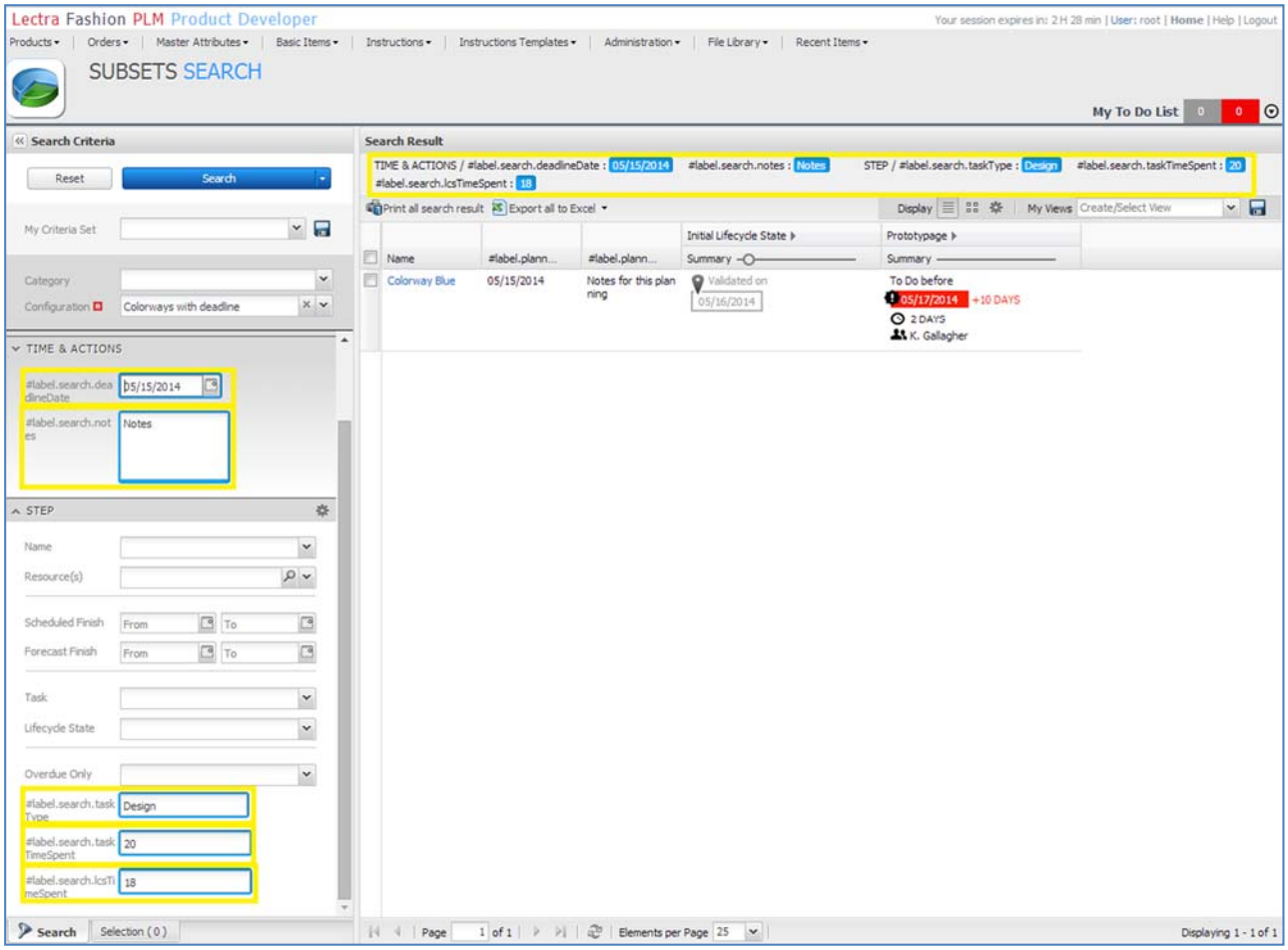
2. In PLM Manager, select Data Configuration / Define Search Criteria
3. Select Partition_**Colorways_StylePlana**and click on Import
4. Search and import .xls created file

Screen custo

On file **namespaces/custom/Lectra.PDM.Search.Assortment.xml**, between namespace tags, add the following XML code

```
<panel name="criteria_Colorways_StylePlana" >
   <panel region="north" name="ConfigInstance" inherits="Configuration" />
   <panel region="center" name="accordionInstance" inherits="accordionPanel">
      <panel inherits="partition"/>
      <panel inherits="partitionAxis"/>
      <panel
       inherits=”http://lectra.com/pdm/search/tna#explorer_criteria/workflow”
       layout="form" >
         <datefield fieldLabelI18n="label.search.deadlineDate"
          name="deadlineDate.values"
          hiddenName="deadlineDate" width="100" chkVisible="true" />
         <textarea name="notes.values" fieldLabelI18n="label.search.notes"
          value="" chkVisible="true"/>
      </panel>
      <panel inherits=”http://lectra.com/pdm/search/tna#explorer_criteria/task”
       layout="form" >
         <textfield fieldLabelI18n="label.search.taskType" name="taskType.values"
          hiddenName="taskType" chkVisible="true" />
         <numberfield fieldLabelI18n="label.search.taskTimeSpent"
          name="taskTimeSpent.values" hiddenName="taskTimeSpent"
          chkVisible="true"/>
         <numberfield fieldLabelI18n="label.search.lcsTimeSpent"
          name="lcsTimeSpent.values" hiddenName="lcsTimeSpent"
          chkVisible="true" />
      </panel>
   </panel>
</panel>
```


Lectra Fashion PLM Product Developer
SUBSETS SEARCH
Search Criteria
Reset
Search
PRODUCT DESCRIPTION
PRODUCT CLASSIFICATION
SUBSETS
ATTRIBUTES
TIME & ACTIONS
Lifecycle State
Scheduled Start
Scheduled Finish
Overdue Only
Process
#label.search.deadlineDate
05/15/2014
#label.search.notes
Notes
STEP
Search Result
TIME & ACTIONS / #label.search.deadlineDate : 05/15/2014 #label.search.notes : Notes
Print all search result
Export all to Excel
Colorway Blue
05/15/2014
Notes for this planning
Validated on
05/16/2014
To Do before
05/17/2014
+10 DAYS
2 DAYS
K. Gallagher
Displaying 1 - 1 of 1


Lectra Fashion PLM Product Developer
SUBSETS SEARCH
Search Criteria
Reset
Search
My Criteria Set
Category
Configuration
Colorways with deadline
TIME & ACTIONS
#label.search.deadlineDate
05/15/2014
#label.search.notes
Notes
STEP
Name
Resource(s)
Scheduled Finish
Forecast Finish
Task
Lifecycle State
Overdue Only
#label.search.taskType
Design
#label.search.taskTimeSpent
20
#label.search.lcsTimeSpent
18
Search
Selection (0)
Search Result
TIME & ACTIONS / #label.search.deadlineDate : 05/15/2014
#label.search.notes : Notes
STEP / #label.search.taskType : Design
#label.search.taskTimeSpent : 20
#label.search.lcsTimeSpent : 18
Print all search result
Export all to Excel
Display
My Views
Create/Select View
Initial Lifecycle State
Prototypage
Name
Summary
Colorway Blue
05/15/2014
Notes for this planning
Validated on
05/16/2014
To Do before
05/17/2014
+10 DAYS
2 DAYS
K. Gallagher
Page 1 of 1
Elements per Page 25
Displaying 1 - 1 of 1

6. SUBSETS

6.1 Subset instance

6.1.1.1 How to add two custom fields "Critical State" (Boolean) and "Comments" (String) to "Deliveries" configuration?

PLM Manager

1. Go to Data Configuration - Extend Data Model
2. Extend Subset item
3. Right click on Deliveries - Add custom field
4. Add field named criticalState as Boolean field
5. Right click on Deliveries - Add custom field
6. Add field named comment as String field

Screen custo

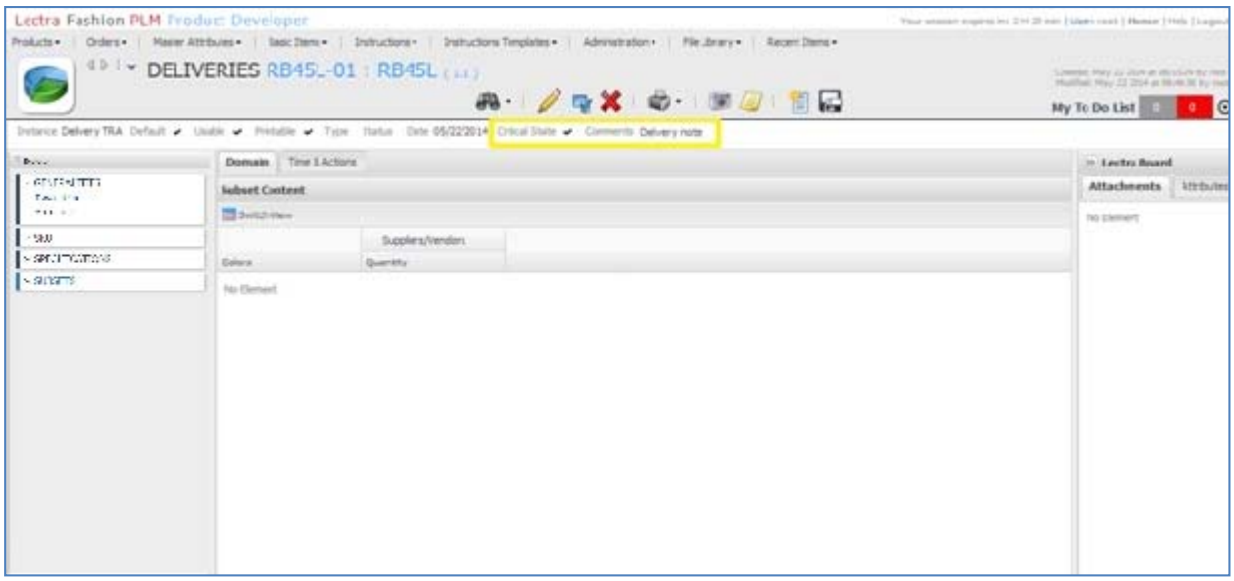
Edit the file screensCusto.xml (LectraPLMParam folder on the server) and declare your assortment by adding the following XML code.

```
<screen name="ProductPartition" displayCustoms="false">
   <icons>
      <icon name="defaultImageField"/>
   </icons>
   <!--         Composition of title of header -->
   <block name="header" displayed="false" columns="4" template="header">
      <field name="product.codeAlpha1" editable="false" alias="code1"/>
      <field name="product.codeAlpha2" editable="false" alias="code2"/>
      <field name="product.version.number" editable="false" alias="version"/>
   </block>

   <!--         Header navigation definition -->
   <block name="navheader" displayed="false"
    template="navheaders/NavHeader.html">
      <block name="navheaderCommon" displayed="false"
       template="navheaders/NavFields.html">
         <field name="defaultInstance" alias="defaultInstance" editable="false"/>
         <field name="usable" alias="usable"/>
         <field name="tobePrinted" alias="printable" editable="true"/>
         <field name="partitionType" alias="partitionType" editable="true"/>
         <field name="partitionStatus" alias="partitionStatus" editable="true"/>
         <field name="statusDate" alias="statusDate" editable="false" />
      </block>
      <block name="navheaderExtra" displayed="true"
       template="navheaders/NavFields.html">
         <field name="criticalState" alias="criticalState" editable="true" />
         <field name="comment" alias="comment" editable="true" />
      </block>
   </block>

   <!--         Storyboard -->
   <block name="storyBoard" displayed="false"
    template="screens/Block_Content.html">
   <tabs name="multiMediaBoardAndAxisTabs" tabPosition="top">
      <tab i18nKey="action.toggle.storyBoard" name="multiMediaBoardTab"
      displayed="true" icon="attachement.png">
```

```
        <block name="multiMediaBoardContainer" template="board/Board.html">
        <!-- here will come the storyboard block define in the root screen -->
        </block>
    </tab>
    <tab i18nKey="ProductPartition.axis" name="axisTab" displayed="true"
     autoLoad="true" autoHeight="true" layout="form" padding="5"
     labelAlign="top" border="false">
        <block name="axisBoardContainer" template="axisBoard/AxisBoard.html"
        displayed="true">
            <field name="colors" />
            <field name="sizes" />
            <field name="suppliers" />
            <field name="channels" />
            <field name="businessCategorys" />
            <field name="seasons" />
            <field name="themes" />
            <field name="brands" />
            <field name="divisions" />
            <field name="options" />
            <field name="genders" />
        </block>
    </tab>
</tabs>
<block>
<block name="multiMediaBoard" displayed="false" columns="1"
 template="storyboard">
    <field name="fileLink" editable="false" alias="fileLink"/>
    <field name="comment" editable="false" alias="comment"/>
</block>

<!-- tabs -->
<tabs id="partitionTabs" name="partitionTabs" displayed="true"
tabManager="com.lectra.pdm.webapp.tabs.ProductPartitionTabManager"
resizerCt="centerPanel">
<!-- Domain tab  -->
    <tab name="domain" i18nKey="ProductPartition.domain" icon="icons/table.png"
    autoHeight="true" autoLoad="true">
        <block name="content">
            <table name="partitionRow"
              customizable="true"
              sortable="true"
              rowCheckable="false"
              title="PartitionContent"
              xtype="Lectra.PDM.Partition.MatrixGrid"
              collapsible="false"
              resizable="false"
              resizerCt="tabPanel.ProductPartition.partitionTabs"
              maxHeightForFixedCols="600">
                <column name="defaultValue" editable="true" showable="true"
                displayed="true" sortable="true" />
            </table>
        </block>
    </tab>
    <!-- Workflow tab  -->
    <tab name="workFlow" i18nKey="TimeAndActions"
     lpfExtPath=http://lectra.com/pdm/timeandaction#generalPanel
     extendedProfileTag="TATab">
    </tab>
    <tab name="workflowTabWarning" i18nKey="workflowTabWarning"
     directChild="true"
     displayed="false" >
        <block name="errorinfo" template="switch/workflowTabWarning.html" />
    </tab>
</tabs>
```

```
   <!-- Tracking area (bottom of the screen)  -->
   <tabs name="trackingChangeZone" displayed="false">
      <tab name="changeZone" i18nKey="ProductGO.changeZone" autoLoad="false"
       directChild="true">
      </tab>
      <tab name="actionsHistoryZone" i18nKey="ProductGO.actionsHistoryZone"
       lpfExtPath="http://lectra.com/pdm/changetracking#actionsHistoryGrid" />
   </tabs>

   <block name="supplier" displayed="false">
      <table name="supplierTasks"
       defaultColumn="default" customizable="true" maxHeightForFixedCols="200"
       thumbView="false" displayed="false" >
         <column name="company.name" editable="false" sortable="false"/>
         <column name="opened" editable="true" sortable="false"/>
         <column name="usable" editable="true" sortable="false"/>
         <column name="proposedDate" editable="true"/>
         <column name="comment" editable="true"/>
      </table>
   </block>
   <!--         This block is used as a Screen in the saveAs popup -->
   <block name="saveAs" displayed="false" columns="1"
    template="partition/PartitionSaveAsPopup.html" >
      <block name="identification" >
         <field name="name" editable="true"/>
      </block>
   </block>

</screen>
```

Screen Result



6.1 Subsets explorer

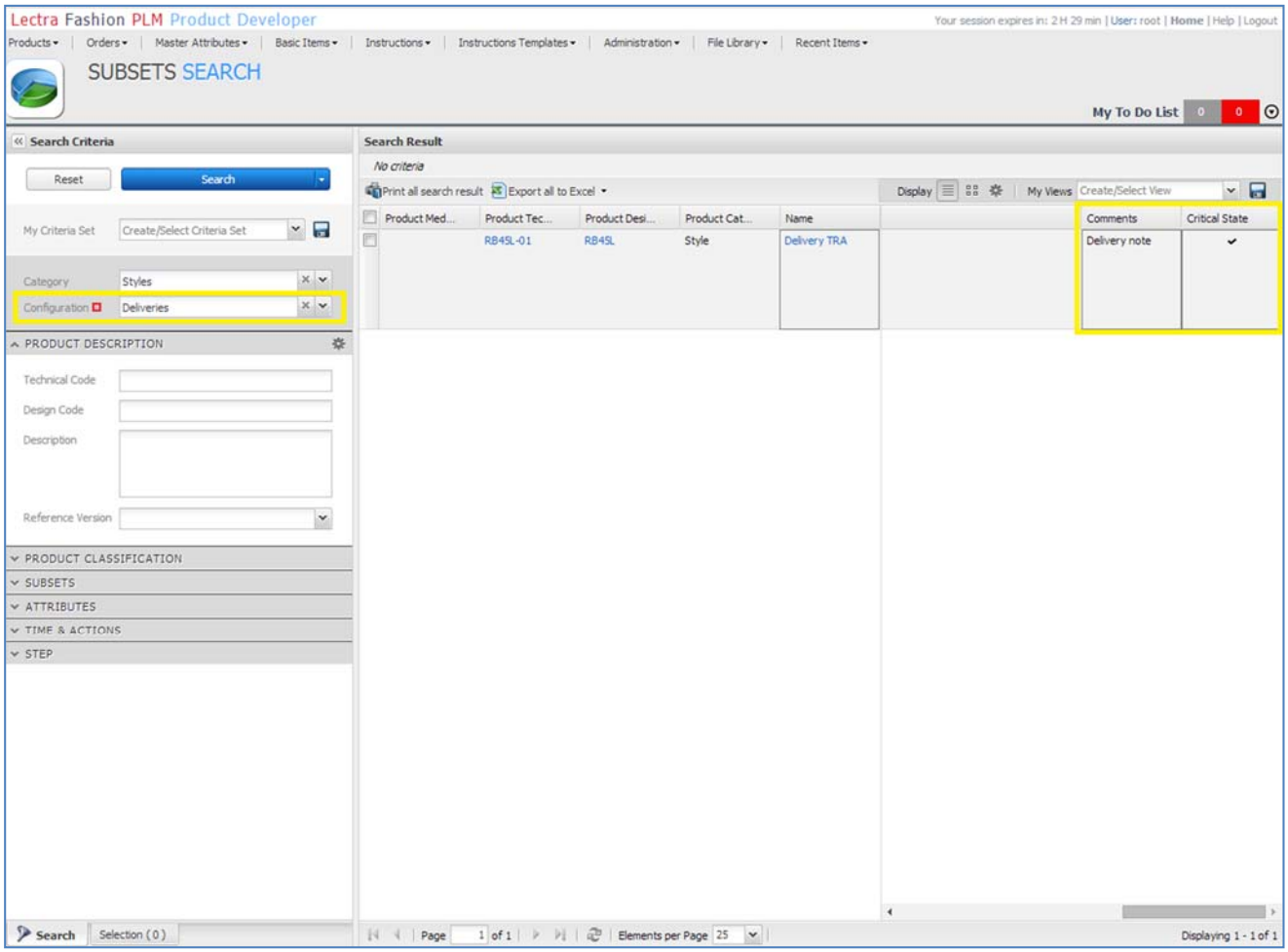
6.1.1.1 In subset explorer, with Deliveries configuration, how to see new custom fields?

Screen custo

On file **namespaces/custom/Lectra.PDM.Search.Assortment.xml**, between <*namespace*> tags, add the following XML code

```
<explorereditablegrid name="resultGrid_Deliveries"
 inherits="resultGrid_default">
   <colModel inherits="resultGrid_default/colModel">
      <column xsi:type="textcolumn" dataIndex="comment" headerI18n="comment"
       editable="true" />
      <column xsi:type="checkboxcolumn" dataIndex="criticalState"
       headerI18n="criticalState" editable="true"/>
   </colModel>
   <store xsi:type="lpfdirectstore" inherits="resultGrid_default/store">
      <dataField name="criticalState" />
      <dataField name="comment" />
   </store>
</explorereditablegrid>
```

Screen Result



6.1.1.2 How to create the associated search criteria?

PLM Manager

1. Create a new .xls file, and paste (in A1 cell) the table below

{core.qualification.Se	{core.qualific	{core.qualifica	{core.qualification.Se	{cor	{core.qualifi	{core.qualifi	{core.qual
Entity Type	**Entity Configu**	**Field Name**	**Field Path**	**Fiel**	**Field Type**	**Field Type C**	**Field Type**
productmanagement	**Deliveries**	comment	customFields.comment		core.customdefinitions.StringDef		
productmanagement	**Deliveries**	criticalState	customFields.criticalState		core.customdefinitions.BooleanDef		

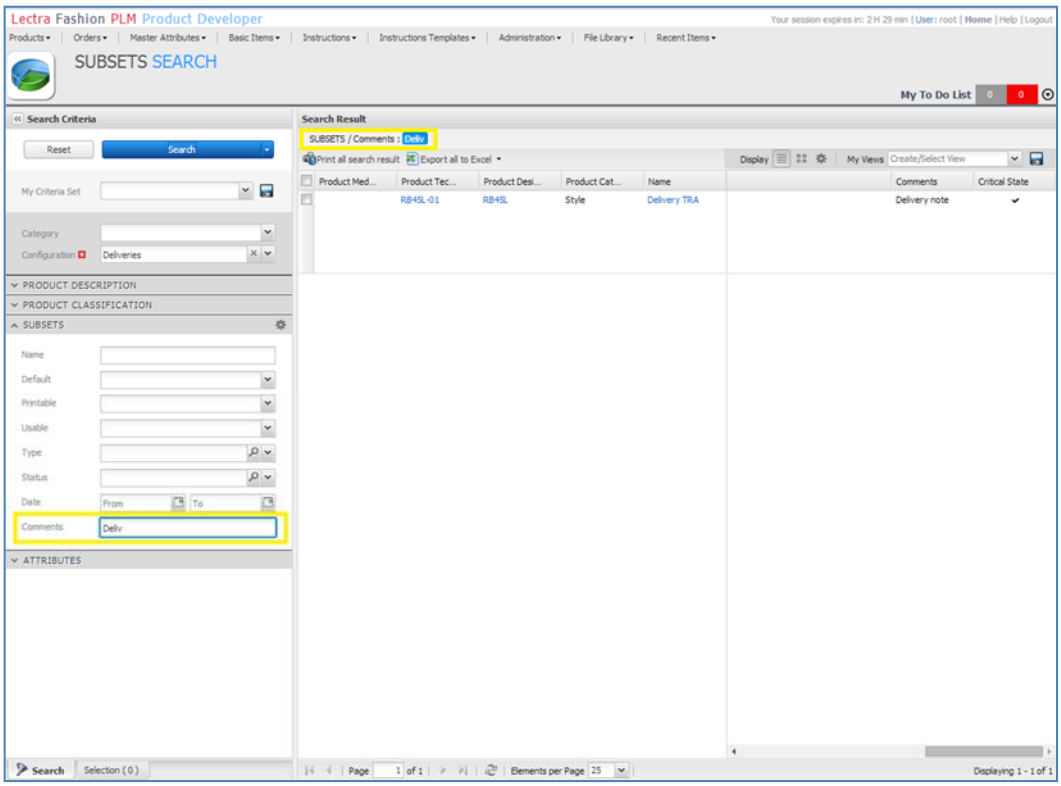
2. In PLM Manager, select Data Configuration / Define Search Criteria
3. Select Partition_**Deliveries** and click on Import
4. Search and import .xls created file

Screen custo

On file **namespaces/custom/Lectra.PDM.Search.Assortment.xml**, between *<namespace>* tags, add the following XML code to add comment field in "partition" criteria panel:

```
<panel name="criteria_Deliveries" >
   <panel region="north" name="ConfigInstance" inherits="Configuration" />
   <panel region="center" name="accordionInstance" inherits="accordionPanel" >
      <panel inherits="partition" >
         <textfield fieldLabelI18n="comment" name="comment.values"
          hiddenName="comment" chkVisible="true"/>
      </panel>
      <panel inherits="partitionAxis"/>
   </panel>
</panel>
```

Screen Result

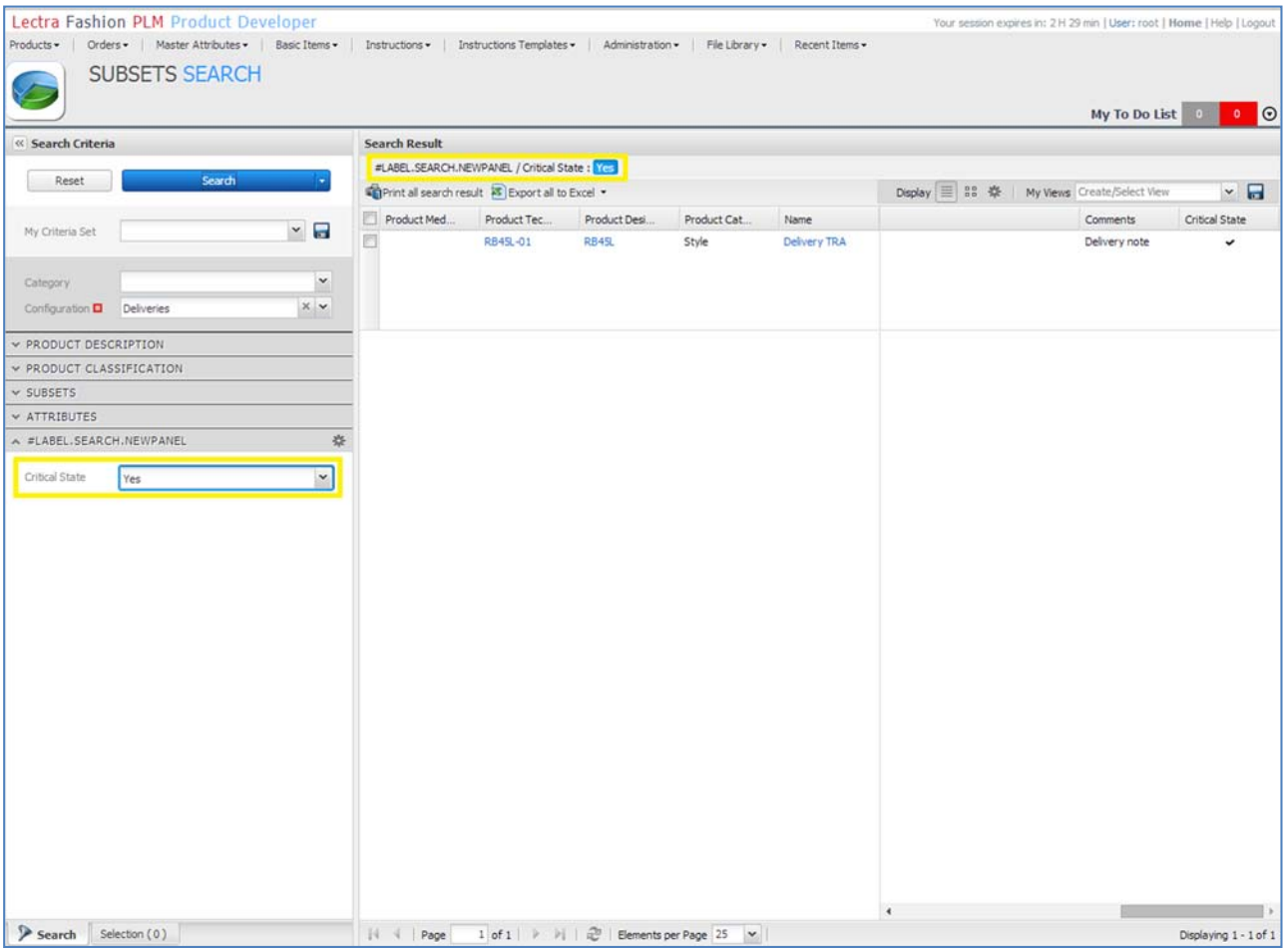


Screen custo

On file **namespaces/custom/Lectra.PDM.Search.Assortment.xml**, to create a new search criteria panel and add criticalState criteria, update XML with adding a new panel:

```
<panel name="criteria_Deliveries" >
   <panel region="north" name="ConfigInstance" inherits="Configuration" />
   <panel region="center" name="accordionInstance" inherits="accordionPanel" >
      <panel inherits="partition" >
         <textfield fieldLabelI18n="comment" name="comment.values"
         hiddenName="comment" chkVisible="true"/>
      </panel>
      <panel inherits="partitionAxis"/>
      <panel titleI18n="label.criteria.newPanel" collapsible="true"
      layout="form">
         <defaults anchor="100%" />
         <twincombo fieldLabelI18n="criticalState"
         hiddenName="criticalState.values"
         inherits="http://lectra.com/pdm/common#yesno"
         conditionCtrl="com.lectra.lpf.uimodel.FeatureFilter"
         chkVisible="true"/>
      </panel>
   </panel>
</panel>
```

Screen Result



7. COLLECTION PLAN

7.1 Detail Collection Plan screen

7.1.1 How customize General Data in detail Collection Plan screen ?

Let's add custom fields and custom roles in General Data :

PLM Manager Steps

1. Go to Data Configuration - Extend Data Model
2. Extend Collection Management item
3. Right click on Collection Plan - Add **custom field** / Add **custom role**
4. Add fields / roles named:
    a. **CP_LongString** as Long string field
    b. **CP_Int** as Integer field
    c. **CP_Date** as Date field
    d. **CP_Boolean** as Boolean field
    e. **CP_Country_Multi** as Country process Pick List allowing multiple
    f. **CP_Country_Single** as Country process Pick List
    g. **CP_Doc** as Document Target
    h. **CP_Color** as Marketing Color

Screen custo

On file **namespaces/custom/Lectra.PDM.CollectionPlan.CollectionPlan.xml**, between *<panel inherits="detailsEdit/detailsEditCenter" overflowX="auto" overflowY="auto" layout="form">* tags, add the following XML code:

```
<textarea name="CP_LongString" fieldLabelI18n="custom.cp.label.longstring" height="100" width="500" margin="10 0 0 10">
        <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_LongString}" />
</textarea>
<checkbox name="CP_Boolean" fieldLabelI18n="custom.cp.label.boolean" margin="10 0 0 10">
        <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Boolean}" />
</checkbox>
<integerfield name="CP_Int" fieldLabelI18n="custom.cp.label.int" margin="10 0 0 10">
        <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Int}" />
</integerfield>
<datefield name="CP_Date" fieldLabelI18n="custom.cp.label.date" margin="10 0 0 10" width="250">
        <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Date}" />
</datefield>
<nodecombo name="CP_Country_Multi" multiSelect="true" propertyName="Country" formatQuery="false" fieldLabelI18n="custom.cp.label.hvl.country.multi" width="300" margin="10 0 0
10">
        <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Country_Multi}" />
</nodecombo>
<nodecombo name="CP_Country_Single" multiSelect="false" displayField="path" propertyName="Country" formatQuery="false" fieldLabelI18n="custom.cp.label.hvl.country.single"
width="300" margin="10 0 0 10">
        <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Country_Single}" />
</nodecombo>
<imagefield name="CP_Doc" fieldLabelI18n="custom.cp.label.document" useREST="true"
                            inherits="http://lectra.com/pdm/common#defaultImageFieldEXT5" managerRef="selSmallImageField" imageWidth="96" imageHeight="96">
        <managerInstance name="Lectra.PDM.Mgrs.imageFieldRestMgr" />
<bind readOnly="{!isPageEditable}" fileLinkObj="{collectionPlan.CP_Doc}" />
</imagefield>
<envitemcombo name="CP_Color" hiddenName="CP_Color" envItemType="MarketingColor" hideTrigger2="true" fieldLabelI18n="custom.cp.label.envitem.color" pageSize="25" >
        <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Color}" />
</envitemcombo>
```

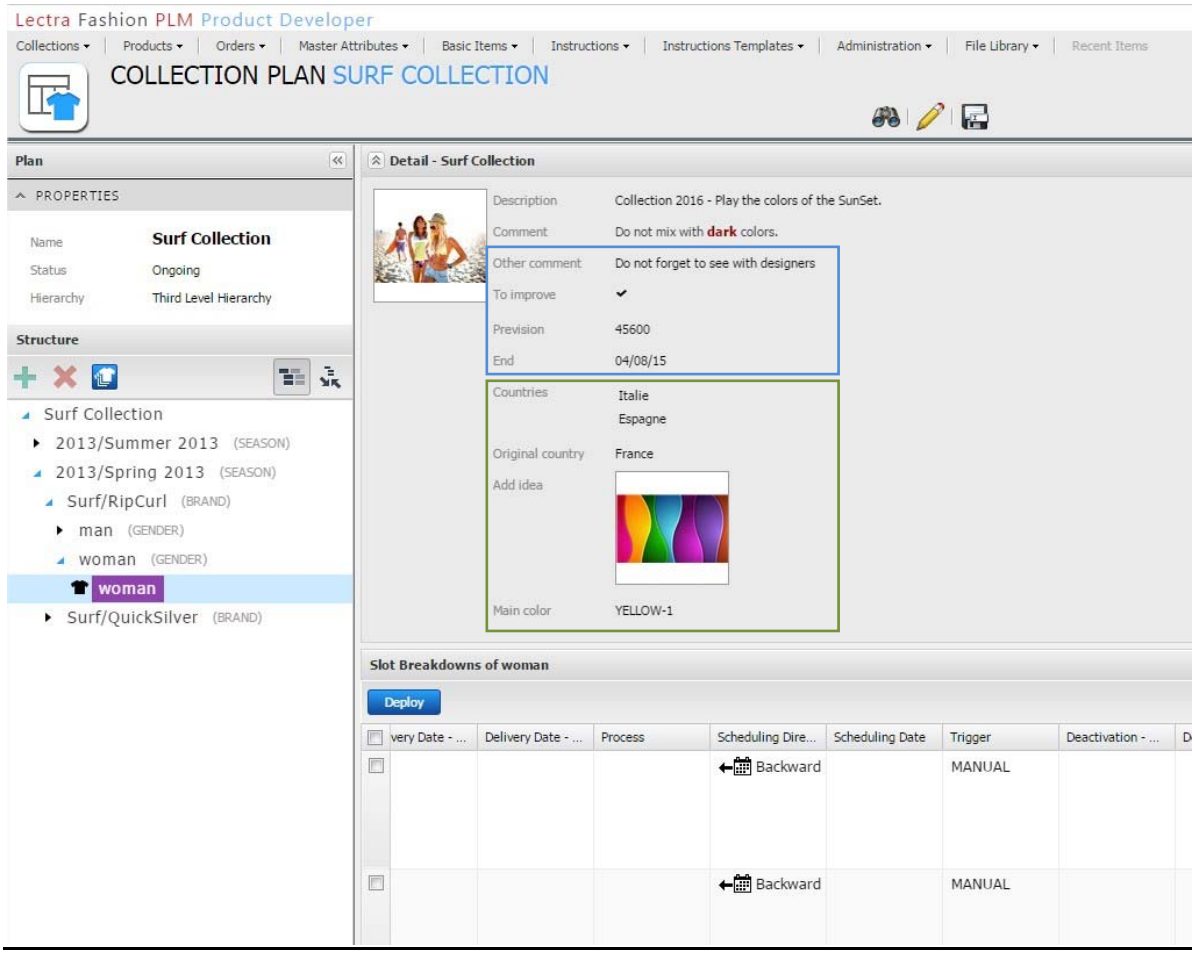
Translations

Add i18 keys and their value by modifying the **PLM-Fashion\PDM\LectraPLMParam\i18n\xls\custom\messagesCusto_en.properties** file:

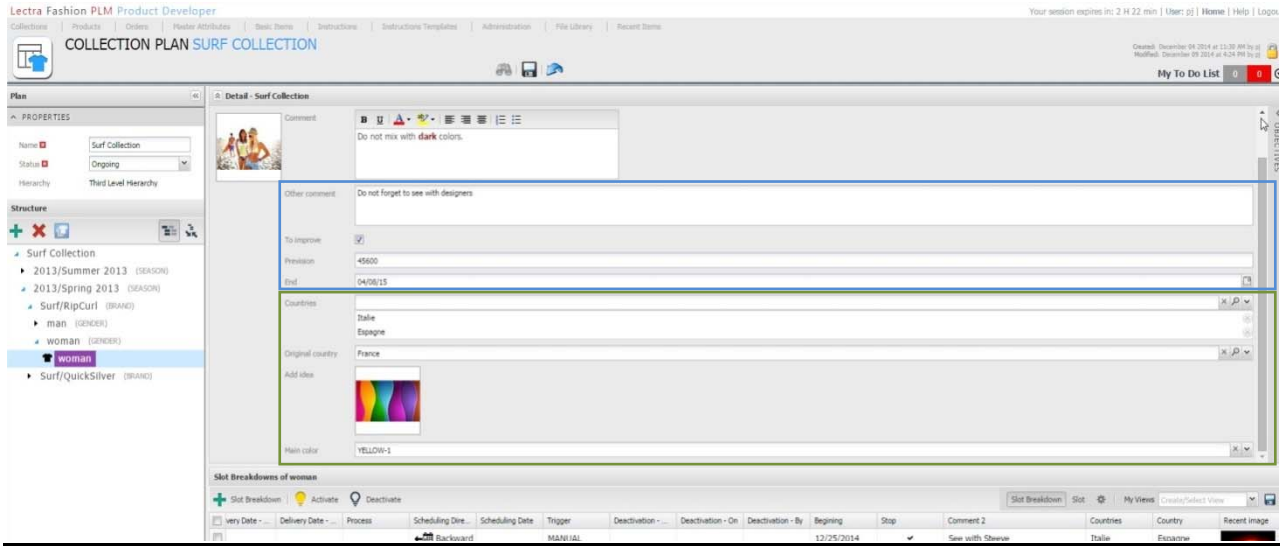
```
custom.cp.label.longstring=Other comment
custom.cp.label.boolean=To improve
custom.cp.label.int=Prevision
custom.cp.label.date= End
custom.cp.label.hvl.country.multi= Countries
custom.cp.label.hvl.country.single=Original country
custom.cp.label.document=Add idea
custom.cp.label.envitem.color=Main color
custom.cp.rp.label.date= Begining
custom.cp.rp.label.boolean= Stop
custom.cp.rp.label.string=Comment 2
custom.cp.rp.label.hvl.country.multi= Countries
custom.cp.rp.label.hvl.country.single= Country
custom.cp.rp.label.document=Recent image
```

Do the same for other language if needed (example : messagesCusto_**fr**.properties etc.)

Screen Result - display mode



Screen Result - edit mode



7.1.2 How customize Slot Breakdown Grid in detail Collection Plan screen?

Let's add custom fields and custom roles in Slot Breakdown Grid :

PLM Manager Steps

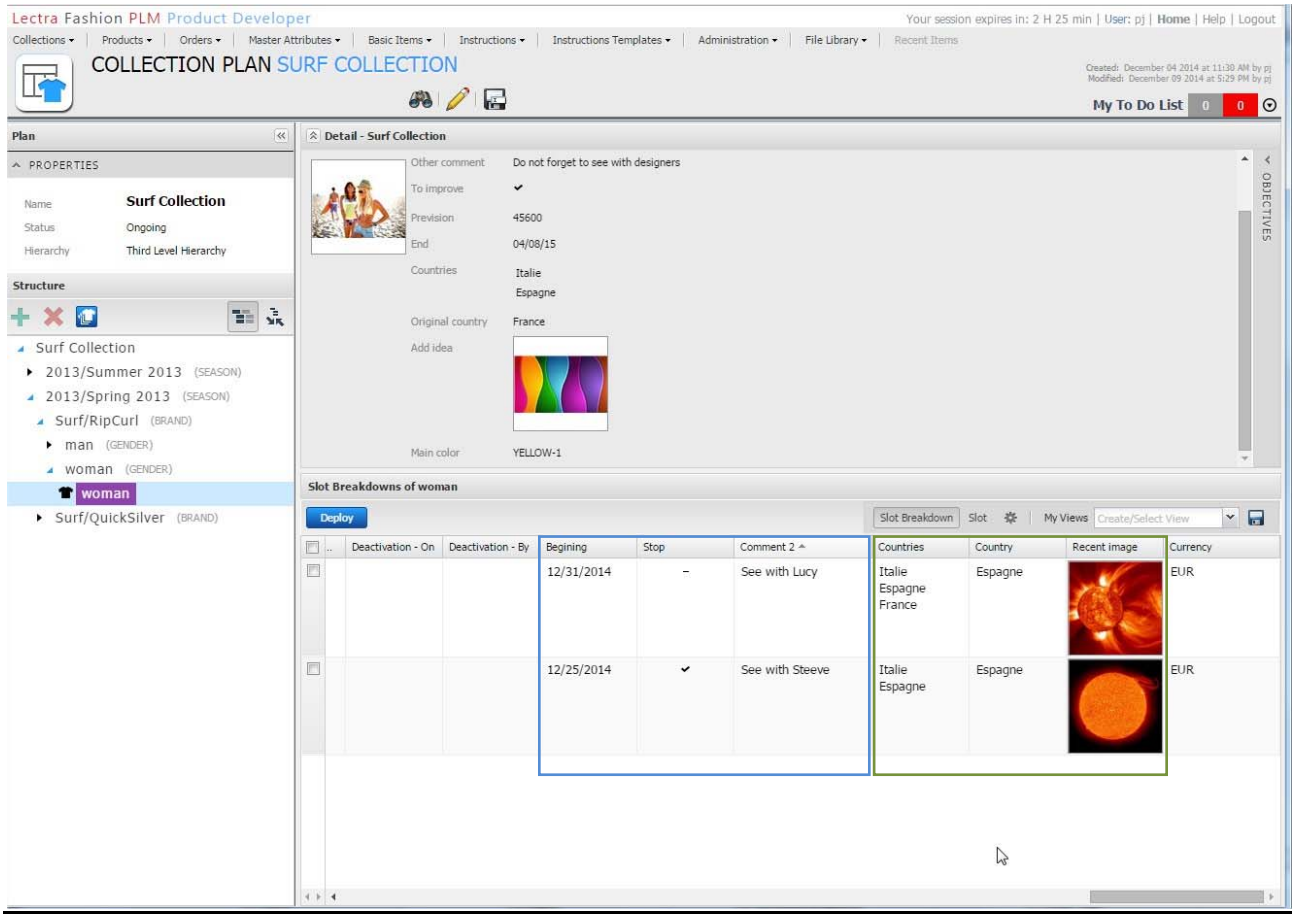
1. Go to Data Configuration - Extend Data Model
2. Extend Collection Management item
3. Right click on Slot Breakdown - Add **custom field** / Add **custom role**
4. Add fields / roles named:
    a. **CP_RP_String** as String field
    b. **CP_RP _Date** as Date field
    c. **CP_RP _Boolean** as Boolean field
    d. **CP_RP_Country_Multi** as Country process Pick List allowing multiple
    e. **CP_RP _Country_Single** as Country process Pick List
    f. **CP_RP _Doc** as Document Target

Screen custo

On file **namespaces/custom/Lectra.PDM.CollectionPlan.CollectionPlan.xml**, between *<grid name="rangeplanslotbreakdowngridDefault" inherits="rangeplanslotbreakdowngrid">* tags, add the following XML code :

```
<column xtype="datecolumn" dataIndex="CP_RP_Date" headerI18n="custom.cp.rp.label.date" format="${#i18n['i18n.extdateformat']}" editable="true">
        <editorField xsi:type="datefield" format="${#i18n['i18n.extdateformat']}" />
</column>
<column xtype="checkboxcolumn" dataIndex="CP_RP_Boolean" headerI18n="custom.cp.rp.label.boolean" headerId="CP_RP_Boolean" editable="true"/>
<column xtype="textcolumn" dataIndex="CP_RP_String" headerI18n="custom.cp.rp.label.string" headerId="CP_RP_String" editable="true"/>
<column dataIndex="CP_RP_Country_Multi" headerI18n="custom.cp.rp.label.hvl.country.multi" headerId="CP_RP_Country_Multi">
        <editorField xsi:type="nodecombo" multiSelect="true" propertyName="Country" formatQuery="false" />
        <rendererFn name="Lectra.PDM.Format.nodesRenderer()" />
</column>
<column dataIndex="CP_RP_Country_Single" headerI18n="custom.cp.rp.label.hvl.country.single" headerId="CP_RP_Country_Single" >
        <editorField xsi:type="nodecombo" multiSelect="false" propertyName="Country" formatQuery="false" />
        <rendererFn name="Lectra.PDM.Format.nodeRenderer()" />
</column>
<column xsi:type="imagecolumn" dataIndex="CP_RP_Doc" headerI18n="custom.cp.rp.label.document" headerId="CP_RP_Doc" imageSize="medium" useREST="true" editable="true" >
        <imageEditor inherits="http://lectra.com/pdm/common#defaultImageFieldEXT5" managerRef="attachmentImageField">
                <managerInstance name="Lectra.PDM.Mgrs.imageFieldRestMgr" />
        </imageEditor>
</column>
```

Screen Result



7.2 List Collection Plan screen

7.2.1 How customize General Data in detail Collection Plan screen ?

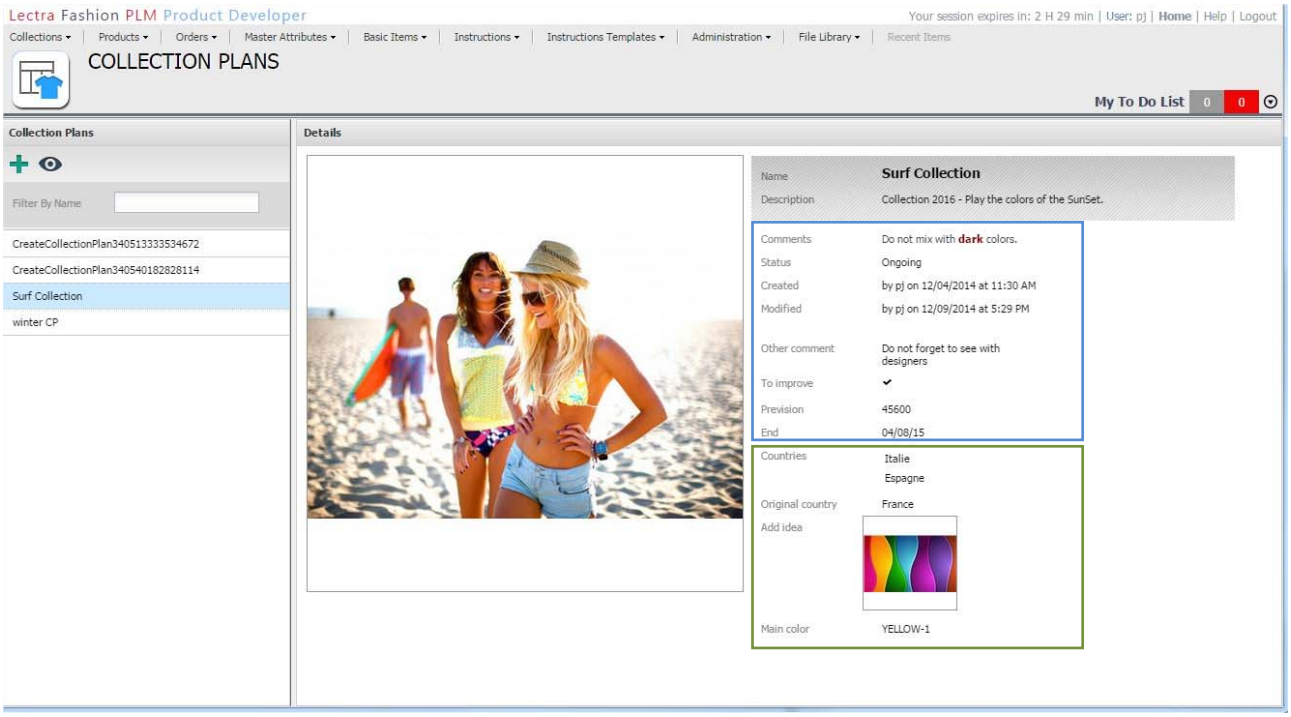
Let's add custom fields and custom roles in General Data :

PLM Manager Steps

See previous PLM Manager steps.

Screen custo

On file **namespaces/custom/Lectra.PDM.CollectionPlan.ListCollectionPlan.xml**, between *<fieldset inherits="listCollectionDetails/listCollectionDetailsRight"* > tags, add the following XML code :

```
<fieldset padding="10" border="false" margin="-3 0 0 0">
        <textarea name="CP_LongString" fieldLabelI18n="custom.cp.label.longstring" readOnly="true" />
        <checkbox name="CP_Boolean" fieldLabelI18n="custom.cp.label.boolean" readOnly="true" />
        <integerfield name="CP_Int" fieldLabelI18n="custom.cp.label.int" readOnly="true" />
        <datefield name="CP_Date" fieldLabelI18n="custom.cp.label.date" readOnly="true" />
        <nodecombo name="CP_Country_Multi" multiSelect="true" propertyName="Country" fieldLabelI18n="custom.cp.label.hvl.country.multi" >
               <bind readOnly="true" />
        </nodecombo>
        <nodecombo name="CP_Country_Single" multiSelect="false" propertyName="Country" fieldLabelI18n="custom.cp.label.hvl.country.single" >
                <bind readOnly="true" />
        </nodecombo>
        <imagefield name="CP_Doc" fieldLabelI18n="custom.cp.label.document" useREST="true"
                inherits="http://lectra.com/pdm/common#defaultImageFieldEXT5" managerRef="selSmallImageField" imageWidth="96" imageHeight="96" readOnly="true" >
                <managerInstance name="Lectra.PDM.Mgrs.imageFieldRestMgr" />
        </imagefield>
        <envitemcombo name="CP_Color" envItemType="MarketingColor" fieldLabelI18n="custom.cp.label.envitem.color" hideTrigger2="true" readOnly="true" />
</fieldset>
```

Screen Result