



LECTRA FASHION PLM PRODUCT DEVELOPER CONFIGURATION DES ECRANS

Guide de configuration

Date de la dernière mise à jour : Janvier 2017

Contenu

Conventions	4#
Exception	4#
Introduction	4#
1.# Nouvelle structure de configuration	5#
1.1# Structure précédente	5#
1.2# Nouvelle structure	5#
1.3# Répertoire Config	6#
1.4# Répertoire i18n	6#
1.5# Répertoire lpf	6#
1.6# Répertoire lpfext	7#
1.7# Répertoire Webtool	7#
1.8# Opération	7#
2.# Définitions et concepts	8#
2.1# Fichiers XML	8#
2.2# Héritage	11#
2.3# Ce que vous devez savoir sur la structure de description des écrans	14#
3.# Configuration par l'ajout de champs à « General Objectives »	16#
3.1# Ajout de champs simples	16#
3.2# Ajout d'une liste	21#
3.3# Ajout de la classification	23#
3.4# Ajout d'un panneau « Properties (Propriétés) »	26#
4.# Configuration en ajoutant des champs dans l'explorateur Produits	29#
4.1# Prérequis du module Administration et configuration	29#
4.2# Présentation de l'écran Product (Produit)	29#
4.3# Fichier XML personnalisable	30#
4.4# Structure du panneau de recherche dans le fichier XML	30#
4.5# Identification de la partie à dupliquer et modification	31#
4.6# Ajout d'un critère à un groupe existant du panneau de recherche	32#
4.7# Ajout d'un groupe de critères dans le panneau de recherche	32#
4.8# Ajout d'une colonne à la grille des résultats	34#
5.# Configuration en ajoutant des champs au Plan de collection	38#
5.1# Prérequis du module Administration et configuration	38#
5.2# Présentation des écrans du plan de collection	38#
6.# Configuration avancée pour l'ajout des champs et des panneaux	62#
6.1# Utilisation de l'héritage	62#

6.2# Exemple spécifique	62#
7.# Modification de la mise en page	63#
7.1# Définition de la mise en page	63#
7.2# Seule la structure ExtJS «containers» possède une « layout (mise en page) »	63#
ANNEXES	65#
#	



Les modifications apportées au document depuis sa dernière publication sont surlignées en [bleu](#).

CONVENTIONS

Module de développement produit = Product Developer

Module d'administration et de configuration = PLM Manager

EXCEPTION

Dans Lectra Fashion PLM V5R1, certains écrans ne sont pas personnalisables :

- Explorateur de jeu de composition
- Forme du jeu de composition
- Onglet Composition dans Objectifs

INTRODUCTION

La technologie utilisée pour la description des écrans a évolué entre les versions V4R1 et V5R1 de Lectra Fashion PLM.

Les fichiers de description des écrans sont toujours au format XML. La structure des fichiers a cependant changé, ainsi que le contenu des écrans.

Ce document a pour objectif de vous aider à identifier les fichiers XML à modifier afin de configurer les écrans, ainsi que les parties à dupliquer et à changer.

Le premier chapitre présente l'évolution de la structure de configuration.

Le second chapitre présente les définitions et les concepts nécessaires pour lire et comprendre les fichiers XML. Certaines définitions liées à la technologie ExtJS sont incluses.

La troisième partie, la configuration, aborde l'ajout de champs et de blocs aux écrans **Objectifs généraux** pour les **Produits**, dans l'explorateur de **Produits** avec le nouveau panneau **Recherche**, ainsi que les écrans **Plan de collection**.

Une fois familiarisé avec cette configuration, vous pouvez utiliser le concept d'héritage pour améliorer l'efficacité de la configuration en diminuant la duplication des blocs.

La quatrième partie vous aidera à réorganiser le contenu de l'écran en modifiant la **mise en page** par défaut. La **mise en page** correspond à la définition de l'organisation des données dans un écran : par colonne, par tableau, etc.

Les annexes contiennent des rappels sur la création des données dans le module Administration et configuration, ainsi que la documentation technique relative aux éléments graphiques utilisés, avec une partie des structures ExtJS et LPFExt.

1. NOUVELLE STRUCTURE DE CONFIGURATION

Le répertoire LectraPLMParam, utilisé en mode de production pour la personnalisation de différents fichiers (xml, modèle html, menu, arbres) a été restructuré pour introduire un répertoire «personnalisé» qui contient tous les fichiers personnalisés.

1.1 Structure précédente

ext3	01/03/2016 06:45
namespaces	01/03/2016 06:45
screens	01/03/2016 06:45
template	01/03/2016 06:45
xls	01/03/2016 06:45
ApplicationConfiguration.xml	29/02/2016 21:36
javascript.properties	29/02/2016 21:36
lastUsed.xml	29/02/2016 21:36
LPFConfiguration.xml	29/02/2016 21:36
menu.dtd	29/02/2016 21:36
menu.xml	29/02/2016 21:38
PDMPreferences.xml	29/02/2016 21:36
rootScreens.xsd	29/02/2016 21:36
saveAsDependencyRules.xml	29/02/2016 21:36
screens.xml	29/02/2016 21:36
screensCusto.xml	29/02/2016 21:36
SpecPackageTreeView.xml	29/02/2016 21:36
TemplateSwitch.xml	29/02/2016 21:36
webclient.config	29/02/2016 21:36

1.2 Nouvelle structure

config
i18n
lpf
lpfext
webtool

{5} produits ont été créés Chacun contient un type de fichier (selon les technologies ou l'utilisation).

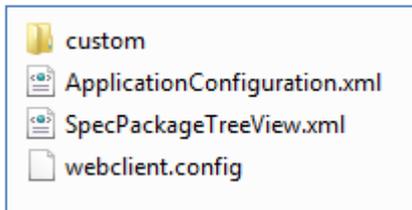
Certains fichiers, qui ne sont plus personnalisés, ou ne l'ont pas été durant une certaine période, ont également été supprimés.



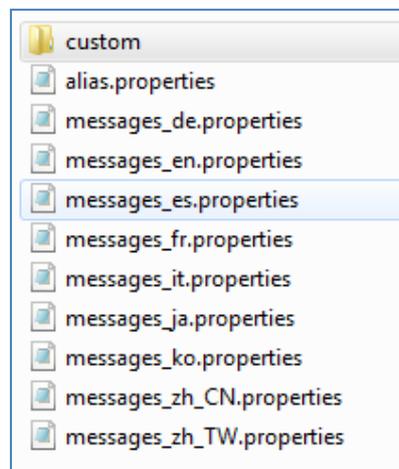
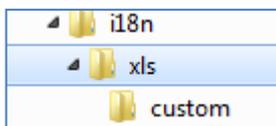
Chacun de ces répertoires contient un répertoire «personnalisé» et ceci est le seul endroit où les

fichiers personnalisés doivent être ajoutés.

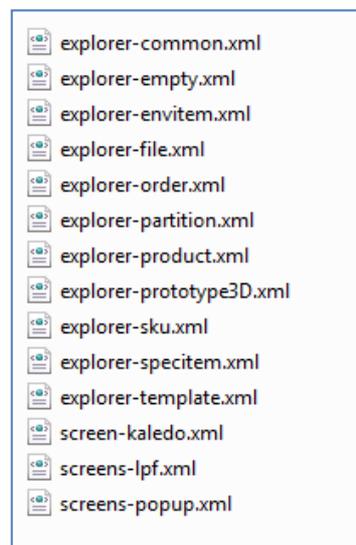
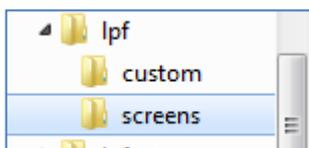
1.3 Répertoire Config



1.4 Répertoire i18n



1.5 Répertoire lpf



L'héritage n'existe pas dans cette technologie, le fichier à personnaliser doit être complètement copié dans le répertoire personnalisé, puis modifié en fonction de la configuration requise.

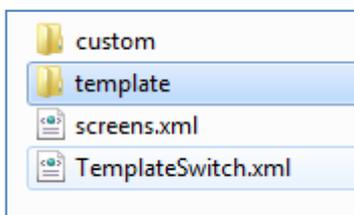
1.6 Répertoire Ipfext

Il contient la structure standard pour les fichiers de namespace (xml).

Il n'y a pas changements majeurs dans cette technologie.

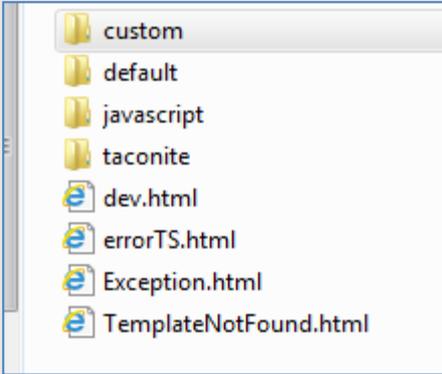
1.7 Répertoire Webtool

Il contient les deux fichiers Screens.xml et Template Switch.xml et le répertoire de modèle.



Le répertoire de modèle contient tous les modèles HTML Product Developer et a son propre répertoire de modèles.

 C'est le seul répertoire spécifié pour la personnalisation ou l'écrasement des modèles existants:



Tous les modèles HTML personnalisés pour les clients doivent être contenus à plat dans le même répertoire 'personnalisé', indépendamment de la hiérarchie initiale dans la norme. Le Product Developer récupère les modèles HTML dans le répertoire «personnalisé» (s'ils sont présents) prioritairement.

1.8 Opération

1.8.1 Opération précédente

Sur l'installation de PLM et avec le démarrage de JBoss, le répertoire LectraPLMParam est créé s'il n'existe pas et son contenu est copié à partir du WebClient WAR.

Chaque fois que JBoss est démarré, la présence de LectraPLMParam est vérifiée. S'il n'existe pas, les ressources sont recopiées.

1.8.2 Nouvelle opération

Chaque fois que JBoss est démarré (après une nouvelle installation ou juste un Stop / Start), le contenu du LectraPLMParam sera remplacé par de nouvelles ressources contenues dans le WebClient WAR à l'exception des répertoires personnalisés qui ne seront pas modifiés.



Toutes personnalisations faites dans les fichiers d'origine seront perdues!

2. DEFINITIONS ET CONCEPTS

2.1 Fichiers XML

La définition des écrans peut être réalisée dans plusieurs fichiers XML de description. Contrairement aux versions précédentes, à compter de la version V3R2, un fichier XML est créé pour un écran et même pour une partie d'écran.

Lorsque vous observez le code source d'un module de développement produit, vous pouvez voir que la variable « **screenName** » (nom de l'écran) peut être entrée dans l'en-tête du code source affiché : sa valeur indique le nom de l'écran.

2.1.1 Fichier screens.xml

Le fichier d'entrée à prendre en compte pour connaître le fichier XML à modifier pour personnaliser un écran est le suivant : `PLM-Fashion\PDM\LectraPLMParam\webtool\screens.xml`. Ce fichier fait référence à tous les écrans définis avec la technologie précédente, ainsi qu'avec la nouvelle.

Exemple :

Pour l'écran ProductGO, nous avons la description du nouvel onglet General Objectives, ainsi que sa partie Description et la description des autres onglets qui ne figurent pas encore dans LPFExt.

```
<!-- Generic screen for Products General Objectives. -->
<screen name="ProductGO" displayCustoms="false">
  <icons>
    <icon name="defaultImageField" />
  </icons>

  <!-- Tabs -->
  <tabs name="pgoTabs" displayed="false" placement="top"
  prefKey="ProductGO.objectives.selectedTab"
  tabManager="com.lectra.pdm.webapp.tabs.ProductTabManager"
  tabSelectionStrategy="com.lectra.pdm.webtool.config.tab.ProductGOTabStrategy">

  <!-- Objectives tab -->
  <tab name="descriptionTab" i18nKey="ProductGO.description.Title"
  autoScroll="true" autoHeight="false"
  lpfExtPath="http://lectra.com/pdm/productgo#description" />

  <!-- Cost and Margin tab -->
  <tab name="costTab" i18nKey="ProductGO.description.cost" autoLoad="true"
  autoScroll="true" autoHeight="false">
```

```
<block name="detailcost" displayed="true" columns="4" template="Content"
extendedProfileTag="PrivateArea" titled="true">
  <block name="detailcostCol2" titled="true"
template="screens\Block_3_hr.html">
    <field name="targetCost.defaultPrice" editable="true"/>
    <field name="targetCost.defaultCollectionPrice" editable="true"/>
    <field name="targetCost.approved" editable="true"/>
  </block>
</...>
```

2.1.2 lpfExtPath

Ces informations sont nécessaires dans le fichier *PLM-Fashion\PDM\LectraPLMParam\webtool\screens.xml* pour chaque écran LPFExt et affiche le chemin du fichier XML correspondant. Tous les fichiers XML se trouvent dans le dossier Namespaces.

Exemple :

```
<!-- Objectives tab -->
<tab name="descriptionTab" i18nKey="ProductGO.description.Title"
autoScroll="true" autoHeight="false"
lpfExtPath="http://lectra.com/pdm/productgo#description" />
```

La description du contenu de l'onglet **descriptionTab** se trouve dans le fichier dont le **prefix** est **productgo**, et son nom sera **Lectra.PDM.ProductGO**.

Le contenu de l'onglet sera décrit dans le bloc **description** à l'intérieur du fichier XML.



Ce bloc de description correspond à un container de type panneau. Veuillez consulter 2.1.4 - [Description de l'écran : approche technique](#).

2.1.3 Namespaces

Un fichier XML de description est structuré comme suit :

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

  <!-- comment -->
  <element1 name="elt1Name"> ... </element1>
  <!-- comment -->
  <element2 name="elt2Name"> ... </element2>

  <!-- Private field(s) : do not edit -->
  <private> ... </private>
  <!-- End of private field(s) -->

</namespace>
```

Exemple : Lectra.PDM.ProductGO.xml

Le fichier de description doit être déclaré par une balise **namespace**.

Cette balise **doit** contenir au moins les attributs suivants :

- **uri** : identifiant du fichier de description. Il comporte une partie qui change en fonction du **prefix** du fichier.
- **prefix** : identifiant qui décrit l'écran. Il est cité dans l'attribut **uri** et dans le fichier **screens.xml**. Il identifie uniquement le fichier.
- **xmlns:xsi** avec la valeur correspondante : « **http://www.w3.org/2001/XMLSchema-instance** »
- **xmlns:xs** avec la valeur correspondante : « **http://www.w3.org/2001/XMLSchema** »

L'attribut **uri** qui contient l'identifiant du fichier XML est également disponible dans le fichier **namespace.properties**.

Ce fichier contient la définition de l'identifiant en fonction de tous les fichiers de description.

Des commentaires commençant par « **<!--** » et se terminant par « **-->** » figurent dans chaque fichier de description.

Il existe également des zones privées qui ne doivent pas être modifiées par les utilisateurs. Elles débutent par le commentaire suivant :

```
<!-- Private field(s) : do not edit -->
```

et se terminent par :

```
<!-- End of private field(s) -->
```

2.1.4 Description de l'écran : approche technique

La description de l'écran débute par la définition de l'en-tête du fichier XML avec la balise **namespace** présentée dans la section [2.1.3 Namespaces](#).

L'écran est alors défini par les éléments graphiques qui le composent et qui sont décrits dans le fichier xml de base (dans le dossier namespaces), ainsi que dans le fichier xml du même nom qui se trouve dans le sous-dossier custom.

Ces éléments sont des éléments graphiques de la bibliothèque ExtJS et de la structure LPFExt. Le plus utilisé est `<panel>`, toujours défini entre `< >`.

Définitions de « `panel` » :

`<panel>` correspond à une balise d'ouverture de panneau

`</panel>` correspond à une balise de fermeture de panneau

Toutes les caractéristiques descriptives de l'élément graphique « `Panel` » sont définies dans la balise d'ouverture : disposition, couleurs, largeur, défilement...

Tous les autres éléments graphiques de ce panneau seront définis entre la balise d'ouverture et celle de fermeture. Un panneau est généralement composé d'autres éléments `<panel>`, qui sont eux-mêmes composés de `<combobox>`, `<nodecombo>`, `<textarea>`, `<textfield>`, `<richText>`, `<image>`, etc...

2.2 Héritage

Avec la nouvelle structure LPFExt, l'accès aux écrans est plus rapide. Par le biais de plusieurs fichiers de description XML, il met en place un système d'héritage qui facilite la lecture des écrans, ainsi qu'une standardisation des éléments génériques.

Contrairement à la technologie de description précédente, un mot-clé unique est désormais utilisé pour définir l'héritage : « **inherits** ». On perd désormais la notion de modification des éléments hérités.

Il existe deux types d'héritage : héritage interne et héritage externe.

2.2.1 Héritage interne

Le mode d'héritage le plus fréquemment utilisé, surtout pour décliner les descriptions d'écrans pour les différentes catégories de produits.

L'héritage interne utilise un élément déclaré à l'intérieur du fichier de description de l'écran.

```
<parent name="parentNameExample">
#   <!-- child element -->
#   <child inherits="childNameExample">
</parent>

<child name="childNameExample">
#   <!-- Title of child -->
#   <title>Titre</title>
</child>
```

La balise <parent> prend la balise <child> (définie ci-dessous) comme un élément. Dans cet exemple, la balise <child name= « childNameExample »> définit le contenu de l'élément <child>.

Ceci équivaut à écrire :

```
<parent name="parentNameExample">
#   <!-- child element -->
#   <child name="childNameExample">
#       <!-- Title of child -->
#       <title>Titre</title>
#   </child>
</parent>
```

Pour un fichier xml de description donné, il est vivement conseillé d'ajouter vos propres héritages dans le fichier xml dédié qui se trouve dans le sous-dossier namespaces/custom et possède le même nom.

Exemple :

- Le fichier de description `namespaces/Lectra.PDM.ProductGO.xml` contient la définition d'un panneau de description qui hérite de plusieurs panneaux possibles :
 - `<panel name="description_wrap" title18n="description" region="north" border="false" height="300" split="true" collapsible="true" statelid="panel_{#type}" layout="fit" maximizableChild="true">`
 - `<panel name="description" inherits="description_{#type},description_{#topCategoryName},description" />`
 - `<plugin xsi:type="xmlmap" ptype="panelCollapsedTitle" />`
 - `</panel>`
- Dans le fichier `namespaces/custom/Lectra.PDM.ProductGO.xml`, vous pouvez définir votre propre description pour une sous-catégorie personnalisée, sous le modèle suivant :

Avec « MyCustomCategory » défini dans le module Administration et configuration

```
<panel name="description_MyCustomCategory">
  <!-- Add your custom panels here -->
</panel>
```

Cette sous-catégorie étant un type, le panneau `description_MyCustomCategory` correspond à l'héritage défini par `description_{#type}`, qui est positionné en premier.

2.2.2 Héritage externe

L'héritage externe utilise un élément déclaré dans un fichier de description (voir l'exemple ci-dessous : `Common.xml`) puis l'utilise dans un autre fichier de description d'écran (voir l'exemple ci-dessous : `Ecran.xml`).

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

<child name="childNameExample">
#   <!-- Title of child -->
#   <title>Titre</title>
</child>

</namespace>
```

Lectra.PDM.Example.Common.xml

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

<parent name="parentNameExample">
  <!-- child element -->
  <child inherits="http://lectra.com/pdm/example/common#childNameExample">
</parent>
```

```
</namespace>
```

Lectra.PDM.Example.Ecran.xml

La balise <parent> prend la balise <child> comme un élément qui indique le titre de l'élément mais qui est défini dans un autre fichier de description.

La valeur dans l'attribut « **inherits** » est construite comme suit :

#{URI} du fichier de description qui contient l'élément **###{NAME}** de l'élément

Exemple :

Par exemple, l'héritage est utilisé pour les barres d'outils qui sont utilisées dans plusieurs écrans.

Namespaces/Lectra.PDM.ChangeTracking.xml utilise une barre d'outils qui hérite la barre d'outils de pagination définie dans le fichier **namespaces/Lectra.PDM.Common.xml**.

```
<paging name="pagingToolbar" displayInfo="true" pageSize="25">
  <tbseparator />
  <plugin xsi:type="pagingtoolbarresizer">
    <option>25</option>
    <option>50</option>
    <option>100</option>
  </plugin>
</paging>
```

Lectra.PDM.Common.xml

```
<bbar xsi:type="paging"
inherits="http://lectra.com/pdm/common#pagingToolbar" pageSize="7" />
```

Lectra.PDM.ChangeTracking.xml

2.2.3 Création de nouveaux fichiers

Il est possible de créer vos propres écrans en rédigeant le fichier XML approprié. Il est recommandé d'identifier ces nouveaux fichiers en les créant dans un dossier dédié : par exemple, dans un sous-dossier de **namespaces** appelé **MyCompany**.

- Créez le fichier xml conformément à la structure décrite dans la section [2.1.3 Namespaces](#).
- Faites référence au nouveau fichier dans le fichier **namespaces.properties**.

```
#Lectra namespaces
${DIRECTORY}namespaces/« MyCompany »/Lectra.PDM.Common =
http://lectra.com/pdm/common
```

- Il est recommandé de faire référence au nouveau fichier dans le fichier **screensCusto.xml** qui contiendra les déclarations .xml d'écrans, dupliquées et modifiées, comme ce qui est fait pour l'écran **ProductGO**, par exemple :

```
<screen name="ProductGO" displayCustoms="false">
<icons>
  <icon name="defaultImageField" />
</icons>

<!-- Tabs -->
<tabs name="pgoTabs" displayed="false" placement="top"
  prefKey="ProductGO.objectives.selectedTab"
  tabManager="com.lectra.pdm.webapp.tabs.ProductTabManager"
  tabSelectionStrategy="com.lectra.pdm.webtool.config.tab.ProductGOTabStrategy">
  <tab name="descriptionTab" il8nKey="ProductGO.description.Title"
    lpfExtPath="http://lectra.com/pdm/productgo#description_Style" />
</tabs>
</screen>
```

2.2.4 Propriétés générales

- L'héritage est défini par l'attribut « **inherits** » et prend la valeur de l'attribut **name** (nom) de l'élément à intégrer.
- Le type de balise (<panel> par exemple) qui appelle l'héritage (déclarant l'attribut « **inherits** ») doit être **identique** à celui de la balise déclarée.

2.3 Ce que vous devez savoir sur la structure de description des écrans

2.3.1 La recherche d'écran débute par la lecture du fichier screens.xml

L'écran recherché est identifié :

balise <screen> suivie d'un champ **name** dont la valeur correspond au **prefix** (préfixe) de l'écran.

Pour cette balise, nous identifions l'onglet qui nous intéresse, la balise <tab>, ainsi que le nom du fichier xml correspondant, et le nom du panneau correspondant dans **lpfExtPath** associé.

Exemple :

```
<!-- Generic screen for Products General Objectives. -->
<screen name="ProductGO" displayCustoms="false">
  <icons>
    <icon name="defaultImageField" />
  </icons>

  <!-- Tabs -->
  <tabs name="pgoTabs" displayed="false" placement="top"
    prefKey="ProductGO.objectives.selectedTab"
    tabManager="com.lectra.pdm.webapp.tabs.ProductTabManager"
    tabSelectionStrategy="com.lectra.pdm.webtool.config.tab.ProductGOTabStrategy">
    <!-- Objectives tab -->
    < !-- NOUVELLE TECHNO -->
```

```

<tab name="descriptionTab" i18nKey="ProductGO.description.Title"
  autoScroll="true" autoHeight="false"
  lpfExtPath="http://lectra.com/pdm/productgo#description" />

<!-- Cost and Margin tab -->
<!-- ANCIENNE TECHNO -->
<tab name="costTab" i18nKey="ProductGO.description.cost"
  autoLoad="true" autoScroll="true" autoHeight="false">
  <block name="detailcost" displayed="true" columns="4"
  template="Content" extendedProfileTag="PrivateArea"
  titled="true">
  <block name="detailcostCol2" titled="true"
  template="screens\Block_3_hr.html">

..

```

Dans la nouvelle technologie, le chemin menant au fichier xml correspondant est exprimé comme suit :

```
lpfExtPath="http://lectra.com/pdm/productgo#description"
```

```
lpfExtPath="http://lectra.com/pdm/ « prefix du fichier xml »# « nom du panel » "
```

Dans cet exemple, le premier onglet (**Objectives**) correspond au fichier **Lectra.PDM.ProducGO** qui possède le même préfixe que **ProductGO** et que le panneau **description**.

 Les blocs décrits avec la technologie antérieure dans le fichier **screens.xml** sont désormais définis par **panels** dans le fichier xml dédié.

2.3.2 Lecture du fichier xml de base

Dans l'exemple donné, la déclaration du panneau description est recherchée dans le fichier Lectra.PDM.ProductGO.xml.

```

<panel name="description"
inherits="description_${#type},description_${#topCategoryName},description"
/>

```

Ceci signifie que *description* <panel>, s'il est défini, hérite,

- Dans la première position de <panel name = "description_\${#type}" ...>;
 La balise *#{#type}* définit le type actuel dans le contexte, à choisir parmi les types possibles : Style, MyCustomStyle, Fabrics, Denim, Fur, Bags, Order, SKU, etc...
- En deuxième position (si la première n'est pas définie) de <panel name = "description_\${#topCategoryName}" ...>
 la balise *#{#topCategoryName}* définit la catégorie principale du contexte : Styles, Fabrics, Trims or PackagingLabel.
- En troisième position, si les deux autres ne sont pas définies, <panel name = "description" ...> définit par défaut dans le fichier xml de base.

Des alternatives doivent être recherchées dans le document ou le fichier xml du même nom dans le sous-dossier custom afin de déterminer quelles sont celles qui sont définies.

2.3.3 Modification de la description de l'écran

Dans les paragraphes et exemples suivants, nous allons voir comment modifier, ajouter <panel>, ajouter des champs de type <nodecombo>, <textarea>, <textfield>, etc...

- L'une des méthodes consiste à réécrire les paragraphes concernés (<panel>) avec le contenu modifié. C'est le moyen le plus facile de débiter.
- Une autre méthode consiste à utiliser la notion d'héritage abordée dans la section 2.2 - [Héritage](#). L'héritage est intégré dans 6 - [Configuration avancée pour l'ajout de champs et de panneaux](#).

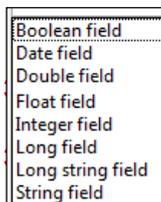
3. CONFIGURATION PAR L'AJOUT DE CHAMPS A « GENERAL OBJECTIVES »

L'écran **General Objectives** des produits (**Styles, Materials, Trims et Packaging Labels**) peut être configuré par la description de l'interface dans le fichier [namespaces/Lectra.PDM.ProductGO.xml](#).

3.1 Ajout de champs simples

3.1.1 Définition

Les champs simples sont des types de champs qui correspondent aux données suivantes dans le module Administration et configuration :



Toutes ces étapes doivent être appliquées :

3.1.4

3.1.5 Étape 1 : Prérequis du module Administration et configuration

Tous les champs à ajouter à votre écran doivent être définis dans le module Administration et configuration

Vous pouvez utiliser les nombreux champs disponibles ou en créer de nouveaux. Dans ce cas, veuillez consulter l'aide en ligne correspondante.



Vous pouvez également consulter le rappel disponible dans l'[Annexe H - Changement d'emplacement du répertoire de LPFExt](#)

[Dans](#) PLM V5R1, l'emplacement des fichiers LPFExt est sous `lpf/ext3/nom` du répertoire.

Rappel de l'application Administration et configuration

3.1.6 Étape 2 : Configuration dans le fichier XML dédié

Pour ajouter des champs à l'écran **General Objectives** des **Produits**, il est nécessaire d'entrer le fichier qui se trouve dans le sous-dossier « **custom** »

namespaces/custom/Lectra.PDM.ProductGO.xml (et pas le fichier **namespaces/Lectra.PDM.ProductGO.xml**)

S'il n'existe pas, il est nécessaire de le créer dans le dossier custom et de le structurer comme suit :

```

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

<!--      AJOUTER CONFIGURATION ICI      -->
<!--      _____      -->

</namespace>
```

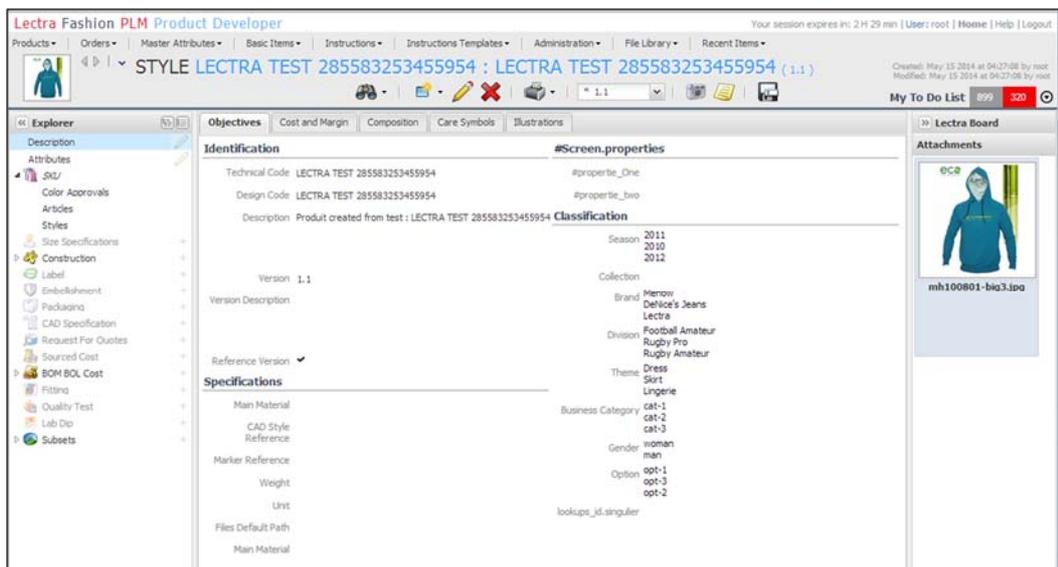
3.1.7 Étape 3 : Identification de la partie à dupliquer et modification

Pour chaque modification :

- Identifiez la partie concernée dans le fichier **/namespaces/LectraPDM.ProductGO.xml** (base xml)
- Copiez le <panel> qui contient cette partie dans le fichier **/namespaces/custom/LectraPDM.ProductGO.xml** (custom xml)
- Apportez la modification dans cette partie dupliquée.
- Mettez à jour l'affichage : utilisez la touche **F5** pour actualiser l'écran.

3.1.7.1 Identification de la partie à modifier dans la base XML

- Pour la catégorie **STYLE (modèle)** :



Recherchez le panneau de description Style (modèle) = contenu de l'onglet **General Objectives**.

Dans le fichier `/namespaces/LectraPDM.ProductGO.xml`, recherchez la partie qui décrit **General Objectives** pour la catégorie Style (modèle).

```
<panel name="description_Style" layout="column" autoScroll="true" >
```

Ce panneau est constitué de plusieurs panneaux...

Recherchez le panneau à modifier.

- Pour modifier le `<panel` « **Identification** », recherchez :

```
<panel name="identification" titleI18n="Screen.identification"
inherits="looklikefieldset" >
    .../...
</panel>
```

- Pour modifier le `<panel` « **Classification** », recherchez :

```
<panel name="collection" titleI18n="Screen.collection"
columnWidth="0.5" inherits="looklikefieldset" >
    .../...
</panel>
```

- Pour modifier le `<panel` « **Specifications** », recherchez :

```
<panel name="detail" titleI18n="Screen.specification"
inherits="looklikefieldset">
    .../...
</panel>
```

- Pour modifier le panneau « **Validation Table (Table de validation)** », recherchez :

```
<panel name="validationTable" ... >
    .../...
</panel>
```

Si vous voulez masquer les champs générés dans le panneau de table de validation, remplacez la propriété `headerOnly="false"` par `headerOnly="true"`.

Si vous voulez afficher un champ de validation dans l'écran ProductGo, vous devez ajouter un champ spécifique en fonction du type de champ à afficher. Le nom du champ correspond à la concaténation du nom de la table et de celui du champ (ex. : table name : 'VT', field name : 'Weight' => field name : 'VT_Weight').

Pour un champ numéroté, ajoutez la ligne suivante :

```
<numberfield isValidaionField="true" title18n="Weight" name="Table_Weight" >
```

Pour un champ textuel, ajoutez la ligne suivante :

```
<textfield isValidationField="true" title18n="Name" name="Table_Name" >
```

Pour un champ de date, ajoutez la ligne suivante :

Si vous voulez effectuer une modification en fonction du fuseau horaire du client :

```
<datefield isValidationField="true" title18n="Birth" name="Table_Birth" >
```

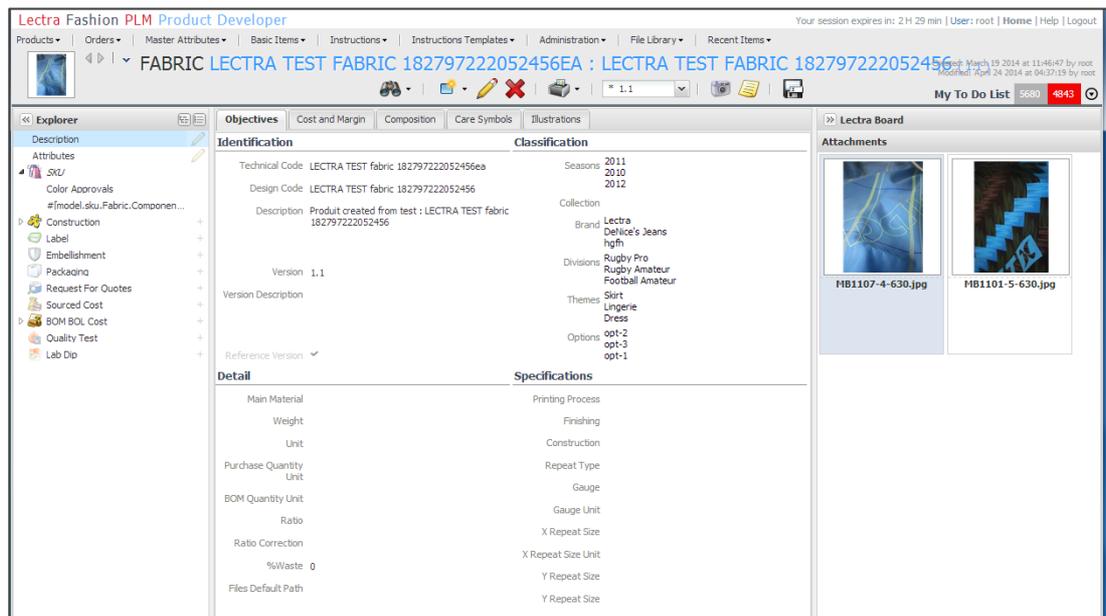
Si vous voulez que la date change en fonction du fuseau horaire du client :

```
<datefield isValidationField="true" title18n="Birth" name="Table_Birth"
useServerTimezone="true">
```

Pour un champ de case à cocher, ajoutez la ligne suivante :

```
<xcheckbox isValidationField="true" title18n="International" name="Table_International"
readOnly="true" >
```

- Pour les catégories FABRIC, TRIMS et PACKAGINGLABEL :



Recherchez le panneau de description Fabric.

Dans le fichier `/namespaces/LectraPDM.ProductGO.xml`, recherchez la partie qui décrit **GeneralObjectives** pour la catégorie Fabric

```
<panel name="description_default" layout="column" autoScroll="true" >
```

Suivez les panneaux liés par héritage jusqu'au panneau qui correspond à la partie à modifier.

Exemple : le `<panel>` « **description** » ci-dessus contient le panneau « **north** » qui hérite du panneau « **description_north_default** » qui contient le `<panel>` « **identification** » et le `<panel>` « **classification** ».



Veuillez consulter la section [2.2 - Héritage](#).

Recherchez le panneau à modifier

- Pour modifier le panel > « **Identification** » <du Fabric, recherchez :

```
<panel name="identification_default"
titleI18n="Screen.identification" inherits="looklikefieldset">
.../...
</panel>
```

- Pour modifier le <panel> « **Classification** », recherchez :

```
<panel inherits="looklikefieldset" name="classification_default"
titleI18n="Screen.collection" >
.../...
</panel>
```

- Pour modifier le panel > « **Details** » <, recherchez :

```
<panel name="detail_FabricTrim" inherits="detail_default">
.../...
</panel>
```

- Pour modifier le <panel> « **Specifications** », recherchez :

```
<panel name="specification_Fabric" inherits="specification_default">
.../...
</panel>
```

3.1.7.2 Duplication de la partie à modifier dans le fichier XML personnalisé

Copiez la description du **panel** qui contient le **panel** à modifier dans le fichier [namespaces/custom/Lectra.PDM.ProductGO.xml](#).

Exemple : Pour modifier le panneau Style identification :

```
<panel name="identification" titleI18n="Screen.identification" inherits="looklikefieldset" >
```

Copiez tout le panneau Description Style :

```
<panel name="description_Style" layout="column" autoScroll="true" >
.../...
</panel >
```

3.1.7.3 Ajout de la description xml d'un champ textuel

Une fois la partie pertinente identifiée et dupliquée (si cela n'est déjà fait) dans le fichier XML personnalisé, ajoutez la ligne suivante au **panel** à l'emplacement voulu :

```
<textfield name="MyCustomStringField" fieldLabelI18n = "MyCustomStringField" />
```



Pour ajouter d'autres types de champs simples, consultez l'[Annexe J - Ajout d'autres types de champs simples dans xml](#).

3.1.8 Étape 4 : Référencement des traductions du champ ajouté

Si vous avez ajouté un champ personnalisé portant le nom MyCustomStringField dans le module Administration et configuration, l'écran du module de développement produit affiche :



Si le nom du fichier apparaît précédé du caractère #, cela signifie que la traduction du nom n'existe pas.

Traduisez ce texte en modifiant le [fichier PLM-Fashion\PDMLectraPLMParam\messages_en.properties](#) en ajoutant à la fin du fichier :

MyCustomStringField = My String Field.

Le résultat sera :



Dans Internet Explorer, exécutez la commande suivante :

http:// « Server name »/pdm/admin.SessionMonitor.reloadConfig.wbx

Dans le menu Internationalization du module Administration et configuration, cliquez sur Vider le cache.

3.2 Ajout d'une liste

3.2.1 Définition

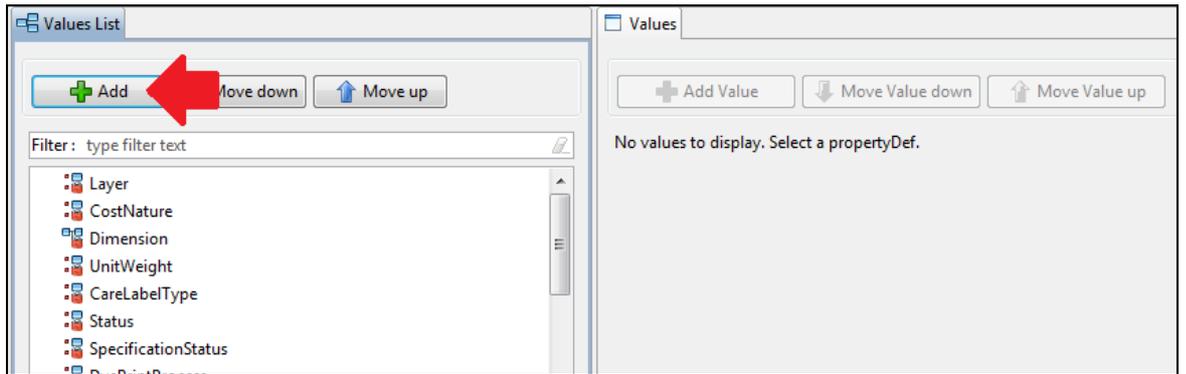
Appelées Value Lists (Listes de valeurs) dans l'application, elles sont associées aux catégories en ajoutant «CustomRole» en tant que «PickList» «Target» pour chaque liste de valeurs.

Dans le module de développement produit, elles sont ajoutées à l'écran via le composant `<nodecombo />`.

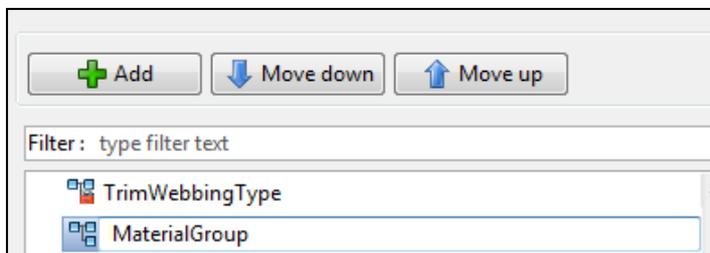
3.2.2 Étape 1 : Définition de la liste dans le module Administration et configuration et association à la catégorie Style

3.2.2.1 Création

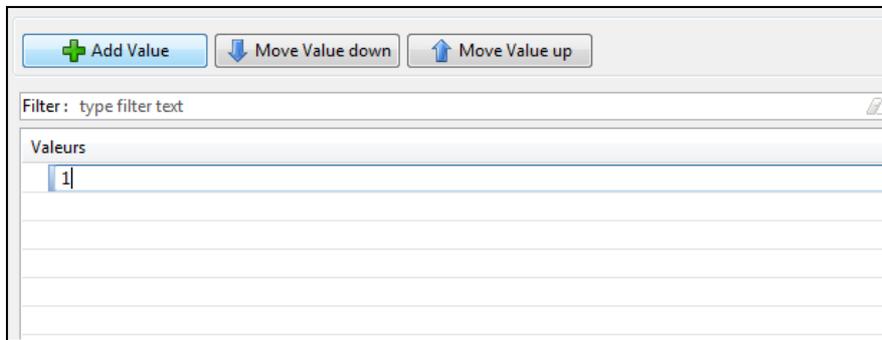
Cliquez sur **Listes de valeurs** dans le menu **Données d'Administration**.



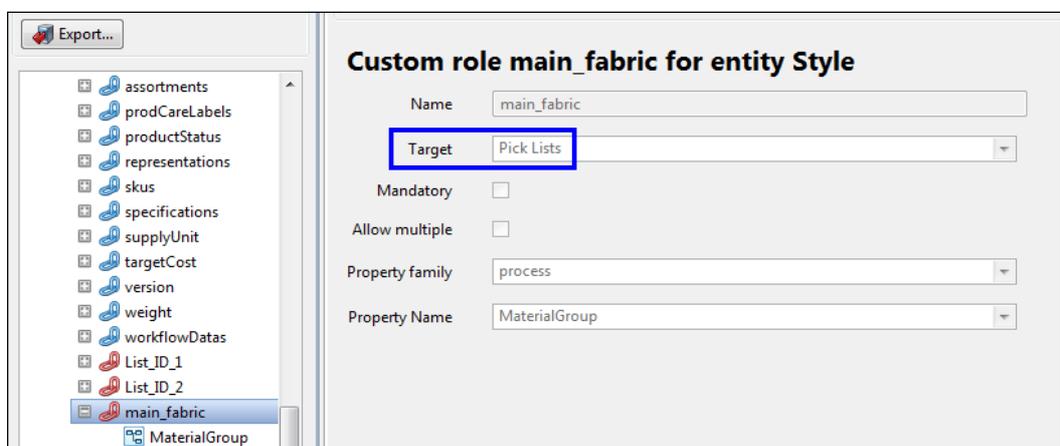
Cliquez sur **Ajouter** et entrez le nom de la liste de valeurs.



Sélectionnez la nouvelle liste et ajoutez les nouvelles valeurs dans l'écran de droite :



3.2.2.2 Association avec CustomRole



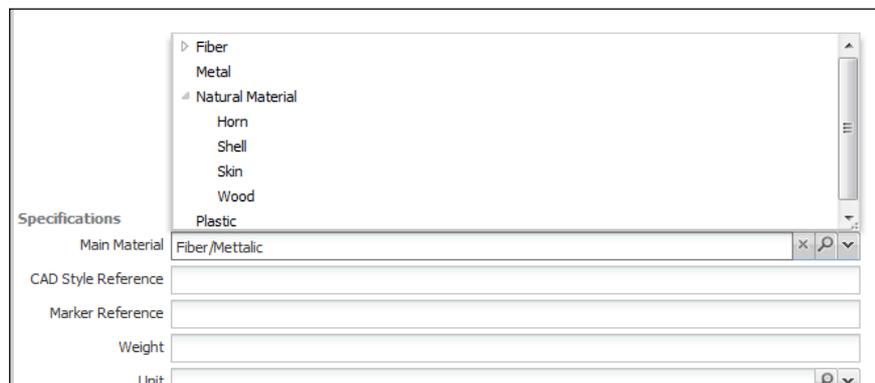
3.2.3 Étape 2 : Description du fichier custom/Lectra.PDM.ProductGO.xml

```
<panel name="description_Style"  
  <panel name="detail":  
    <nodecombo fieldLabelI18n="Style.mainFabric"  
              hiddenName="mainFabric" />
```

3.2.4 Étape 3 : Référencement des traductions du champ ajouté

Dans le fichier messages_en.properties.

3.2.5 Résultat : Affichage dans le module de développement produit



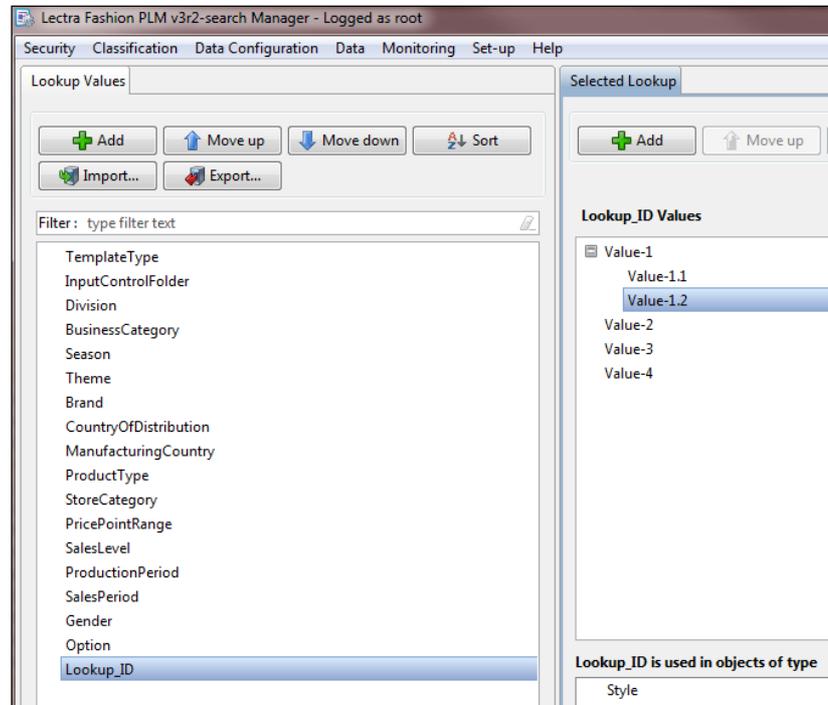
3.3 Ajout de la classification

3.3.1 Définition

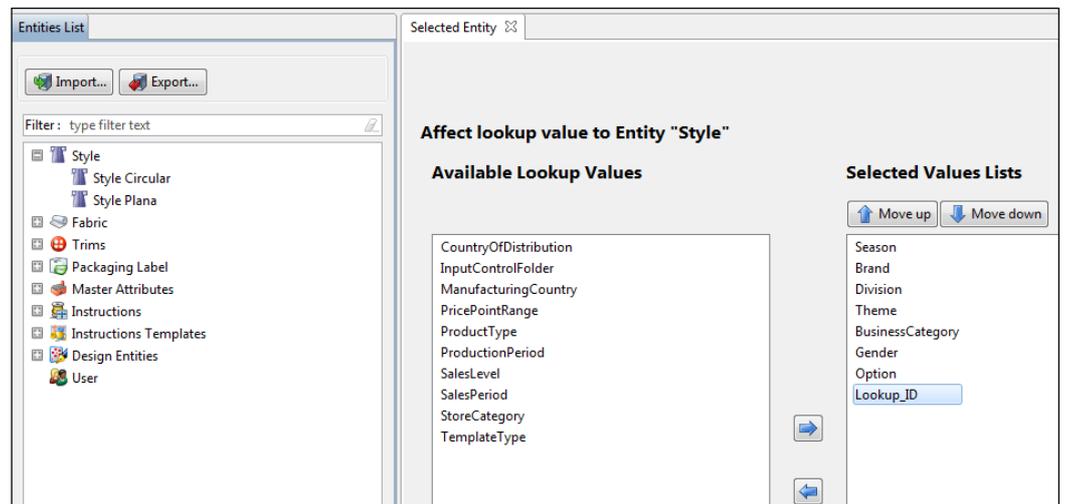
Appelées Lookup Values (Valeurs de classification) dans le menu Classification du module Administration et configuration, elles peuvent être créées en définissant les valeurs de classification de manière hiérarchique et elles sont associées à une ou plusieurs catégories (Affecter les valeurs de classification).

3.3.2 Étape 1 : Définition dans le module Administration et configuration

3.3.2.1 Création



3.3.2.2 Association à la catégorie Modèle



3.3.3 Étape 2 : Description dans le fichier Lectra.PDM.ProductGO.xml

Dans le module de développement produit, cette nouvelle liste doit apparaître automatiquement dans la partie Classification des produits concernés, sur la page GeneralObjectives.

3.3.3.1 Description automatique de la classification

- Vérifiez que le **panel Collection** est présent dans le fichier XML de base, celui présent dans le dossier **namespaces**.

```
<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>
```

</panel>

- **Custom Serializer**

Custom Serializer est utilisé pour générer un champ unique ou un groupe de champs. Il est défini dans les fichiers XML à l'aide de la balise `<customSerializer .../>` qui se trouve généralement dans une zone privée :

```
<!-- Private field(s) : do not edit -->
  <customSerializer ...
<!-- End of private field(s) -->
```

Cette définition ne doit pas être modifiée. Vous risqueriez de corrompre la génération des pages.

3.3.3.2 Description manuelle de la classification

- Dans un `<panel>` similaire à celui ci-dessus, il est possible de définir une liste de `<nodecombo>` en faisant référence à une liste HVL définie dans le module Administration et configuration, sans suivre le processus **Custom Serializer**.

Exemple :

- Avec une **Pick List (Liste de sélection)** définie dans le module Administration et configuration :
 - `<nodecombo fieldLabelI18n="ApprovalStatus" hiddenName="ApprovalStatus"
 propertyName="ApprovalStatus" multiSelect="false" chkVisible="false"/>`
- Avec une **Classification Lookup Value (Valeur de classification)** définie dans le module Administration et configuration : « **Brand** » (marque) par exemple
 - `<nodecombo fieldLabelI18n="brands_isa" hiddenName="brandsClassif"
 propertyFamily="process" propertyName="Brand" multiSelect="true" />`

 `hiddenName` doit être composé comme suit : `"Lookup Value Name"sClassif`.

- Si la **Pick List (Liste de sélection)** ou la **Classification Lookup Value (Valeur de classification)** est insérée comme critère de recherche dans le panneau **Search**

(Recherche) de l'explorateur de Produits, « *.values* » doit être référencé dans « *hiddenName* ».

- `<nodecombo fieldLabel18n="ApprovalStatus" hiddenName="ApprovalStatus.values" propertyName="ApprovalStatus" multiSelect="false" chkVisible="false"/>`

3.3.4 Étape 3 : Référencement des traductions du champ ajouté

Dans le fichier `messages_en.properties`.

3.3.5 Résultats : Affichage dans le module de développement produit



3.4 Ajout d'un panneau « Properties (Propriétés) »

Il est également possible de configurer l'écran avec un nouveau « panneau » composé de champs existants ou nouveaux.

3.4.1 Étape 1 : Identification du panneau dans lequel le panneau doit être ajouté

Le « panneau père » dans lequel le nouveau panneau sera inséré doit être copié intégralement dans le fichier `custom/Lectra.PDM.ProductGO.xml`.



Si l'un des panneaux du « panneau père » concerné a déjà été dupliqué et configuré dans le fichier XML du dossier personnalisé, il est nécessaire de copier la définition du panneau déjà modifié.

3.4.2 Étape 2 : Description dans le fichier « custom/Lectra.PDM.ProductGO.xml »

Pour ajouter le panneau « properties (propriétés) » à l'écran GeneralObjectives de la catégorie Styles :

3.4.2.1 Duplication de la définition du panneau « description_Style »

```
<panel name="description_Style" layout="column" autoScroll="true" padding="4" >
```

```

<panel columnWidth="0.5" border="false">
  <panel name="identification" titleI18n="Screen.identification"
    inherits="looklikefieldset" >
    .../...
  </panel>
  <panel name="detail" titleI18n="Screen.specification"
    inherits="looklikefieldset" >
    .../...
  </panel>
  <panel name="collection" titleI18n="Screen.collection"
    columnWidth="0.5" inherits="looklikefieldset" >
    .../...
  </panel>
</panel>

```

Pour ajouter le panneau « **Propriétés (Propriétés)** » à l'écran **GeneralObjectives** des catégories Fabrics, Trims et PackagingLabel (Tissus, Fournitures et Étiquette – Conditionnement) :

3.4.2.2 Duplication de la définition du panneau « description_north_default » ou « description_south_default »

```

<panel name="description_north_default" border="false" layout="column">
  <panel name="identification"
    inherits="identification_${#categoryName},identification_${#top
      CategoryName},identification_default" columnWidth="0.5"/>
  <panel name="classification"
    inherits="classification_${#categoryName},classification_${#top
      CategoryName},classification_default" columnWidth="0.5"/>
</panel>
<panel name="description_south_default" border="false" layout="column">
  <panel name="detail" inherits="detail_${#categoryName},
    detail_${#topCategoryName}, detail_default" columnWidth="0.5"/>
  <panel name="specification" inherits="specification_${#categoryName},
    specification_${#topCategoryName}, specification_default"
    columnWidth="0.5"/>
</panel>

```

3.4.2.3 Ajout de la définition du panneau « Propriétés (Propriétés) »

```

<panel name="description_Style" layout="column" autoScroll="true"
  padding="4">
  <panel name="identification" titleI18n="Screen.identification"
    inherits="looklikefieldset">
    .../...
  </panel>
  <panel name="detail" titleI18n="Screen.specification"
    inherits="looklikefieldset">
    .../...
  </panel>
  <panel name="properties" titleI18n="Screen.properties"
    columnWidth="0.5" inherits="looklikefieldset">
    <textfield fieldLabelI18n="propertie_One" />
    <textfield fieldLabelI18n="propertie_two" />
  </panel>
  <panel name="collection" titleI18n="Screen.collection"
    columnWidth="0.5" inherits="looklikefieldset">

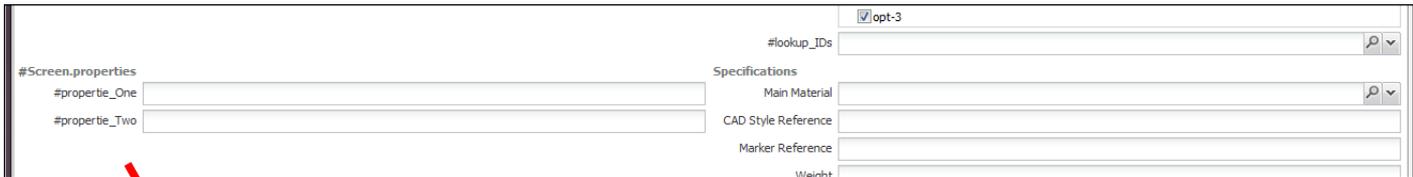
```

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

3.4.3 Étape 3 : Référencement des traductions des noms des champs ajoutés

Dans le fichier `messages_en.properties`, traduction de `Screen.properties`, `propertie_One`, `propertie_Two`

3.4.4 Résultat : Affichage dans le module de développement produit



4. CONFIGURATION EN AJOUTANT DES CHAMPS DANS L'EXPLORATEUR PRODUITS

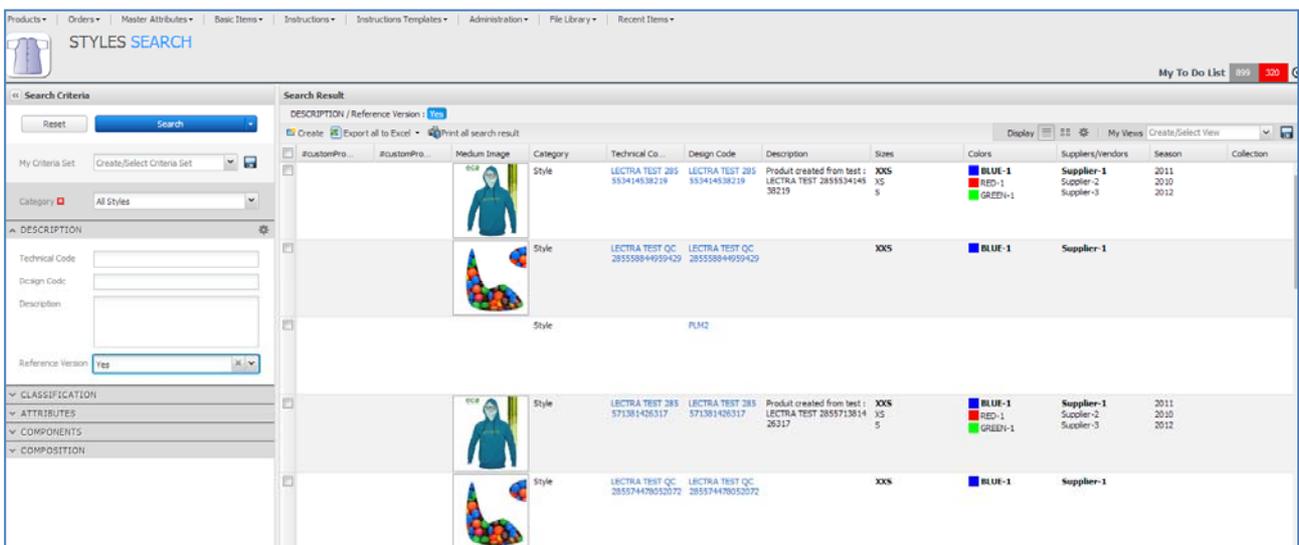
4.1 Prérequis du module Administration et configuration

Les champs personnalisés qui seront ajoutés à cet écran doivent être définis dans le module Administration et configuration, comme indiqué dans l'Annexe H - [Changement d'emplacement du répertoire de LPFExt](#)

Dans PLM V5R1, l'emplacement des fichiers LPFExt est sous lpf/ext3/nom du répertoire.

Rappel de l'application Administration et configuration.

4.2 Présentation de l'écran Product (Produit)



Cet écran est identique pour différents produits : **Styles (Modèles)**, **Fabrics (Tissus)**, **Trims (Fournitures)** et **PackagingLabel (Étiquette – Conditionnement)**. Il est constitué de deux parties personnalisables :

- Le panneau **Search (Recherche)** affiché sur la gauche et appelé **Search Criteria (Critère de recherche)**. Cette partie peut être configurée via le module Administration et configuration et en ajoutant les définitions dans le fichier XML (voir ci-dessous comment procéder).
- La grille **Results (Résultats)**, sur la droite, peut être configurée directement dans le module Administration et configuration en ajoutant/supprimant des colonnes, en modifiant l'ordre d'apparition, etc. Veuillez consulter le Guide d'utilisation du module de développement produit.

Pour configurer le panneau de recherche, il est important de comprendre comment le fichier XML est défini et comment intervenir dans cette définition.

4.3 Fichier XML personnalisable

L'explorateur Produits peut être configuré en modifiant la description de l'interface décrite dans le fichier [Lectra.PDM.Search.Product.xml](#). Il est nécessaire de compléter le fichier se trouvant dans le sous-dossier **custom namespaces/custom/Lectra.PDM.Search.Product.xml** (et pas [namespaces/Lectra.PDM.Search.Product.xml](#))

S'il n'existe pas, il est nécessaire de le créer dans le dossier custom et de le structurer comme suit :

```

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

<!-- AJOUTER CONFIGURATION ICI -->

<!-- _____ -->
  <explorergrid name="resultGrid_XXX" inherits="resultGrid">
    <!-- XXX : categoryName (Style, Fabric, etc ....) -->
    <!-- Columns description -->
    <!-- securityCells : define the tagNames tagName1, tagName1 etc
    -->
    <!-- tagName:column Name can be controlled before render
    explorer -->
    <colModel inherits="resultGrid/colModel"
      securityColumns="colorTag,supplierTag,descTag"
    securityBranch="Explorer" securityType="{#categoryName}"
    colorTag="colors,sizes"
    supplierTag="suppliers,seasonsClassif"
    descTag="description,codeAlpha1" />
    <store xsi:type="lpfdirectstore" inherits="resultGrid/store" />
  </explorergrid>

</namespace>
```

4.4 Structure du panneau de recherche dans le fichier XML

namespaces/Lectra.PDM.Search.Product.xml

Le panneau **Search (Recherche)** est défini par :

- un explorateur `<explorerform name="SearchForm"` qui est lui-même défini par :
- le panneau `<panel name="accordionPanel" id="myaccordionPanel"`

Chaque groupe de critères (autrement dit : groupe « **Description** » qui contient le critère « **Reference Version (Version de référence)** ») est défini par un panneau. Le contenu de chaque « **panel criteria group** » est défini dans le fichier, puis référencé dans la définition du panneau « **accordionPanel** ».

Exemple : Pour le groupe de critère « Classification » :

La description du panneau « **classification** » s'effectue comme suit :

```
<!-- ***** -->
<!--      Collection block definition (Classification)      -->
<!-- ***** -->
<panel name="classification" titleI18n="ProductGO.collection" collapsible="true">
  <nodecombo fieldLabelI18n="{#axeRoleName}" hiddenName="{#classifRoleName}.values" forceSelection="true"
    multiSelect="true" propertyName="{#propertyName}" genProductClassif="product"
    chkVisibleExpr="{#propertyName eq 'Season'} or {#propertyName eq 'Division'} or {#propertyName eq 'Theme'} or

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

Son inclusion dans « **AccordionPanel** » s'effectue comme suit :

```
<!-- ***** -->
<!--      PANEL : Accordion Panel      -->
<!-- ***** -->
<panel name="accordionPanel" id="myaccordionPanel" layout="accordionform" activeItem="0"
  autoScroll="true" border="false" isCriteriaPanel="true" style='border-top: 1px solid #fff'>
  <defaults style="border-top: 1px solid #000"/>
  <plugin xsi:type="xmlmap" ptype="panelchildrenvisibility"/>

  <!-- ***** -->
  <!--      PANEL : description      -->
  <!-- ***** -->
  <panel name="description" inherits="description_{#explorerName},description_{#topCategoryName},description" layout="form"
    border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
  <panel name="classification" inherits="classification_{#explorerName},classification_{#topCategoryName},classification"
    layout="form" border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
  <panel name="axis" inherits="axis_{#explorerName},axis_{#topCategoryName},axis" layout="form"
    border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
  <panel name="mainProduct" inherits="mainProduct_{#explorerName},mainProduct_{#topCategoryName},mainProduct" condition="{!#is
    border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
  <panel name="workflow" inherits="workflow_{#explorerName},workflow_{#topCategoryName},workflow" condition="{!#isPickerExplor
    border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
  <panel name="task" inherits="task_{#explorerName},task_{#topCategoryName},task" condition="{!#isPickerExplorer}" layout="for
    border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
</panel>
<!-- ***** -->
<!--      END PANEL Accordion      -->
<!-- ***** -->
```

4.5 Identification de la partie à dupliquer et modification

Pour chaque modification :

- Identifiez la partie concernée dans le fichier [//namespaces/LectraPDM.Search.Product.xml](#) (base xml)
- Copiez le <panel> qui contient cette partie dans le fichier [/namespaces/custom/LectraPDM. Search.Product.xml](#) (custom xml)
- Apportez la modification dans cette partie dupliquée.
- Mettez à jour l'affichage : utilisez la touche **F5** pour actualiser l'écran.

4.6 Ajout d'un critère à un groupe existant du panneau de recherche

Pour ajouter un critère au groupe « Description » par exemple, comme dans l'exemple précédent :

4.6.1 Identification du bloc de description du panneau « Description » dans le fichier XML de base

```
<!-- ***** -->
<!--           Description block definition           -->
<!-- ***** -->
<panel name="description" 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 fieldLabelI18n="RefVersion" hiddenName="refVersion.values"
    ignoreFormClear="true" inherits="http://lectra.com/pdm/common#yesno"
    conditionCtrl="com.lectra.lpf.uimodel.FeatureFilter" />
</panel>
```

4.6.2 Duplication de ce bloc dans le fichier XML personnalisé

4.6.3 Ajout des champs voulus dans ce bloc

Exemple 1 : Ajout d'une chaîne de caractères pour un champ Custom_Notes (Remarques_perso) créé dans le module Administration et configuration :

Ajoutez la ligne suivante au début du bloc :

```
<textarea name="Custom_Notes.values" fieldLabelI18n="Custom_Notes"
  chkVisible="true" />
```

Exemple 2 : Ajout d'une liste de pays créée dans le module Administration et configuration :

Ajoutez la ligne suivante au début du bloc :

```
<nodecombo fieldLabelI18n="Countries" propertyFamily="process"
  propertyName="Country" multiselect="false" chkVisible="false" />
```

4.7 Ajout d'un groupe de critères dans le panneau de recherche

À la description de **AccordionPanel** vue dans la section 4.4 - [Structure du panneau de recherche dans le fichier XML](#), dans le fichier `namespaces/Lectra.PDM.Search.Product.xml`, nous ajouterons un nouveau groupe composé de plusieurs champs créés dans le module Administration et configuration.

4.7.1 Identification du bloc de description « accordionPanel » dans le fichier XML de base

```

<!-- ***** -->
<!-- PANEL : Accordion Panel -->
<!-- ***** -->
<panel name="accordionPanel" id="myaccordionPanel" layout="accordionform" activeItem="0"
  autoScroll="true" border="false" isCriteriaPanel="true" style="border-top: 1px solid #fff">
  <defaults style="border-top: 1px solid #000"/>
  <plugin xsi:type="xmlmap" ptype="panelchildrenvisibility"/>

  <!-- ***** -->
  <!-- PANEL : description -->
  <!-- ***** -->
  <panel name="description" inherits="description_${explorerName},description_${topCategoryName},description" layout="form"
    border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
  <panel name="classification" inherits="classification_${explorerName},classification_${topCategoryName},classification"
    layout="form" border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
  <panel name="axis" inherits="axis_${explorerName},axis_${topCategoryName},axis" layout="form"
    border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
  <panel name="mainProduct" inherits="mainProduct_${explorerName},mainProduct_${topCategoryName},mainProduct" condition="$(!#is
    border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
  <panel name="workflow" inherits="workflow_${explorerName},workflow_${topCategoryName},workflow" condition="$(!#isPickerExplor
    border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
  <panel name="task" inherits="task_${explorerName},task_${topCategoryName},task" condition="$(!#isPickerExplorer)" layout="form"
    border="false" labelWidth="65" labelAlign="left" autoScroll="true" bodyStyle="padding: 14px 20px 10px 20px">
    <defaults anchor="100%" />
  </panel>
</panel>
<!-- ***** -->
<!-- END PANEL Accordion -->
<!-- ***** -->

```

4.7.2 Duplication de ce bloc dans le fichier XML personnalisé

4.7.3 Ajout de la description de ce nouveau groupe dans ce bloc

```

<panel name="CustomGroup" inherits="description_${explorerName},
  description_${topCategoryName}, CustomGroup" layout="form"
  border="false" labelWidth="65" labelAlign="left" autoScroll="true"
  bodyStyle="padding: 14px 20px 10px 20px">
  <defaults anchor="100%" />
</panel>

```

Ajoutez ce bloc avec le nom de groupe correct à l'emplacement voulu, sous le commentaire « PANEL: description ».

4.7.4 Définition de ces champs comme critères de recherche

Les champs ajoutés à ce panneau de recherche de l'écran Produits (quelle que soit la catégorie) sont des champs spécifiques. Leurs valeurs ne sont pas seulement affichées (comme dans l'écran GeneralObjectives) ; elles facilitent la recherche de produits. Ces champs doivent être identifiés comme critères de recherche dans le module Administration et configuration.

Cette définition s'effectue dans un fichier Excel dont la structure est établie et doit être correctement entrée. Ce fichier doit être importé dans le module Administration et configuration.

 Consultez l'aide en ligne du module Administration et configuration, paragraphe 13 : **Critères de recherche**
 Vous pouvez également consulter le rappel disponible dans l'[Annexe H - Changement d'emplacement du répertoire de LPFExt](#)

Dans PLM V5R1, l'emplacement des fichiers LPFExt est sous lpf/ext3/nom du répertoire.

Rappel de l'application Administration et configuration

4.8 Ajout d'une colonne à la grille des résultats

4.8.1 Identification de la description de la grille des résultats

Identifiez la partie concernée dans le fichier namespaces/Lectra.PDM.Search.Product.xml.

```
<!-- ***** -->
<!-- Grid definition (result): explorerGrid (colModel and store) -->
<!-- ***** -->
<explorereditablegrid name="resultGrid" explorerSubType="Product"
  explorerType="${#categoryName}" lockColumn="true"
  forceValidation="true" multiCell="true"
  stateId="ExplorerProduct_grid" stateful="true"
  tempCreationId="codeNum2" singleSelectExpr="${#singleSelect}"
  quickCreateMode="false" >

.../ ...

</explorereditablegrid>
```

Cette description <explorereditablegrid> contient 2 parties à modifier :

- La liste des colonnes de la grille :

```
<!-- Columns description -->
<colModel> .../... </colModel>
```

- Le « Store (Magasin) » de données à collecter dans la base pour remplir la grille :

```
<!-- Data description -->
<!-- Advice : Do not edit to keep all data definition -->
<store xsi:type="lpfdirectstore" remoteSort="true"
  forceUpdateRecordsOnFail="true">
.../...
</store>
```

4.8.2 Duplication de ce bloc dans le fichier XML personnalisé

Si vous ne connaissez pas la notion de la section 2.2 - Héritage, dupliquez tout le bloc <explorereditablegrid> dans le fichier personnalisé namespaces/custom/Lectra.PDM.Search.Product.xml et ajoutez les 2 lignes nécessaires à l'ajout d'une colonne.

4.8.3 Ajout de la description d'une colonne de type textuel

Pour ajouter une colonne qui insère un type de données simple :

Ajoutez la colonne.

Dans `<colModel>`, ajoutez, par exemple :

Pour du texte :

```
<column      xsi:type="textcolumn"      headerI18n="TEXT_NAME"
dataIndex="TEXT_NAME_INDEX" />
```

Pour la date, si vous voulez que la date change en fonction du fuseau horaire du client :

```
<column      xsi:type="datecolumn"      headerI18n="TEXT_NAME"
dataIndex="DATE_INDEX" />
```

Pour la date, si vous ne voulez pas que la date change en fonction du fuseau horaire du client :

```
<column      xsi:type="datecolumn"      headerI18n="TEXT_NAME"
dataIndex="DATE_INDEX" useServerTimezone="true" />
```

1. Ajoutez la référence au « Store (Magasin) »

Dans le bloc `<store>`, ajoutez, par exemple :

```
<dataField name="TEXT_NAME_INDEX" />
<dataField name=" DATE_INDEX" />
```



Vous pouvez également consulter l'[Annexe Erreur ! Source du renvoi introuvable.](#) - [Erreur ! Source du renvoi introuvable.](#)

4.8.4 Le store (Magasin)

Le Store (Magasin) doit être mis à jour pour que la grille soit correctement complétée avec les valeurs requises. Pour la mise à jour, chaque type de données contenu dans la grille doit être référencé à l'aide d'une balise `<dataField>` dans la description du Store (Magasin). Par conséquent, dans le cadre du processus de récupération du « produit » actuel et de ses attributs, seules les valeurs requises des attributs du « produit » seront extraites.

Pour chaque attribut nécessaire pour l'affichage, 3 modes permettent de définir un niveau de filtre supplémentaire. Avec le mode = «full», «shallow» ou «strict», les données seront triées par ordre décroissant.

mode="full": total. L'intégralité de l'objet (y compris ses attributs) est chargée depuis la base.

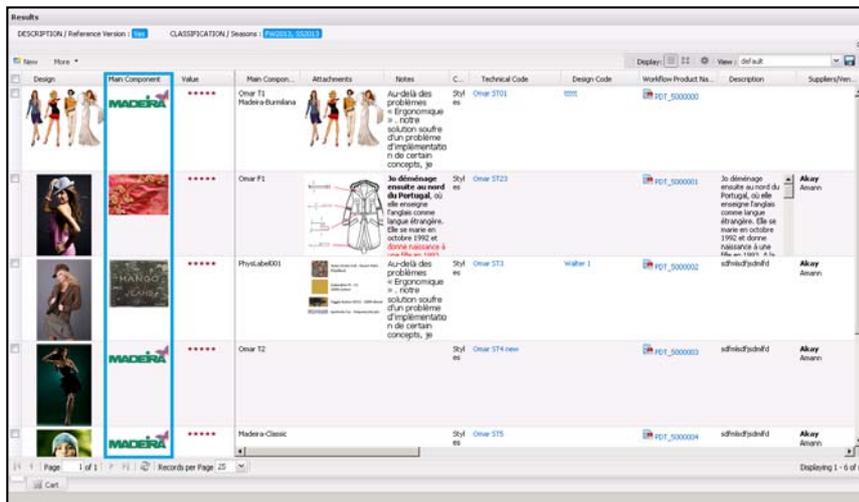
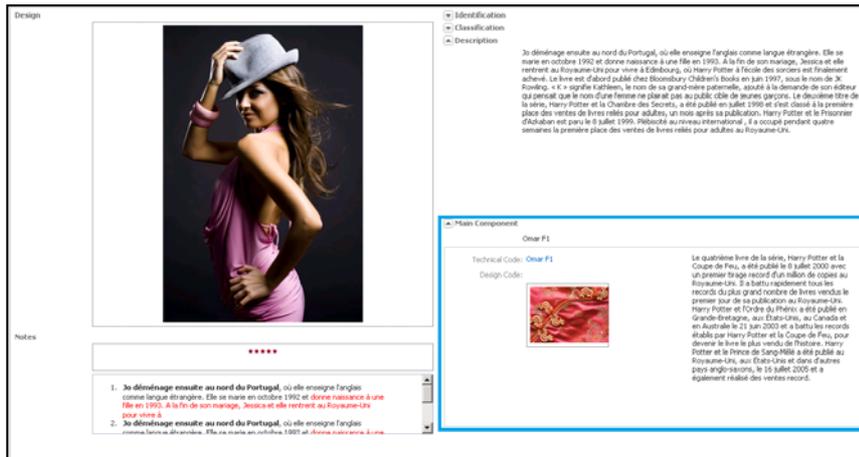
mode="shallow": partiel. Les attributs simples de l'objet seront récupérés de la base.

mode="strict": minimal. Seul le nom est récupéré.

4.8.5 Ajout de colonnes aux champs personnalisés

Voici un exemple de configuration :

Objectif : Lier le produit dans Style GO et l'insérer dans la grille de recherche.



Étape 1 : Prérequis du module Administration et configuration

Définition d'un « **Custom Role** » avec « **Product** » « **Target** » sur l'entité « **Style** ». Dans l'exemple, il est appelé « **mainComponent** ».

Étape 2 : Ajout du panneau «mainComponent» à l'écran GeneralObjectives

fichier `LectraPLMParam\namespaces\custom\Lectra.PDM.ProductGO.xml`

```
<panel name="mainComponent" titleI18n="mainComponent" columnWidth="1.0"
border="false" collapsible="true" >
  <productcombo hiddenName="mainComponent" openerType="{#type}" />
  <panel columnWidth="1." layout="column" >
    <panel columnWidth="0.4" border="false">
      <renderfield name="mainComponent" fieldLabelI18n="technicalCode"
listenToProductCombo="mainComponent">
      <renderFn name=
"LECTRA.PDM.Format.productLinkRenderer('codeAlpha1')" />
    </renderfield>
      <renderfield name="mainComponent" fieldLabelI18n="studyCode"
listenToProductCombo="mainComponent">
      <renderFn name="LECTRA.PDM.Format.productLinkRenderer
('codeAlpha2')" />
    </renderfield>
  </panel>
  </panel>
</panel>
```

```

        <imagefield name="mainComponent.defaultImageField"
            imageSize="large" readOnly="true" submitValue="false"
            listenToProductCombo="mainComponent" >
            <managerInstance name="Lectra.PDM.Mgrs.imageFieldMgr" />
        </imagefield>
    </panel>
    <panel columnWidth="0.6" border="false" >
        <textarea name="mainComponent.description" readOnly="true"
            submitValue="false" listenToProductCombo="mainComponent" />
    </panel>
</panel>

```

Étape 3 : Ajout de la colonne à la grille des résultats de l'explorateur Produits

fichier LectraPLMParam\namespaces\custom\Lectra.PDM.Search.Product.xml

```

<!-- Grid definition (result) : explorerGrid (colModel and store)-->
<explorergrid name="resultGrid" explorerSubType="Product"
    explorerType="{#categoryName}" lockColumn="true"
    forceValidation="true" multiCell="true"
    stateId="ExplorerProduct_grid" stateful="true"> <!--
    autoLoadOnRender="true">-->

    <!-- Columns description -->
    <colModel>
        .../...
        <column xsi:type="productcolumn" headerI18n="mainComponent"
            dataIndex="mainComponent" subDataIndex="defaultImageField"
            editable="true" />
        <column xsi:type="imagecolumn" headerI18n="mainComponent"
            dataIndex="mainComponent" subDataIndex="defaultImageField" />
        .../....
    </colModel>
    <!-- Data description -->
    <!-- Advice : Do not edit to keep all data definition -->
    <store xsi:type="lpfdirectstore" remoteSort="true"
        forceUpdateRecordsOnFail="true">
        .../...
        <dataField name="mainComponent" mode="shallow">
            <filter name="defaultImageField" mode="full" />
        </dataField>
        .../...
    </store>
    .../...
</explorergrid>

```

Code à ajouter

5. CONFIGURATION EN AJOUTANT DES CHAMPS AU PLAN DE COLLECTION

5.1 Prérequis du module Administration et configuration

Les champs personnalisés qui seront ajoutés à cet écran doivent être définis dans le module Administration et configuration, comme indiqué dans l'[Annexe H - Changement d'emplacement du répertoire de LPFExt](#)

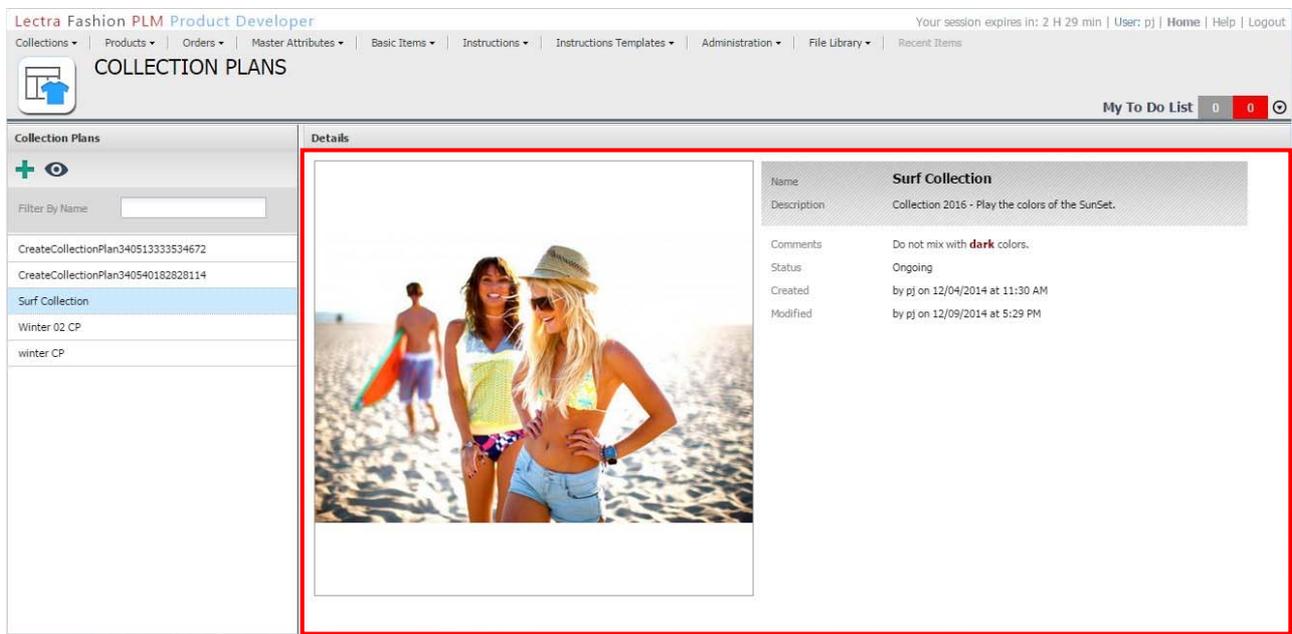
Dans PLM V5R1, l'emplacement des fichiers LPFExt est sous lpf/ext3/nom du répertoire.

Rappel de l'application Administration et configuration, [Appendix K - Ajout de champs et de rôles personnalisés aux données générales d'un plan de collection](#), [Annexe L – Ajout de champs et de rôles personnalisés à la grille Déclinaison d'un plan de collection](#).

5.2 Présentation des écrans du plan de collection

5.2.1 Écran du plan de collection

5.2.1.1 Présentation de l'écran



COLLECTION PLANS															
<p>Filter By Name: <input type="text"/></p> <ul style="list-style-type: none"> CreateCollectionPlan340513333534672 CreateCollectionPlan340540182828114 Surf Collection Winter 02 CP winter CP 	<p>Details</p> <div style="border: 2px solid red; padding: 5px;">  <table border="1"> <thead> <tr> <th colspan="2">Surf Collection</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Surf Collection</td> </tr> <tr> <td>Description</td> <td>Collection 2016 - Play the colors of the SunSet.</td> </tr> <tr> <td>Comments</td> <td>Do not mix with dark colors.</td> </tr> <tr> <td>Status</td> <td>Ongoing</td> </tr> <tr> <td>Created</td> <td>by pj on 12/04/2014 at 11:30 AM</td> </tr> <tr> <td>Modified</td> <td>by pj on 12/09/2014 at 5:29 PM</td> </tr> </tbody> </table> </div>	Surf Collection		Name	Surf Collection	Description	Collection 2016 - Play the colors of the SunSet.	Comments	Do not mix with dark colors.	Status	Ongoing	Created	by pj on 12/04/2014 at 11:30 AM	Modified	by pj on 12/09/2014 at 5:29 PM
Surf Collection															
Name	Surf Collection														
Description	Collection 2016 - Play the colors of the SunSet.														
Comments	Do not mix with dark colors.														
Status	Ongoing														
Created	by pj on 12/04/2014 at 11:30 AM														
Modified	by pj on 12/09/2014 at 5:29 PM														

Cet écran comporte une partie personnalisable unique : le centre (red framed). Remarque - les informations sont en 'lecture seule'. Le mode de modification n'est pas possible sur cet écran.

5.2.1.2 Fichier XML personnalisable

L'écran de liste du plan de collection peut être personnalisé en modifiant le fichier [Lectra.PDM.CollectionPlan.ListCollectionPlan.xml](#). Il est nécessaire de compléter le

fichier se trouvant dans le sous-dossier **custom namespaces/custom/**
[Lectra.PDM.CollectionPlan.ListCollectionPlan.xml](#) (et non dans l'original <2756namespaces/
[Lectra.PDM.CollectionPlan.ListCollectionPlan.xml](#))

S'il n'existe pas, il est nécessaire de le créer dans le dossier custom et de le structurer comme suit :

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

  <fieldset name="ListCollectionDetailsDefault" border="false" hidden="true"
layout="column" padding="0">
    <fieldset inherits="ListCollectionDetails/ListCollectionDetailsLeft" >
      <!-- See Dictionary.xml for the available kinds of fields -->
    </fieldset>
    <fieldset inherits="ListCollectionDetails/ListCollectionDetailsRight" >
      <!-- See Dictionary.xml for the available kinds of fields -->
    </fieldset>
  </fieldset>
</namespace>
```

5.2.1.3 Identification de la partie à modifier



La partie centrale possède la structure suivante :

listCollectionDetails (fieldset)

 |_ listCollectionDetailsLeft (fieldset)

 |_ imagefield

 |_ listCollectionDetailsRight (fieldset)

 |_ no name (fieldset)

 |_ name (displayfield)

 |_ description (displayfield)

 |_ no name (fieldset)

 |_ comments (displayfield)

 |_ status (displayfield)

 |_ created (displayfield)

 |_ modified (displayfield)

5.2.1.4 Ajout de champs sous imagefield de listCollectionDetailsLeft

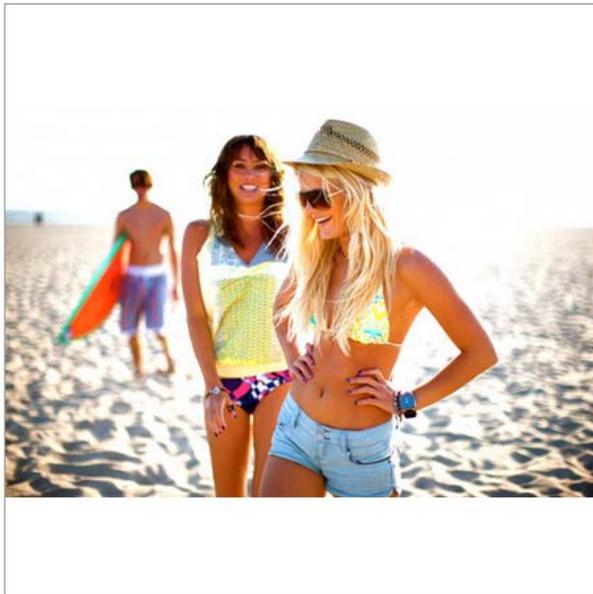
Ajoutez des champs et des rôles que vous avez créés dans le module Administration et configuration, dans la balise fieldset :

```
<fieldset inherits="listCollectionDetails/listCollectionDetailsLeft" >
  <!-- See Dictionary.xml for the available kinds of fields -->
</fieldset>
```

Et réactualisez votre explorateur pour voir la modification.

Exemple : case à cocher affichée sous l'image

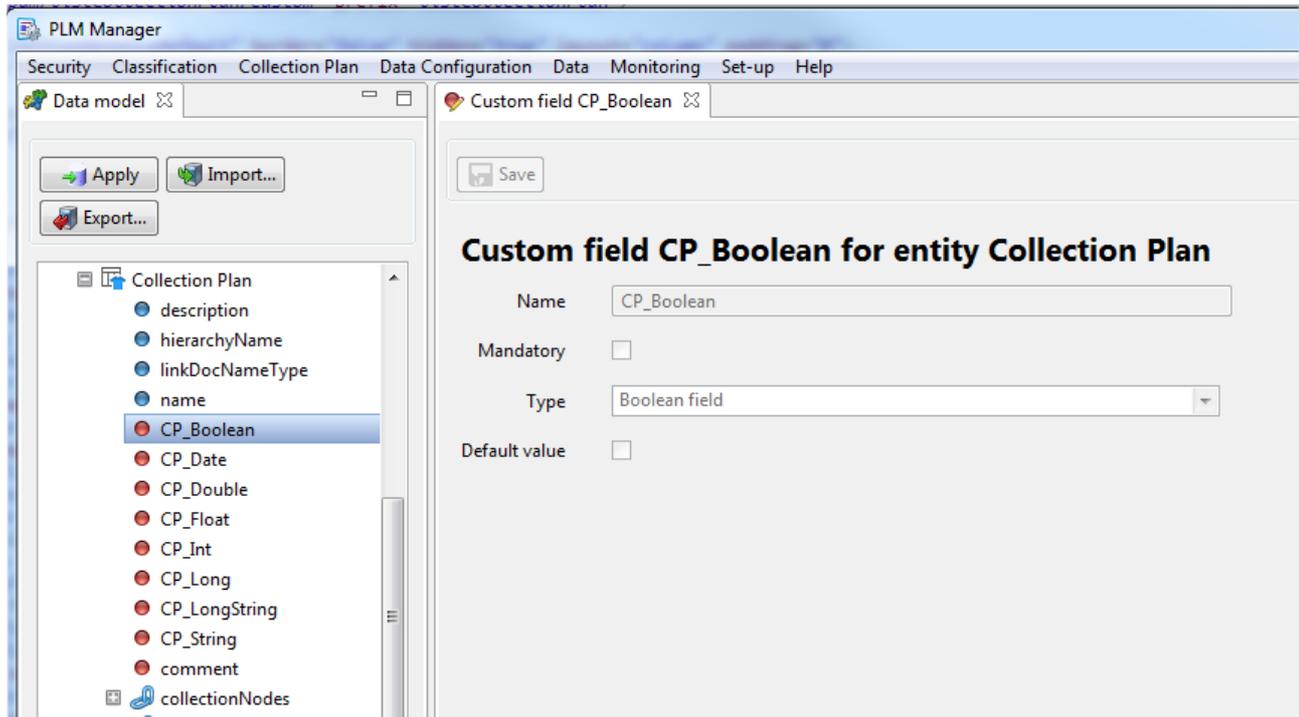
```
<fieldset inherits="listCollectionDetails/ListCollectionDetailsLeft" >  
  <!-- See Dictionary.xml for the available kinds of fields -->  
  <checkbox name="CP_Boolean" fieldLabelI18n="custom.cp.label.boolean" readOnly="true" />  
</fieldset>
```



To improve

Name	Surf Collection
Description	Collection 2016 - Play the colors of the SunSet.
Comments	Do not mix with dark colors.
Status	Ongoing
Created	by pj on 12/04/2014 at 11:30 AM
Modified	by pj on 12/09/2014 at 5:29 PM

Notez que le nom de la case à cocher est identique à celui créé dans le module Administration et configuration :



5.2.1.5 Ajout de champs jusqu'à imagefield de listCollectionDetailsLeft

Si vous devez ajouter des champs et des rôles jusqu'à imagefield, vous ne pouvez plus utiliser le mécanisme d'héritage. La balise fieldset doit être redéfinie en copiant celle du fichier [namespaces/Lectra.PDM.CollectionPlan.ListCollectionPlan.xml](#) d'origine :

```
<fieldset name="ListCollectionDetailsLeft" padding="10" border="false" margin="-4 0 0 0">
  <imagefield fieldLabelI18n="action.toggle.storyBoard" hideLabel="true" name="attachment" readOnly="true" useREST="true"
    inherits="http://Lectra.com/pdm/common#defaultImageFieldEXT5" managerRef="attachmentImageField"
    imageWidth="450" imageHeight="450" width="460" height="460" imageSize="full">
    <managerInstance name="Lectra.PDM.Mgrs.imageFieldRestMgr"/>
  </imagefield>
</fieldset>
```

À la suite de cela, elle doit être copiée dans le fichier personnalisé [namespaces/custom/Lectra.PDM.CollectionPlan.ListCollectionPlan.xml](#), à la place de la balise fieldset héritée, et modifiée comme il convient.

Exemple : case à cocher affichée au-dessus de l'image

```
<fieldset name="ListCollectionDetailsLeft" padding="10" border="false" margin="-4 0 0 0">
  <checkbox name="CP_Boolean" fieldLabelI18n="custom.cp.label.boolean" readOnly="true" />
  <imagefield fieldLabelI18n="action.toggle.storyBoard" hideLabel="true" name="attachment" readOnly="true" useREST="true"
    inherits="http://Lectra.com/pdm/common#defaultImageFieldEXT5" managerRef="attachmentImageField"
    imageWidth="450" imageHeight="450" width="460" height="460" imageSize="full">
    <managerInstance name="Lectra.PDM.Mgrs.imageFieldRestMgr"/>
  </imagefield>
</fieldset>
```

To improve



Name	Surf Collection
Description	Collection 2016 - Play the colors of the SunSet.
Comments	Do not mix with dark colors.
Status	Ongoing
Created	by pj on 12/04/2014 at 11:30 AM
Modified	by pj on 12/09/2014 at 5:29 PM

5.2.1.6 Ajout de champs sous les balises fieldset de listCollectionDetailsRight

Ajoutez des champs et des rôles que vous avez créés dans le module Administration et configuration, dans la balise fieldset :

```
<fieldset inherits="ListCollectionDetails/ListCollectionDetailsRight" >

  <!-- See Dictionary.xml for the available kinds of fields -->

</fieldset>
```

Réactualisez votre explorateur pour voir la modification.

Exemple : une zone de texte, une case à cocher, un nombre entier, une date, une combinaison multiple, une combinaison unique, une image et une combinaison envitem affichée sous les balises fieldset

```
<fieldset inherits="ListCollectionDetails/ListCollectionDetailsRight" >

  <!-- See Dictionary.xml for the available kinds of fields -->
  <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>
</fieldset>
```



Surf Collection	
Name	Surf Collection
Description	Collection 2016 - Play the colors of the SunSet.
Comments	Do not mix with dark colors.
Status	Ongoing
Created	by pj on 12/04/2014 at 11:30 AM
Modified	by pj on 12/09/2014 at 5:29 PM
Other comment	Do not forget to see with designers
To improve	✓
Prevision	45600
End	04/08/15
Countries	Italie Espagne
Original country	France
Add idea	
Main color	YELLOW-1

5.2.1.7 Ajout de champs sous la première balise fieldset de listCollectionDetailsRight

Si vous devez ajouter des champs et des rôles juste sous cette balise fieldset, vous ne pouvez plus utiliser le mécanisme d'héritage. Vous devez alors redéfinir la balise fieldset englobant listCollectionDetailsRight, en copiant celle du fichier [namespaces/Lectra.PDM.CollectionPlan.ListCollectionPlan.xml](#) d'origine :

```
<fieldset name="listCollectionDetailsRight" width="520" padding="10" border="false" margin="-1 0 0 -20">
  <fieldset padding="10" margin="0" border="false" cls="pdm-identitycard">
    <defaults labelSeparator=" " />
    <displayfield name="name" fieldLabelI18n="name" fieldCls="pdm-header-1" />
    <displayfield name="description" fieldLabelI18n="description" />
  </fieldset>
  <fieldset padding="10" border="false" margin="-3 0 0 0">
    <defaults labelSeparator=" " />
    <displayfield name="comment" fieldLabelI18n="comment" />
    <displayfield name="statusName" fieldLabelI18n="status" />
    <displayfield name="formatCreatedDateWithUser" fieldLabelI18n="Label.authoring.created.nosemicolon" />
    <displayfield name="formatModifiedDateWithUser" fieldLabelI18n="Label.authoring.modified.nosemicolon" />
  </fieldset>
</fieldset>
```

À la suite de cela, elle doit être collée dans le fichier personnalisé [namespaces/custom/Lectra.PDM.CollectionPlan.ListCollectionPlan.xml](#), à la place de la balise fieldset héritée, et modifiée comme il convient :

Exemple : case à cocher affichée après le champ de description.

```
<fieldset name="listCollectionDetailsRight" width="520" padding="10" border="false" margin="-1 0 0 -20">
  <fieldset padding="10" margin="0" border="false" cls="pdm-identitycard">
    <defaults labelSeparator=" " />
    <displayfield name="name" fieldLabelI18n="name" fieldCls="pdm-header-1" />
    <displayfield name="description" fieldLabelI18n="description" />
    <checkbox name="CP_Boolean" fieldLabelI18n="custom.cp.label.boolean" readOnly="true" />
  </fieldset>
  <fieldset padding="10" border="false" margin="-3 0 0 0">
    <defaults labelSeparator=" " />
    <displayfield name="comment" fieldLabelI18n="comment" />
    <displayfield name="statusName" fieldLabelI18n="status" />
    <displayfield name="formatCreatedDateWithUser" fieldLabelI18n="Label.authoring.created.nosemicolon" />
    <displayfield name="formatModifiedDateWithUser" fieldLabelI18n="Label.authoring.modified.nosemicolon" />
  </fieldset>
</fieldset>
```

	Name Surf Collection
	Description Collection 2016 - Play the colors of the SunSet.
	To improve ✓
	Comments Do not mix with dark colors.
	Status Ongoing
	Created by pj on 12/04/2014 at 11:30 AM
	Modified by pj on 12/09/2014 at 5:29 PM

5.2.1.8 Ajout de champs et modification de la balise globale fieldset listCollectionDetails

Si vous devez ajouter des champs et des liens, de façon à disposer d'un autre rendu de disposition, consultez la section Modification de la mise en page.

5.2.1.9 Remarques relatives aux champs et aux liens dans cet écran

Voici différents champs et liens (avec des exemples) que vous pouvez ajouter à cet écran, et qui sont en lecture seule :

Champ de chaîne

```
<textfield name="CP_String" fieldLabel18n="custom.cp.label.string" readOnly="true" />
```

LongString field (Long champ de chaîne)

```
<textarea name="CP_LongString" fieldLabel18n="custom.cp.label.longstring" readOnly="true" />
```

Boolean field (Champ booléen)

```
<checkbox name="CP_Boolean" fieldLabel18n="custom.cp.label.boolean" readOnly="true" />
```

Long field (Long champ)

```
<longfield name="CP_Long" fieldLabel18n="custom.cp.label.long" readOnly="true" />
```

Integer field (Champ d'entier)

```
<integerfield name="CP_Int" fieldLabel18n="custom.cp.label.int" readOnly="true" />
```

Double field (Champ double)

```
<doublefield name="CP_Double" fieldLabel18n="custom.cp.label.double" readOnly="true" />
```

Float field (Champ flottant)

```
<floatfield name="CP_Float" fieldLabel18n="custom.cp.label.float" readOnly="true" />
```

Date field (Champ de date)

```
<datefield name="CP_Date" fieldLabel18n="custom.cp.label.date" readOnly="true" />
```

Exemple de HVL pick list (liste de sélection HVL) avec Pays et Sélection multiple

```
<nodecombo name="CP_Country_Multi" multiSelect="true" propertyName="Country" fieldLabel18n="custom.cp.label.hvl.multi" >
```

```
<bind readOnly="true" />
```

```
</nodecombo>
```

Exemple de HVL pick list (liste de sélection HVL) avec Pays et Sélection unique

```
<nodecombo name="CP_Country_Single" multiSelect="false" propertyName="Country" fieldLabel18n="custom.cp.label.hvl.single" >
```

```
<bind readOnly="true" />
```

```
</nodecombo>
```

Sélection du document cible

```
<imagefield name="CP_Doc" fieldLabel18n="custom.cp.label.document" useREST="true"
```

```
inherits="http://lectra.com/pdm/common#defaultImageFieldEXT5"
```

```
managerRef="selSmallImageField" imageWidth="96" imageHeight="96" readOnly="true" >
```

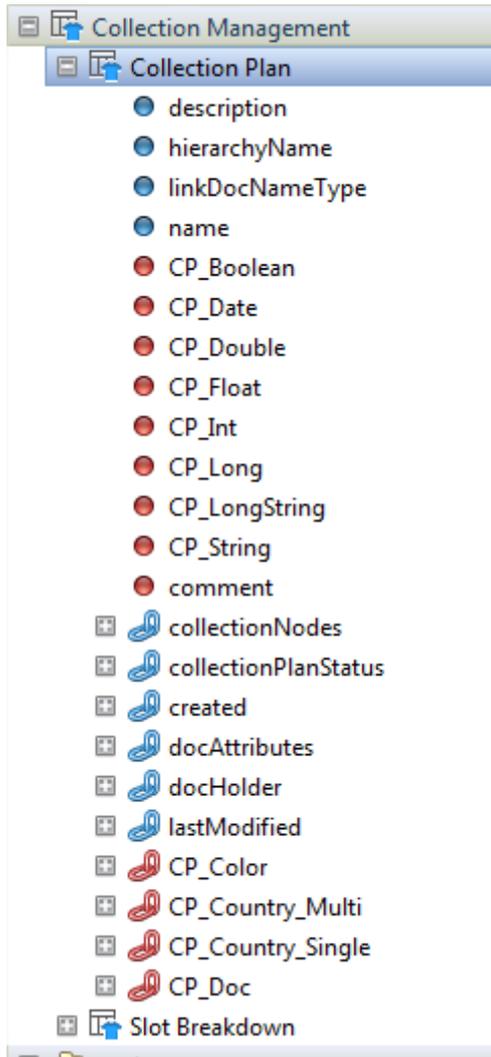
```
<managerInstance name="Lectra.PDM.Mgrs.imageFieldRestMgr" />
```

```
</imagefield>
```

envitem (sélection multiple interdite) : exemple avec Couleur marketing

```
<envitemcombo name="CP_Color" envItemtype="MarketingColor" fieldLabel18n="custom.cp.label.envitem.color" hideTrigger2="true" readOnly="true" />
```

Notez que l'attribut de la valeur *name* est identique à celui créé dans le module Administration et configuration :



Notez que l'attribut de la valeur *fieldLabel18n* doit être défini dans le fichier [PLM-Fashion\PDM\LectraPLMParam\messagesCusto_en.properties](#) (et dans un autre fichier de langue, si nécessaire ; exemple [messagesCusto_fr.properties](#) pour le français)

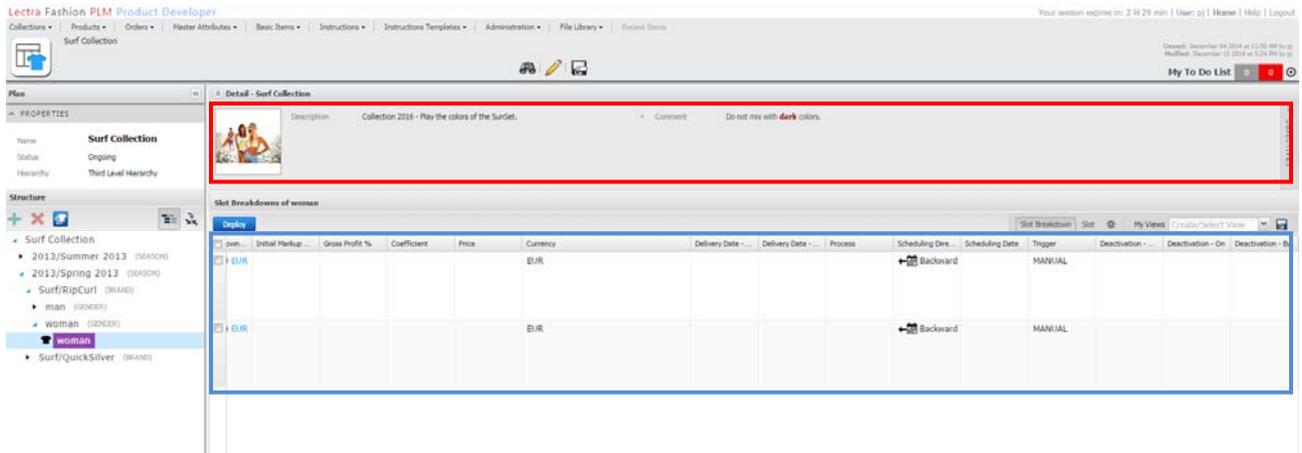
```

1 custom.cp.label.longstring=Other comment
2 custom.cp.label.boolean=To improve
3 custom.cp.label.int=Prevision
4 custom.cp.label.date=End
5 custom.cp.label.hvl.country.multi=Countries
6 custom.cp.label.hvl.country.single=Original country
7 custom.cp.label.document=Add idea
8 custom.cp.label.envitem.color=Main color
9 custom.cp.rp.label.date=Begining
10 custom.cp.rp.label.boolean=Stop
11 custom.cp.rp.label.string=Comment 2
12 custom.cp.rp.label.hvl.country.multi=Countries
13 custom.cp.rp.label.hvl.country.single=Country
14 custom.cp.rp.label.document=Recent image
15 custom.cp.rp.label.envitem.color=Main color
16 custom.cp.rp.label.envitem.unit=Unit
  
```

5.2.2 Détail du plan de collection

5.2.2.1 Présentation de l'écran

Mode Consultation :

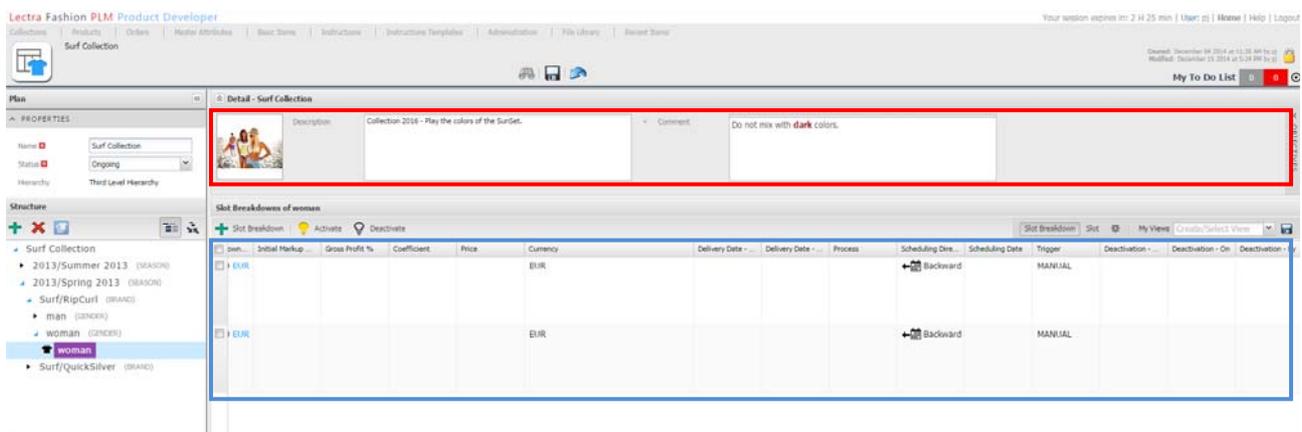


The screenshot shows the 'Detail - Surf Collection' page. The top section, highlighted with a red border, contains the following information:

- Description:** Collection 2016 - Play the colors of the SunSet.
- Comment:** Do not mix with dark colors.

Below this is a table titled 'Slot Breakdowns of women' with a blue border. The table has the following columns: Item, Initial Markup, Gross Profit %, Coefficient, Price, Currency, Delivery Date, Process, Scheduling Dir., Scheduling Date, Trigger, Deactivation, and Deactivation - On. Two rows are visible, both for 'EUR' currency and 'Backward' scheduling direction, with a 'MANUAL' trigger.

Mode Modification :



This screenshot is identical in layout to the consultation mode, but the fields within the red-bordered box are now input fields for editing. The 'Description' and 'Comment' fields contain the same text as in the consultation mode, but they are now ready to be modified.

Cet écran comporte deux parties personnalisables : les données générales (red framed (encadrées de rouge)) et la grille des déclinaisons (blue framed (encadrée de bleu)). Notez qu'il y a deux modes : consultation et modification.

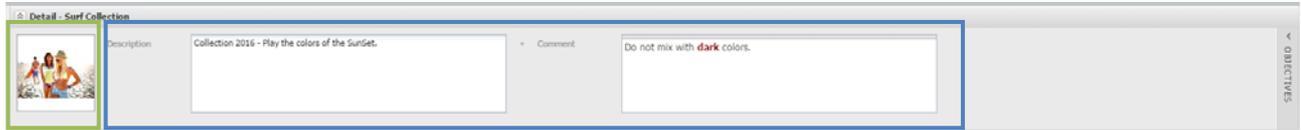
5.2.2.2 Fichier XML personnalisable

L'écran de liste du plan de collection peut être personnalisé en modifiant le fichier [Lectra.PDM.CollectionPlan.CollectionPlan.xml](#). Il est nécessaire de compléter le fichier se trouvant dans le sous-dossier **custom namespaces/custom/** [Lectra.PDM.CollectionPlan.CollectionPlan.xml](#) (et non dans l'original [namespaces/Lectra.PDM.CollectionPlan.CollectionPlan.xml](#))

S'il n'existe pas, il est nécessaire de le créer dans le dossier custom et de le structurer comme suit :

```
<namespace xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
uri="http://Lectra.com/pdm/collectionPlan/custom"
  prefix="collectionPlan">
  <!-- General Data Form - Display and Edit Mode -->
  <form name="detailsEditDefault" layout="border" id="editForm"
reference="collectionPlanDetailsForm" border="false">
  <panel inherits="detailsEdit/detailsEditWest" />
  <panel inherits="detailsEdit/detailsEditCenter" overflowX="auto" overflowY="auto"
layout="form">
    <!-- See Dictionnary.xml for the available kinds of fields -->
  </panel>
</form>
  <!-- Grid Range Plan Slot Breakdown -->
  <grid name="rangePlansLotbreakdowngridDefault" inherits="rangePlansLotbreakdowngrid" >
    <!-- See Dictionnary.xml for the available kinds of fields -->
  </grid>
</namespace>
```

5.2.2.3 Identification de la partie à modifier



La partie consacrée aux données générales possède la structure suivante :

detailsEdit (form)

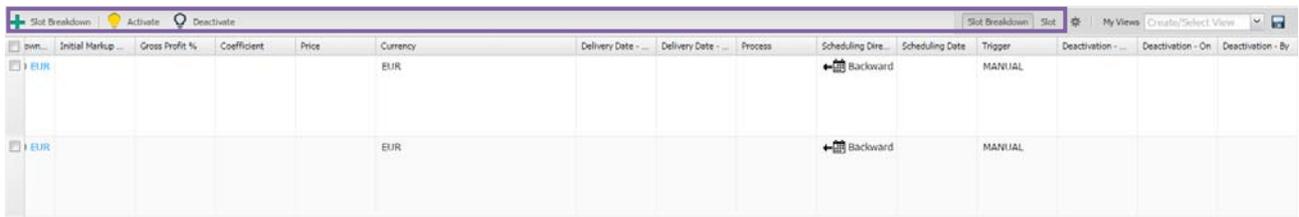
 |_ detailsEditWest (panel)

 |_ imagefield

 |_ detailsEditCenter (panel)

 |_ description (textarea)

 |_ comment (htmleditor)



Initial Markup	Gross Profit %	Coefficient	Price	Currency	Delivery Date	Delivery Date	Process	Scheduling Dire.	Scheduling Date	Trigger	Deactivation	Deactivation - On	Deactivation - By
				EUR				Backward		MANUAL			
				EUR				Backward		MANUAL			

La partie consacrée à la grille des déclinaisons possède la structure suivante :

rangeplanslotbreakdowngrid (rangeplangrid)

 |_ no name (toolbar containing buttons)

 |_ (column 1)

 |_ (column 2)

 |_ etc.

5.2.2.4 Ajout de champs sous imagefield de detailsEditWest (données générales)

Ajoutez des champs et des rôles que vous avez créés dans le module Administration et configuration, dans la balise panel :

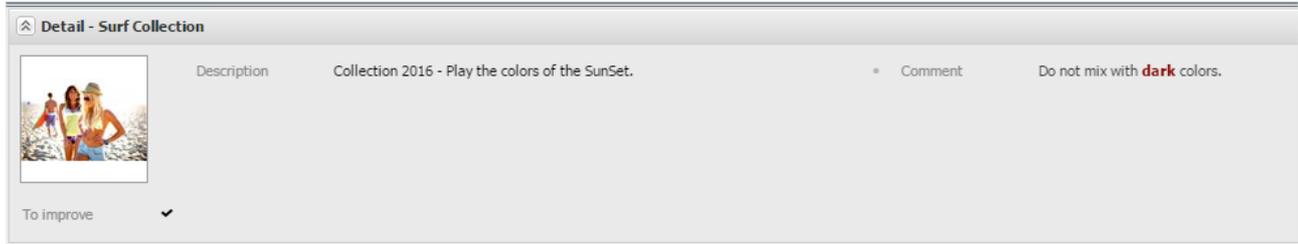
```
<panel inherits="detailsEdit/detailsEditWest" />
```

Et réactualisez votre explorateur pour voir la modification.

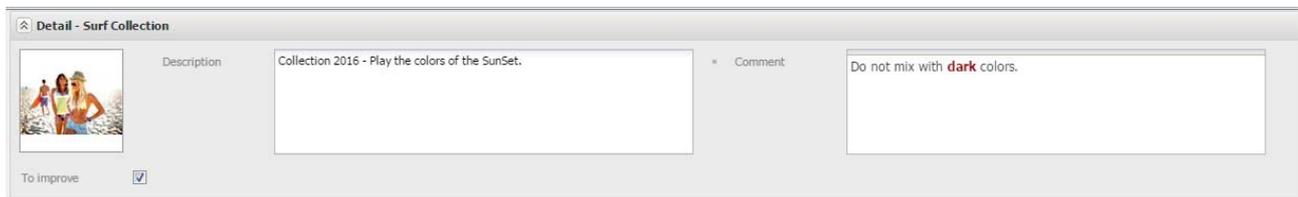
Exemple : case à cocher affichée sous l'image

```
<panel inherits="detailsEdit/detailsEditWest" >
  <checkbox name="CP_Boolean" fieldLabelI18n="custom.cp.label.boolean" margin="10 0 0 10">
    <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Boolean}" />
  </checkbox>
</panel>
```

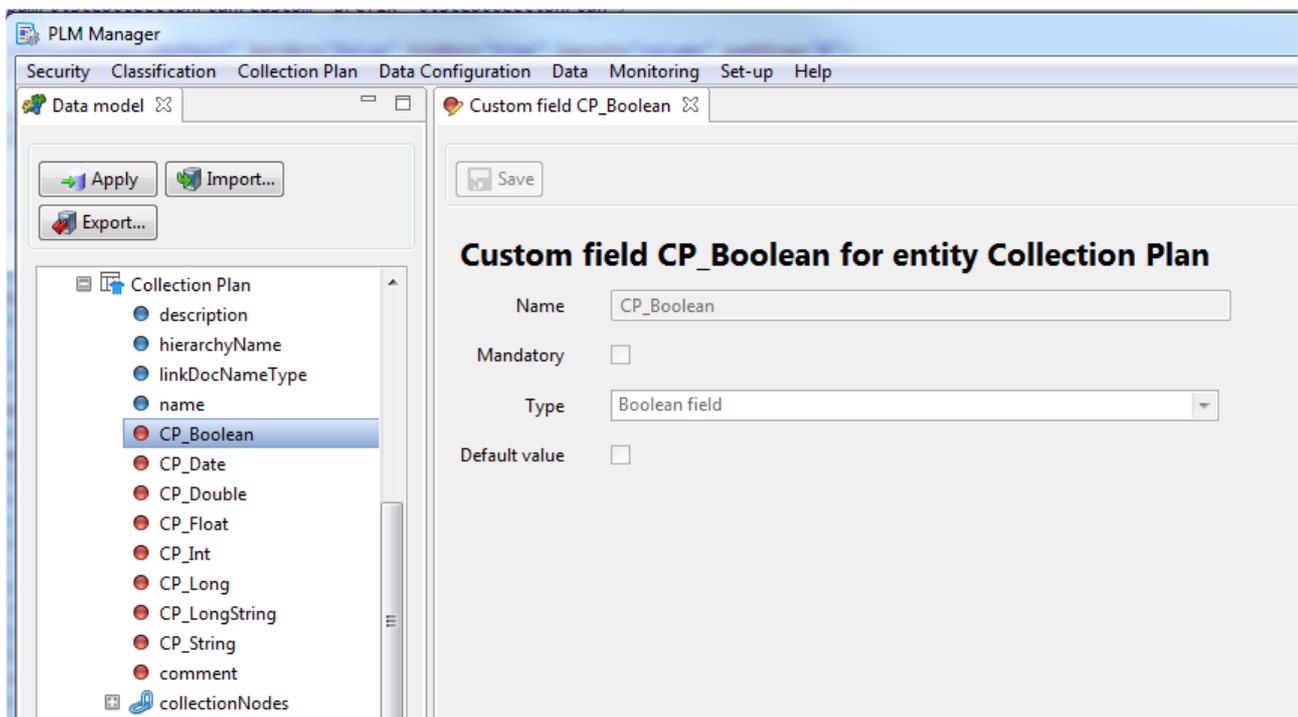
Mode Consultation :



Mode Modification :



Remarque : le nom de la case à cocher est identique à celui créé dans le module Administration et configuration :



Notez que l'attribut de la valeur *bind* est « {collectionPlan.created_name} ». Vous devez utiliser ce modèle.

Remarque : le passage du mode de consultation à celui de modification, et inversement, est conditionné par la balise *bind* avec l'attribut *readOnly*. Vous devez utiliser ce mécanisme.

5.2.2.5 Ajout de champs jusqu'à imagefield de « detailsEditWest » (données générales)

Si vous devez ajouter des champs et des rôles jusqu'à imagefield, vous ne pouvez plus utiliser le mécanisme d'héritage. Le panneau doit être redéfini en copiant celui du fichier d'origine namespaces/ Lectra.PDM.CollectionPlan.CollectionPlan.xml :

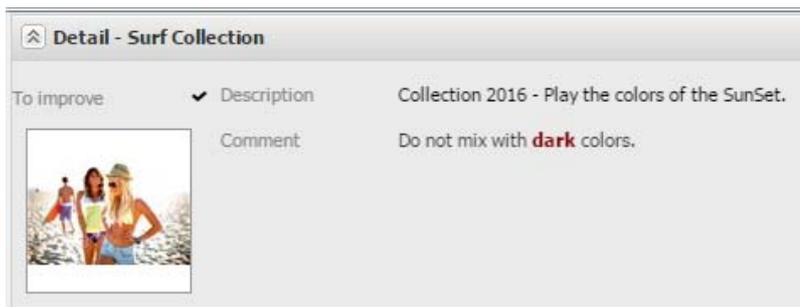
```
<panel name="detailsEditWest" border="false" region="west">
  <imagefield fieldLabelI18n="action.toggle.storyBoard" hideLabel="true" name="attachment"
    useREST="true" inherits="http://lectra.com/pdm/common#defaultImageFieldEXT5" managerRef="selSmallImageField"
    imageSize="medium" imageWidth="96" imageHeight="96" padding="8 0 0 8" >
    <managerInstance name="Lectra.PDM.Mgrs.imageFieldRestMgr" />
    <bind fileLinkObj="{collectionPlan.attachment}" readOnly="{!isPageEditable}"/>
  </imagefield>
</panel>
```

Collez-le ensuite dans le fichier personnalisé namespaces/custom/ Lectra.PDM.CollectionPlan.CollectionPlan.xml, à la place du panneau hérité, et modifiez-le comme il convient.

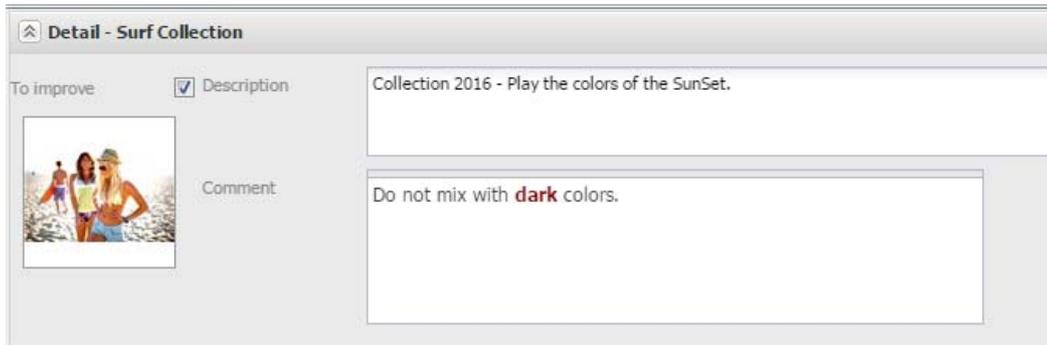
Exemple : case à cocher affichée jusqu'à l'image

```
<panel name="detailsEditWest" border="false" region="west">
  <checkbox name="CP_Boolean" fieldLabelI18n="custom.cp.label.boolean" margin="10 0 0 10">
    <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Boolean}"/>
  </checkbox>
  <imagefield fieldLabelI18n="action.toggle.storyBoard" hideLabel="true" name="attachment"
    useREST="true" inherits="http://lectra.com/pdm/common#defaultImageFieldEXT5" managerRef="selSmallImageField"
    imageSize="medium" imageWidth="96" imageHeight="96" padding="8 0 0 8" >
    <managerInstance name="Lectra.PDM.Mgrs.imageFieldRestMgr" />
    <bind fileLinkObj="{collectionPlan.attachment}" readOnly="{!isPageEditable}"/>
  </imagefield>
</panel>
```

Mode Consultation :



Mode Modification :



5.2.2.6 Ajout de champs sous commentaire et description de detailsEditCenter (données générales)

Ajoutez des champs et des rôles que vous avez créés dans le module Administration et configuration, dans la balise panel :

```
<panel inherits="detailsEdit/detailsEditCenter" overflowX="auto" overflowY="auto" layout="form">
```

Et réactualisez votre explorateur pour voir la modification.

Exemple : une zone de texte, une case à cocher, un nombre entier, une date, une combinaison multiple, une combinaison unique, une image et une combinaison envitem affichée sous commentaire et description

```
<panel inherits="detailsEdit/detailsEditCenter" overflowX="auto" overflowY="auto" layout="form">
  <!-- See Dictionary.xml for the available kinds of fields -->
  <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>
</panel>
```

Mode Consultation :

Detail - Surf Collection



Description: Collection 2016 - Play the colors of the SunSet.

Comment: Do not mix with **dark** colors.

Other comment: Do not forget to see with designers

To improve:

Prevision: 45600

End: 04/08/15

Countries: Italie
Espagne

Original country: France

Add idea: 

Mode Modification :

Detail - Surf Collection



Description:

Comment:

B **U**        

Do not mix with **dark** colors.

Other comment:

To improve:

Prevision:

End:

Countries:

Original country:

Add idea: 

Main color:

Notez que la disposition a changé : nous avons choisi une disposition de formulaire pour cet exemple, avec une barre de défilement, si nécessaire (attribut *overflow*).

5.2.2.7 Remarques relatives aux champs et aux rôles dans cet écran (partie consacrée aux données générales)

Voici différents champs et rôles (avec des exemples) que vous pouvez ajouter à cet écran. Ils sont affichés en mode consultation ou modification, selon l'action de l'utilisateur (cliquez sur le bouton Modifier) :

String field (Champ de chaîne)

```
<textfield name="CP_String" fieldLabelI18n="custom.cp.Label.string" maxLength="255">  
  <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_String}"/>  
</textfield>
```

Long String field (Champ de chaîne long)

```
<textarea name="CP_LongString" fieldLabelI18n="custom.cp.Label.Longstring" >  
  <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_LongString}"/>  
</textarea>
```

Boolean field (Champ booléen)

```
<checkbox name="CP_Boolean" fieldLabelI18n="custom.cp.Label.boolean">  
  <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Boolean}"/>  
</checkbox>
```

Long field (Long champ)

```
<longfield name="CP_Long" fieldLabelI18n="custom.cp.Label.Long" >  
  <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Long}"/>  
</longfield>
```

Integer field (Champ d'entier)

```
<integerfield name="CP_Int" fieldLabelI18n="custom.cp.Label.int" >  
  <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Int}"/>  
</integerfield>
```

Double field (Champ double)

```
<doublefield name="CP_Double" fieldLabelI18n="custom.cp.Label.double" >  
  <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Double}"/>  
</doublefield>
```

Float field (Champ flottant)

```
<floatfield name="CP_Float" fieldLabelI18n="custom.cp.Label.float" >  
  <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Float}"/>  
</floatfield>
```

Date field (Champ de date)

```
<datefield name="CP_Date" fieldLabelI18n="custom.cp.Label.date" >  
  <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Date}"/>  
</datefield>
```

HVL Pick list multi field (Champ multiple de liste de sélection HVL)

```
<nodecombo name="CP_Country_Multi" multiSelect="true" propertyName="Country"  
fieldLabelI18n="custom.cp.Label.hvl.country.multi" formatQuery="false" >  
  <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Country_Multi}"/>  
</nodecombo>
```

HVL Pick list single field (Champ unique de liste de sélection HVL)

```
<nodecombo name="CP_Country_Single" multiSelect="false" displayField="path"
propertyName="Country" fieldLabelI18n="custom.cp.Label.hvl.country.single"
formatQuery="false" >
    <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Country_Single}"/>
</nodecombo>
```

Sélection du document cible

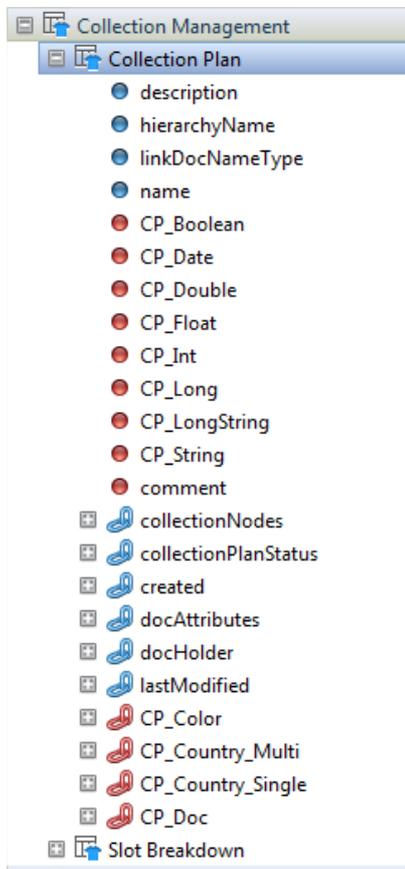
```
<imagefield name="CP_Doc" fieldLabelI18n="custom.cp.Label.document"
useREST="true" inherits="http://lectra.com/pdm/common#defaultImageFieldEXT5"
managerRef="selSmallImageField" imageWidth="96" imageHeight="96">
    <managerInstance name="Lectra.PDM.Mgrs.imageFieldRestMgr" />
    <bind readOnly="{!isPageEditable}" fileLinkObj="{collectionPlan.CP_Doc}" />
</imagefield>
```

Envitem field (Champ Envitem)

```
<envitemcombo name="CP_Unit" hiddenName="CP_Unit" envItemType="Unit"
hideTrigger2="true" fieldLabelI18n="custom.cp.Label.envitem.unit" pageSize="25"
formatQuery="false">
    <bind readOnly="{!isPageEditable}" value="{collectionPlan.CP_Unit}" />
</envitemcombo>
```

Notez que l'attribut `formatQuery="false"` est nécessaire pour permettre la recherche sur les champs de liste.

Notez que l'attribut de la valeur `name` est identique à celui créé dans le module Administration et configuration :



Notez que l'attribut de la valeur *fieldLabelI18n* doit être défini dans le fichier PLM-Fashion\PDM\Lectra\PLMParam\messagesCusto_en.properties (et dans un autre fichier de langue, si nécessaire ; exemple messagesCusto_fr.properties pour le français)

5.2.2.8 Ajout de colonnes après les colonnes d'origine de la grille des déclinaisons (partie consacrée à la grille)

Ajoutez des champs et des rôles que vous avez créés dans le module Administration et configuration, dans la balise grid :

```
<!-- Grid Range Plan Slot Breakdown -->
<grid name="rangeplanslotbreakdowngridDefault" inherits="rangeplanslotbreakdowngrid" >

    <!-- See Dictionary.xml for the available kinds of fields -->

</grid>
```

Et réactualisez votre explorateur pour voir la modification.

Exemple : une colonne de date, une colonne de case à cocher, une colonne de texte, une colonne de combinaison multiple, une colonne de combinaison unique, une colonne d'image, une colonne de combinaison envitem d'unité et une colonne de combinaison envitem de couleur marketing affichés dans la grille des déclinaisons.

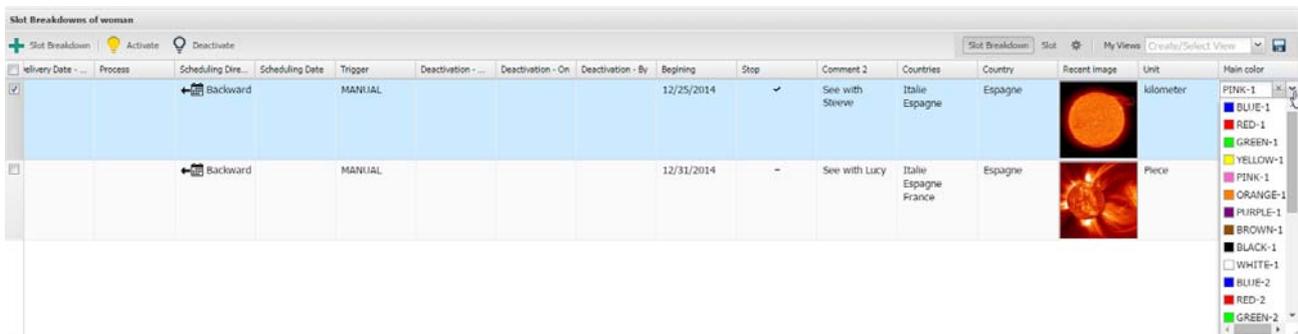
```
<!-- Grid Range Plan Slot Breakdown -->
<grid name="rangeplanslotbreakdowngridDefault" inherits="rangeplanslotbreakdowngrid">
  <column xtype="datecolumn" dataIndex="CP_RP_Date" headerI18n="custom.cp.rp.label.date" format="{#{i18n['i18n.extdateformat']}" editable="true">
    <editorField xsi:type="datefield" 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" massEditable="false">
    <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>
  <column xsi:type="envitemcolumn" envItemType="Unit" headerI18n="custom.cp.rp.label.envitem.unit" dataIndex="CP_RP_Unit"
  headerId="CP_RP_Unit" sortable="false" editable="true" formatQuery="false">
    <rendererFn name="Lectra.PDM.Rest.Format.otherEnvItemRenderer(false)" />
    <editorConfig>
      <listConfig cls="pdm-collectionplan-rangeplanelditor" resizable="true" />
    </editorConfig>
  </column>
  <column xsi:type="envitemcolumn" envItemType="MarketingColor" headerI18n="custom.cp.rp.label.envitem.color" dataIndex="CP_RP_Color"
  headerId="CP_RP_Color" editable="true" formatQuery="false">
    <rendererFn name="Lectra.PDM.Rest.Format.colorEnvItemRenderer(false)" />
    <editorConfig>
      <listConfig cls="pdm-collectionplan-rangeplanelditor"
        resizable="true" />
      <store pageSize="25">
        <dataField name="name" type="string" />
        <proxy xsi:type="rest" type="rest" url="/marketing-colors">
          <extraParams restrequest="true" />
          <reader type="json" />
        </proxy>
      </store>
    </editorConfig>
  </column>
</grid>
```

Mode Consultation :



Delivery Date	Process	Scheduling Dire...	Scheduling Date	Trigger	Deactivation - ...	Deactivation - On	Deactivation - By	Beginning	Stop	Comment 2	Countries	Country	Recent image	Unit	Main color
	Backward			MANUAL				12/25/2014	✓	See with Steeve	Italie Espagne	Espagne		kilometer	PINK-1
	Backward			MANUAL				12/31/2014	-	See with Lucy	Italie Espagne France	Espagne		Piece	RED-2

Mode Modification :



Delivery Date	Process	Scheduling Dire...	Scheduling Date	Trigger	Deactivation - ...	Deactivation - On	Deactivation - By	Beginning	Stop	Comment 2	Countries	Country	Recent image	Unit	Main color
	Backward			MANUAL				12/25/2014	✓	See with Steeve	Italie Espagne	Espagne		kilometer	PINK-1
	Backward			MANUAL				12/31/2014	-	See with Lucy	Italie Espagne France	Espagne		Piece	RED-2

5.2.2.9 Remarques relatives aux champs et aux rôles dans cette grille des déclinaisons (partie consacrée à la grille)

Voici différents champs et rôles (avec des exemples) que vous pouvez ajouter à cette grille. Ils sont affichés en mode consultation ou modification, selon l'action de l'utilisateur (cliquez sur le bouton Editer) :

Date field in grid (Champ de date dans la grille)

```
<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>
```

Boolean field in grid (Champ booléen dans la grille)

```
<column xtype="checkboxcolumn" dataIndex="CP_RP_Boolean"
headerI18n="custom.cp.rp.Label.boolean" headerId="CP_RP_Boolean" editable="true"/>
```

String field in grid (Champ de chaîne dans la grille)

```
<column xtype="textcolumn" dataIndex="CP_RP_String"
headerI18n="custom.cp.rp.Label.string" headerId="CP_RP_String" editable="true"/>
```

Liste de sélection HVL dans la grille : exemple avec Pays et Sélection multiple

```
<column dataIndex="CP_RP_Country_Multi" headerI18n="custom.cp.rp.Label.hvl.multi"
headerId="CP_RP_Country_Multi" massEditable="false" >
  <editorField xsi:type="nodecombo" multiSelect="true" propertyName="Country"
formatQuery="false" />
  <rendererFn name="Lectra.PDM.Format.nodesRenderer()" />
</column>
```

Liste de sélection HVL dans la grille : exemple avec Pays et Sélection unique

```
<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>
```

Document cible dans la grille

```
<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">
    <managerInstance name="Lectra.PDM.Mgrs.imageFieldRestMgr" />
  </imageEditor>
</column>
```

EnvItem (sélection multiple interdite) dans la grille : exemple avec Unité

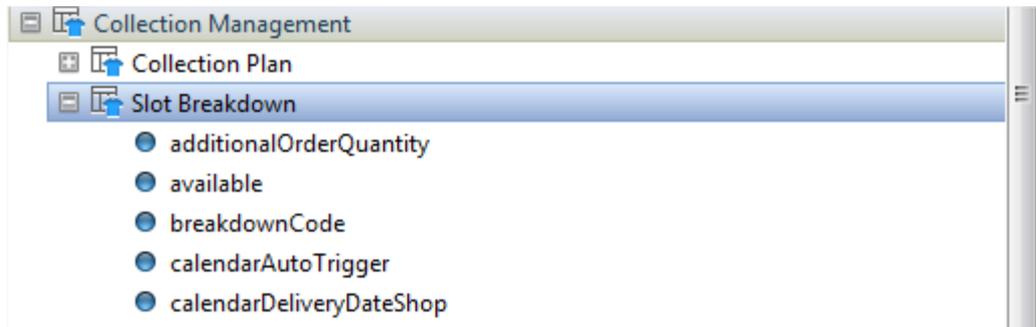
```
<column headerI18n="custom.cp.rp.Label.envitem.unit" xsi:type="envitemcolumn"
envItemType="Unit" dataIndex="CP_RP_Unit" headerId="CP_RP_Unit" sortable="false"
editable="true" formatQuery="false">
  <rendererFn name="Lectra.PDM.Rest.Format.otherEnvItemRenderer(false)" />
  <editorConfig>
    <listConfig cls="pdm-collectionplan-rangeplaneditor" resizable="true" />
  </editorConfig>
</column>
```

EnvItem (sélection multiple interdite) dans la grille avec Couleur marketing uniquement

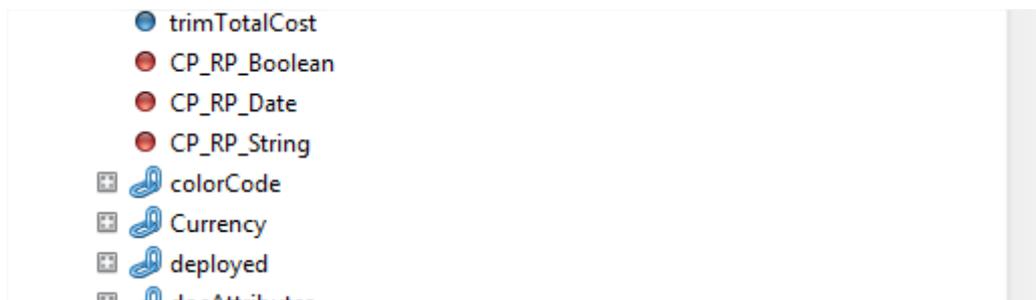
```
<column headerI18n="custom.cp.rp.Label.envitem.color" xsi:type="envitemcolumn"
envItemType="MarketingColor" dataIndex="CP_RP_Color" headerId="CP_RP_Color"
editable="true" formatQuery="false">
  <rendererFn name="Lectra.PDM.Rest.Format.colorEnvItemRenderer(false)" />
  <editorConfig>
    <listConfig cls="pdm-collectionplan-rangeplaneditor"
      resizable="true" />
    <store pageSize="25">
      <dataField name="name" type="string" />
      <proxy xsi:type="rest" type="rest" url="/marketing-colors">
        <extraParams restrequest="true" />
        <reader type="json" />
      </proxy>
    </store>
  </editorConfig>
</column>
```

Notez que l'attribut `formatQuery="false"` est nécessaire pour permettre la recherche sur les champs de liste.

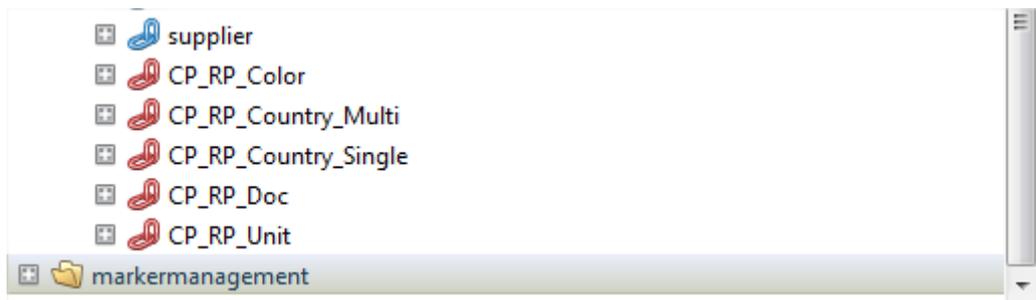
Notez que l'attribut de la valeur `dataIndex` est identique à celui créé dans le module Administration et configuration :



...



...



Notez que l'attribut de la valeur *header18n* doit être défini dans le fichier [PLM-Fashion\PDM\LectraPLMParam\messagesCusto_en.properties](#) (et dans un autre fichier de langue, si nécessaire ; exemple [messagesCusto_fr.properties](#) pour le français)

Notez que *marketingColor envitem* possède une balise store en fonction de l'url `/marketing-colors` ; ceci est nécessaire pour afficher les vignettes de couleur.

6. CONFIGURATION AVANCEE POUR L'AJOUT DES CHAMPS ET DES PANNEAUX

6.1 Utilisation de l'héritage

Au lieu de copier des descriptions de panneaux entières du fichier XML de base dans le dossier « custom » du fichier XML, il est possible, si vous êtes sûr de la description XML, d'utiliser le concept d'héritage pour décrire sa description personnalisée.

L'héritage est défini dans la section [2.2 - Héritage](#).

Toute déclaration d'élément graphique comme <panel> peut être appliquée à d'autres définitions tout en conservant la définition par défaut existant dans le fichier.

Si l'héritage correspondant à **#type** ou **#topCategoryName** est déjà défini ou ne répond pas à vos besoins, il est possible d'étendre l'héritage pour la définition d'un nouveau panneau à prendre en compte de manière prioritaire.

6.2 Exemple spécifique

La description du panneau hérite de 3 définitions de description de panneaux possibles :

```
<panel name="description"
inherits="description_#{@type},description_#{@topCategoryName},description" />
```

Si le panneau que vous voulez appliquer est différent de <panel name="description_tous_les_types">, différent de <panel name="description_Styles, Fabrics, Trims, PackagingLabel"> :

- <panel name= « MaDescription_Custom »> peut être créé
- le nouvel héritage peut être ajouté à la déclaration de la description du panneau ci-dessus.

```
<panel name="description"
inherits="MaDescription_Custom" description_#{@type},description_#{@topCategory
Name},description" />
```

En première position, il sera recherché en premier, avant la description par **type** et **topCategoryName**.

Il est inutile de dupliquer le code XML : il suffit d'ajouter le code de

```
<panel name="MaDescription_Custom" .../... />
```

Constraint (Contrainte) :

En cas d'héritage d'une définition de l'Explorateur, il est nécessaire de définir la balise <contextparameter> suivante :

```
<panel name="Explorer_CustomStyle" inherits="Explorer">
  <contextparameter name="explorerTitle" value="model.Style" />
  <contextparameter name="categoryName" value="CustomStyle" />
  <contextparameter name="topCategoryName" value="Style" />
  <contextparameter name="explorerName" value="Style" />
</panel>
```

7. MODIFICATION DE LA MISE EN PAGE

La configuration des écrans peut également être effectuée en modifiant la présentation des écrans.

Pour cela, modifiez les mises en pages utilisées. Cela implique cependant d'apporter des modifications aux définitions XML de écrans et de respecter une grammaire XML avancée.

Les éléments graphiques utilisés dans notre grammaire XML proviennent de la bibliothèque ExtJS. Lectra a également développé la structure LPFExt qui définit les autres éléments graphiques qui héritent de la structure ExtJS.

Dans les paragraphes suivants, certains éléments de la bibliothèque ExtJS ou de la structure LPFExt seront mentionnés.

7.1 Définition de la mise en page

La mise en page définit le mode d'organisation des différents blocs de données à l'écran : dans les colonnes, dans les tableaux, dans les grilles, etc.

Suivez ce lien pour voir quelques [Exemples de mise en page](#).

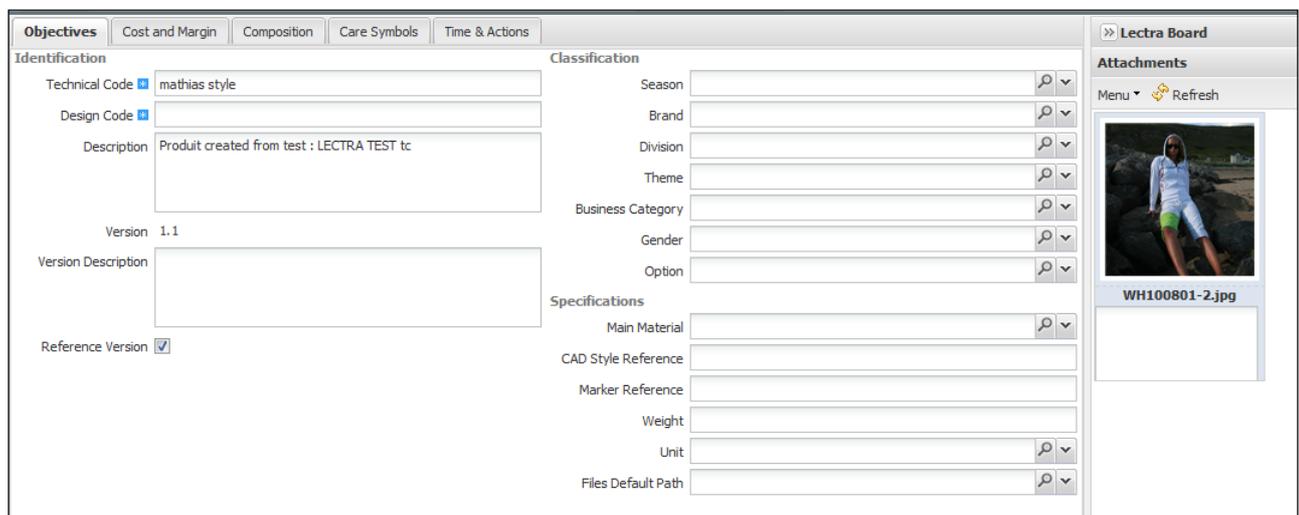
7.2 Seule la structure ExtJS **Erreur ! Source du renvoi introuvable.** possède une « ! layout (mise en page) »

Les conteneurs utilisés dans les écrans étudiés (dans le paragraphe précédent) sont « panels », « tabpanels », « gridpanels ». Pour obtenir la liste complète des conteneurs, consultez l'Annexe **Erreur ! Source du renvoi introuvable.** - **Erreur ! Source du renvoi introuvable.**

7.2.1 Mise en page des colonnes du panneau et des panneaux de l'écran GO d'un Style

Déjà mentionnés et abordés dans les paragraphes précédents, les panneaux sont des blocs de données couramment utilisés dans les écrans.

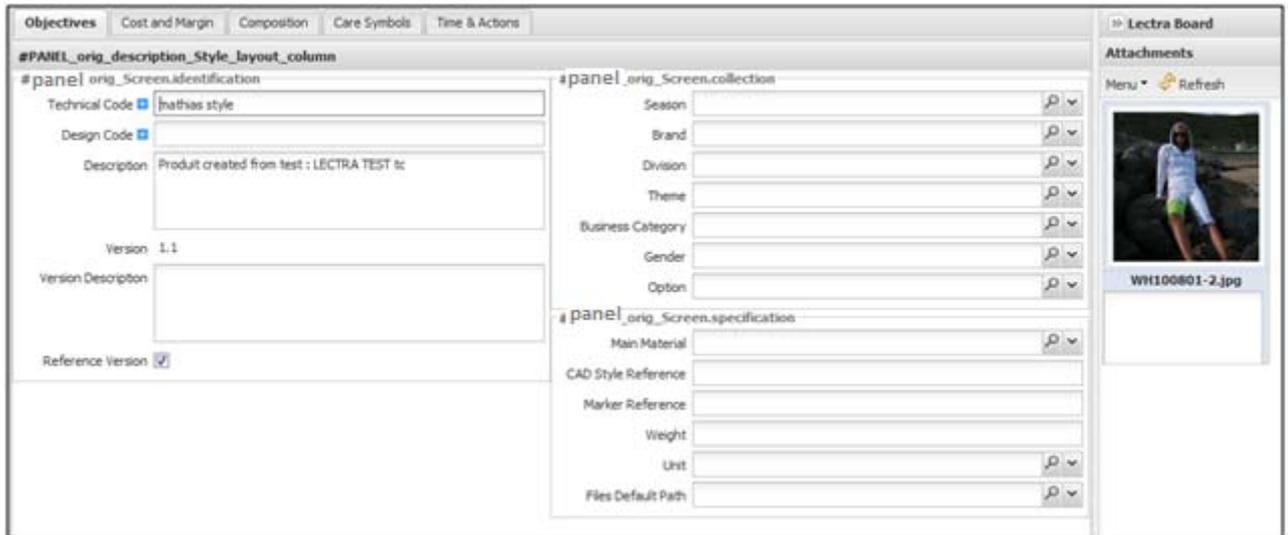
La modification des panneaux est illustrée dans les 2 captures d'écran de l'écran « style » GeneralObjectives. La visualisation des panneaux est donc plus facile.



The screenshot shows the 'Objectives' tab in the Lectra Board software. The interface is divided into several sections:

- Identification:**
 - Technical Code: mathias style
 - Design Code: (empty)
 - Description: Produit created from test : LECTRA TEST tc
 - Version: 1.1
 - Version Description: (empty)
 - Reference Version:
- Classification:**
 - Season: (dropdown)
 - Brand: (dropdown)
 - Division: (dropdown)
 - Theme: (dropdown)
 - Business Category: (dropdown)
 - Gender: (dropdown)
 - Option: (dropdown)
- Specifications:**
 - Main Material: (dropdown)
 - CAD Style Reference: (text input)
 - Marker Reference: (text input)
 - Weight: (text input)
 - Unit: (dropdown)
 - Files Default Path: (dropdown)

On the right side, there is a 'Lectra Board' header, an 'Attachments' section with a 'Menu' dropdown and a 'Refresh' button, and a preview image of a person in a white outfit with the filename 'WH100801-2.jpg'.



Le panneau de l'onglet **Objectives** est défini par la mise en page en **column (colonne)**,

```
<panel name="description_Style" titleI18n="PANEL orig description_Style layout column" layout="column" autoScroll="true" border="true">
```

... et les 3 panneaux qui le composent sont définis avec l'attribut `columnWidth="0.5"`, qui signifie qu'ils occupent chacun 50 % de la largeur principale du panneau.

```
<panel name="identification" titleI18n="fieldset orig Screen.identification" columnWidth="0.5" border="true">
```

```
<panel name="collection" titleI18n="fieldset orig Screen.collection" columnWidth="0.5" border="true">
<panel name="collection" titleI18n="fieldset orig Screen.collection" columnWidth="0.5" border="true">
```

Dans le module de développement produit, les attributs de mise en page les plus fréquemment utilisés sont la marge et la largeur.

ANNEXES

A. ExtJS 3.4 et ExtJS 5.0.1

PDM V4R2 utilise ExtJS 3.4, sauf pour les écrans de plan de collection, de validation de table, les onglets Composition et Symbole d'entretien qui sont basés sur ExtJS 6.0.2.

La couche d'abstraction LPFExt lui permet d'être partiellement indépendant de la version d'ExtJS utilisée.

Dans le fichier xml de description LPFExt, les attributs d'une balise sont interprétés par LPFExt ou ExtJS.

Pour ce qui est des attributs ExtJS, certains sont compatibles uniquement avec ExtJS 3.4, et d'autres avec aExtJS 3.4 et 6.0.2.

Exemple :

```
<displayfield fieldLabelI18n="objective" topLabelCls="" labelAlign="top" labelSeparator=":" bind="{objectives.processDevelopmentType}" />
```

- fieldLabelI18n : interprété par la couche LPFExt
- labelAlign et labelSeparator : interprétés par ExtJS 3.4 ou ExtJS 6.
- topLabelCls : interprété par ExtJS 6.

Avec l'instruction suivante

```
<displayfield fieldLabelI18n="objective" topLabelCls="" labelAlign="top" labelSeparator=":" bind="{objectives.processDevelopmentType}" />
```

Est totalement interprété par un écran du plan de collection (mode ExtJS 5).

Dans les autres écrans (mode ExtJS 3.4), l'attribut topLabelCls est ignoré.

B. Bascule entre ExtJS 3.4 et ExtJS 6.0.2

Le Product Developer gère la cohabitation entre ExtJS 3.4 et ExtJS 6. Selon l'url traitée par le serveur, le système bascule entre le mode ExtJS 6 et le mode ExtJS 3.4 :

- Url du plan de collection → Ext 6 mod.
- Autre URL : Ext 3 mod

Si vous voulez explicitement utiliser la bibliothèque ExtJS 5 (et supérieur) sur ExtJS 3.4 mod, vous devez appliquer un suffixe au nom du panneau par EXT5.

Exemple :

```
<!-- BEGIN WIDGET EXT5 DEV MODE -->
<panel name="detail_Panel_PickerEXT5">
  <managerInstance name="Lectra.PDM.Mgrs.aiPickerMgr"/>
  <graphicobjectsviewer name="localgr" width="300" itemId="bdetail"
titleI18n="action.filelink.info" managerRef="boardAIDetail"
inherits="http://lectra.com/pdm/picker#graphicContainer">
  <managerInstance name="Lectra.PDM.Mgrs.aiPickerMgr"/>
  </graphicobjectsviewer>
  <panel itemId="other" managerRef="emptyDetail" border="false"/>
</panel>
```

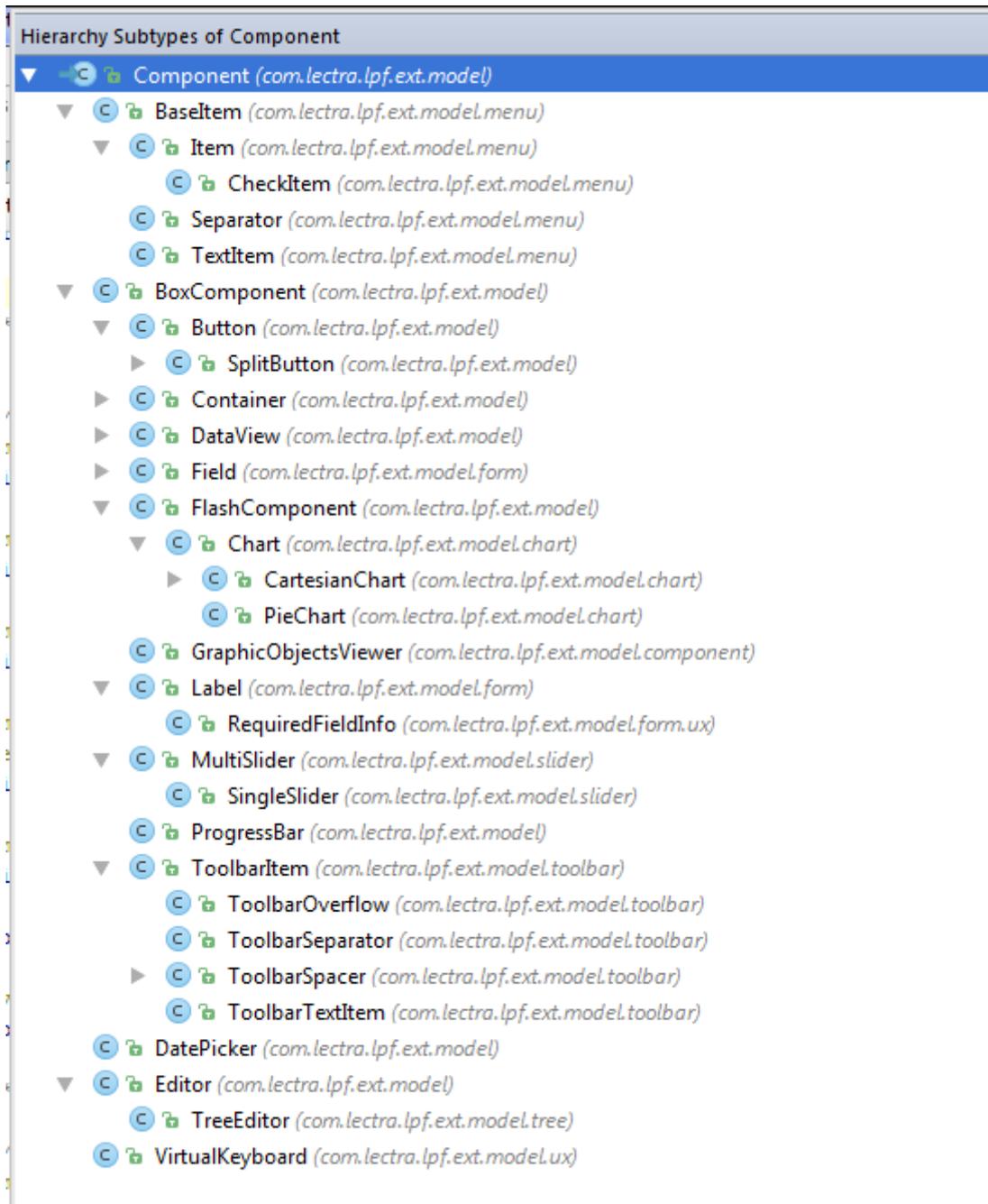
C. Liste des widgets de LPFExt

Un widget LPFExt est un composant graphique qui peut être créé à l'aide de la syntaxe xml LPFExt.

Exemple de widget ExplorerGrid :

```
<explorerGrid modePick="false"
inherits="http://lectra.com/pdm/search/product#searchGrid">
  <title>${#i18n['search.result']}</title>
</explorerGrid>
```

La liste des widgets LPFExt est la suivante :

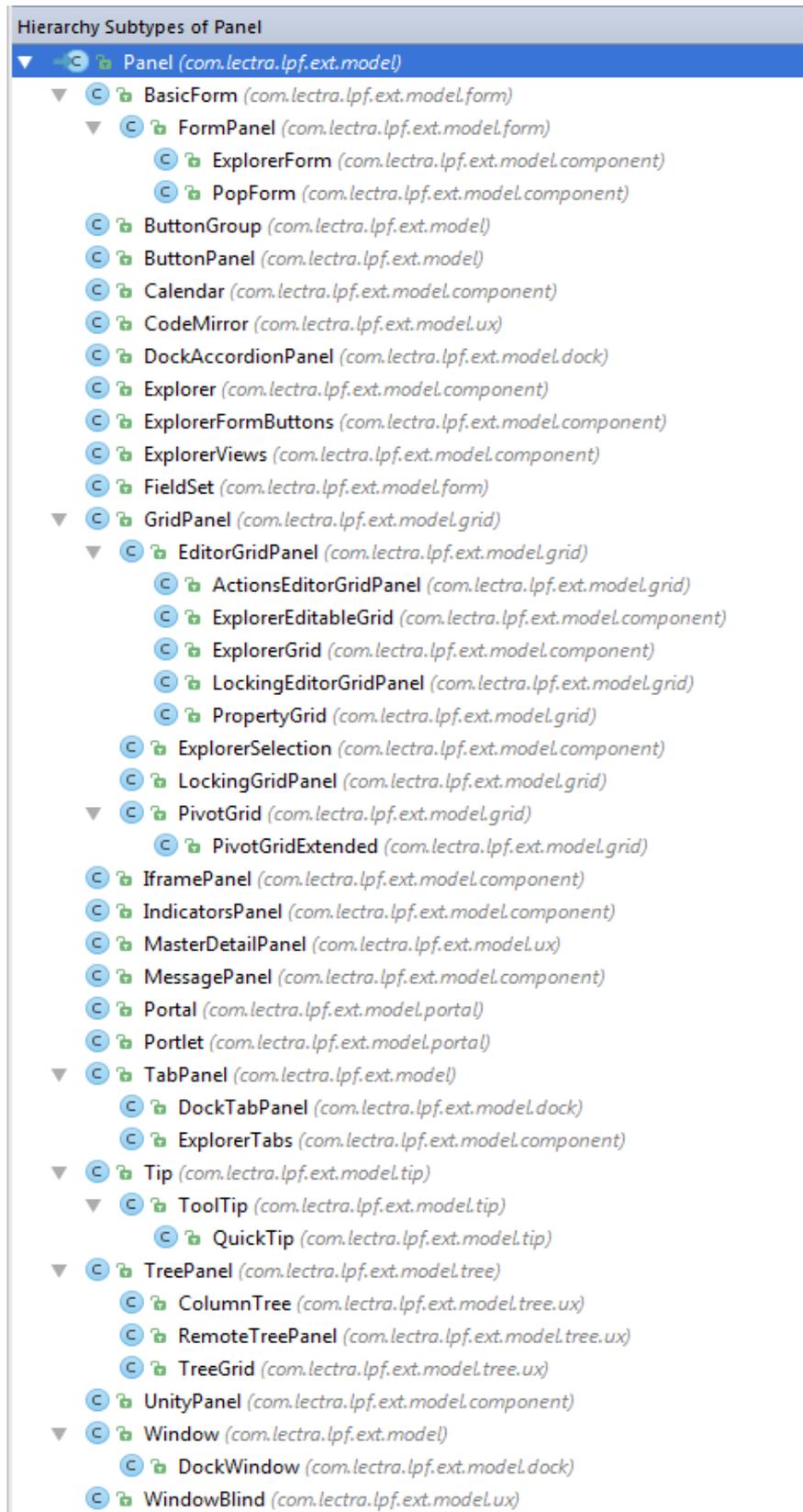


Détails des sous-composants du conteneur :

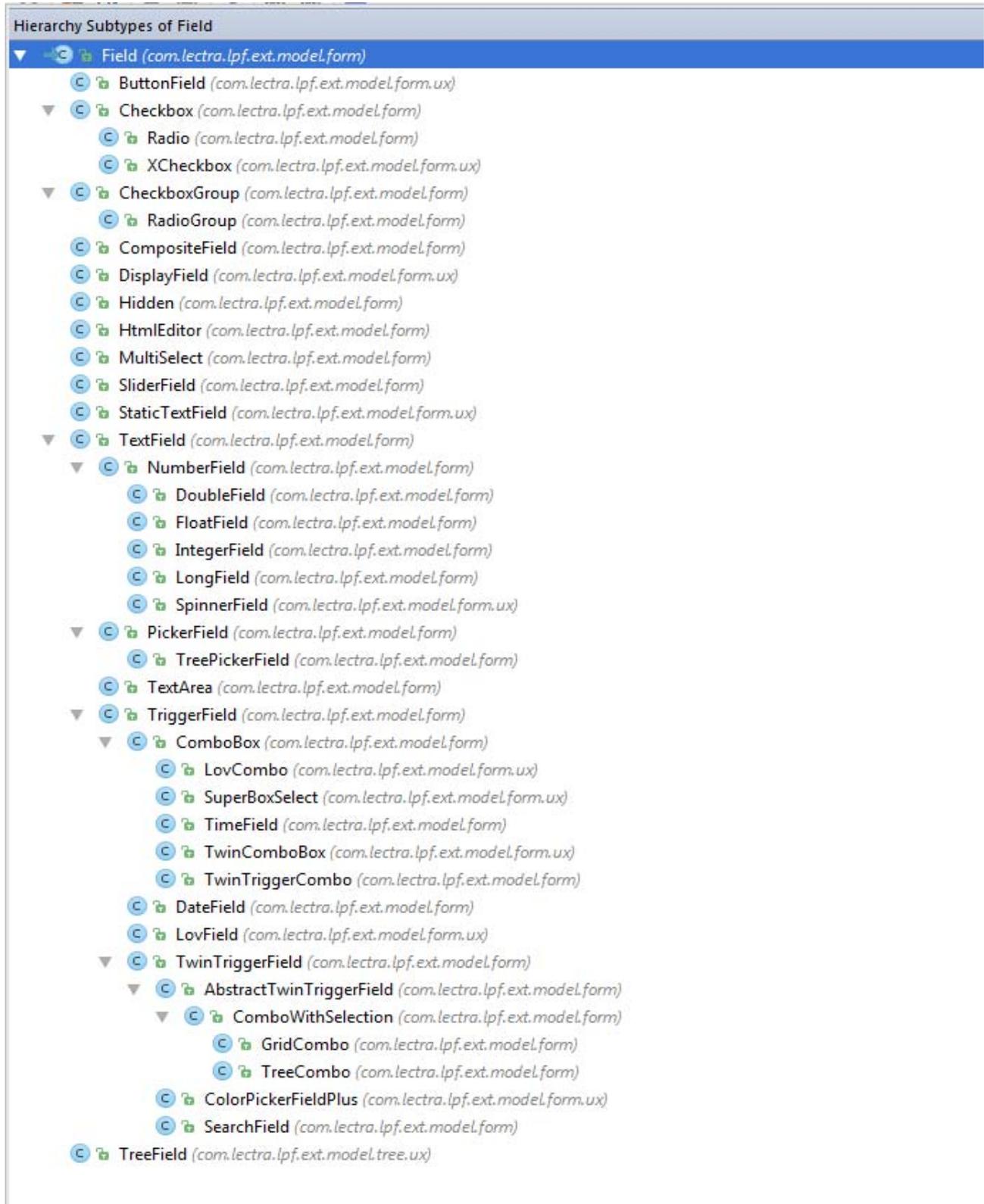
Hierarchy Subtypes of Container

- ▼ Container (*com.lectra.lpf.ext.model*)
 - ▼ FieldContainer (*com.lectra.lpf.ext.model.form*)
 - MultiSelectExt5 (*com.lectra.lpf.ext.model.form*)
 - GroupComponent (*com.lectra.lpf.ext.model*)
 - ▼ Menu (*com.lectra.lpf.ext.model.menu*)
 - StoreMenu (*com.lectra.lpf.ext.model.menu.ux*)
 - ▼ Panel (*com.lectra.lpf.ext.model*)
 - ▼ BasicForm (*com.lectra.lpf.ext.model.form*)
 - FormPanel (*com.lectra.lpf.ext.model.form*)
 - ButtonGroup (*com.lectra.lpf.ext.model*)
 - ButtonPanel (*com.lectra.lpf.ext.model*)
 - Calendar (*com.lectra.lpf.ext.model.component*)
 - CodeMirror (*com.lectra.lpf.ext.model.ux*)
 - DockAccordionPanel (*com.lectra.lpf.ext.model.dock*)
 - Explorer (*com.lectra.lpf.ext.model.component*)
 - ExplorerFormButtons (*com.lectra.lpf.ext.model.component*)
 - ExplorerViews (*com.lectra.lpf.ext.model.component*)
 - FieldSet (*com.lectra.lpf.ext.model.form*)
 - GridPanel (*com.lectra.lpf.ext.model.grid*)
 - IframePanel (*com.lectra.lpf.ext.model.component*)
 - IndicatorsPanel (*com.lectra.lpf.ext.model.component*)
 - MasterDetailPanel (*com.lectra.lpf.ext.model.ux*)
 - MessagePanel (*com.lectra.lpf.ext.model.component*)
 - Portal (*com.lectra.lpf.ext.model.portal*)
 - Portlet (*com.lectra.lpf.ext.model.portal*)
 - TabPanel (*com.lectra.lpf.ext.model*)
 - Tip (*com.lectra.lpf.ext.model.tip*)
 - TreePanel (*com.lectra.lpf.ext.model.tree*)
 - UnityPanel (*com.lectra.lpf.ext.model.component*)
 - Window (*com.lectra.lpf.ext.model*)
 - WindowBlind (*com.lectra.lpf.ext.model.ux*)
 - PanelHeader (*com.lectra.lpf.ext.model*)
 - PortalColumn (*com.lectra.lpf.ext.model.portal*)
 - ▼ Toolbar (*com.lectra.lpf.ext.model.toolbar*)
 - Paging (*com.lectra.lpf.ext.model.toolbar*)
 - PagingToolbar (*com.lectra.lpf.ext.model.toolbar*)
 - StatusBar (*com.lectra.lpf.ext.model.toolbar*)
 - ViewPort (*com.lectra.lpf.ext.model*)

Détails des sous-composants du panneau :

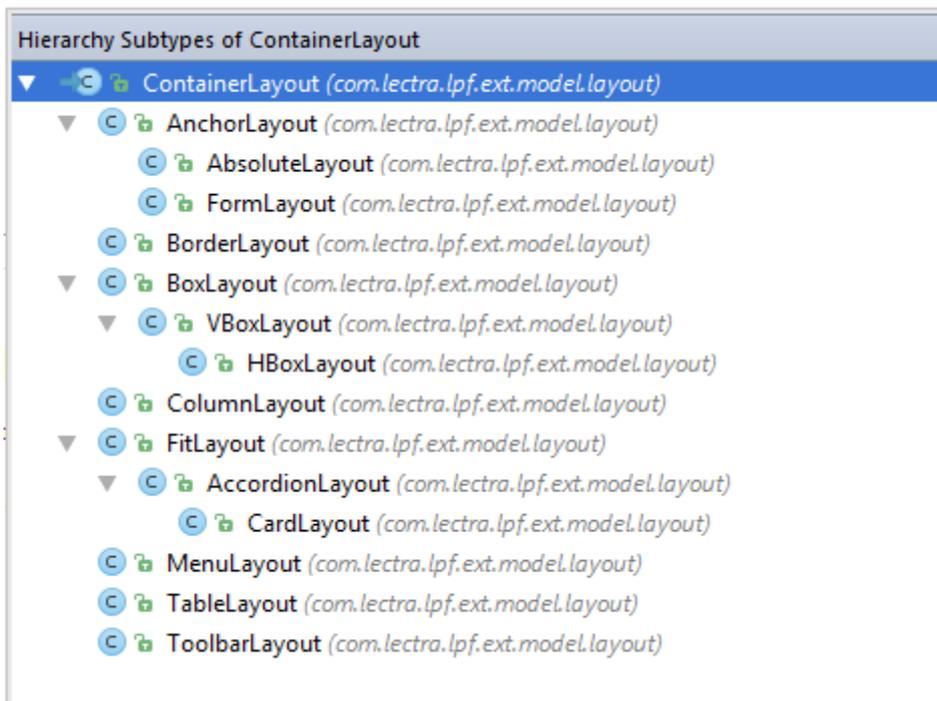


Détails des sous-composants du champ :

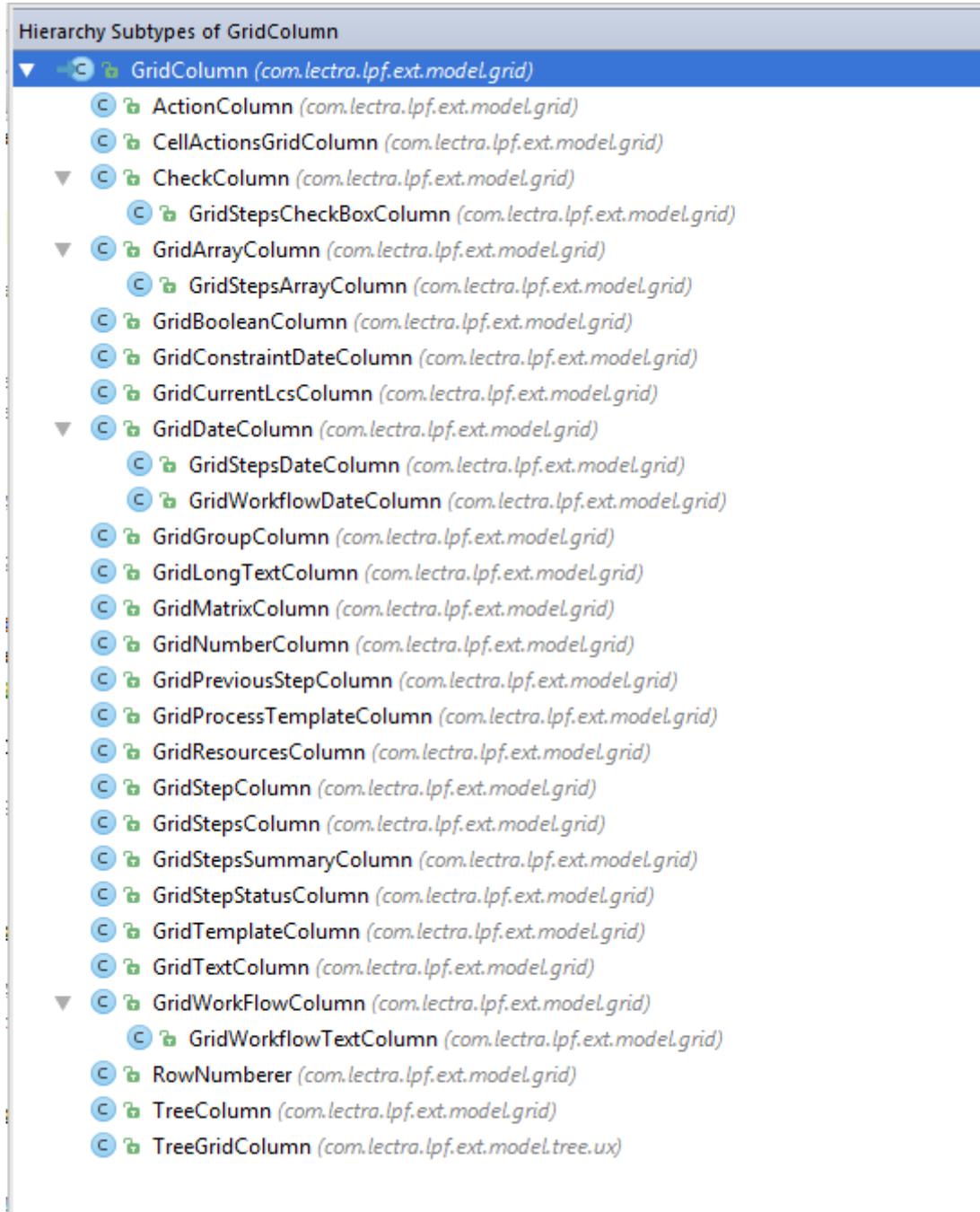


Certains widgets correspondent exactement au widget ExtJS (comme un bouton), alors que d'autres non (comme TreePickerField).

D. "Mises en page" LPFExt



E. Différents types de colonnes d'une grille LPFExt



F. Documentation ExtJS complète

- Pour ExtJS 3.4 : <http://docs.sencha.com/ext-js/3-4/#!/api>
- Pour ExtJS 5.0.1 : <http://docs-origin.sencha.com/extjs/5.0/5.0.1-apidocs/>

et exemples

- Pour ExtJS 3.4 : <http://dev.sencha.com/deploy/ext-3.4.0/examples/>
- Pour ExtJS 6.0.2 : <http://examples.sencha.com/extjs/6.0.2/examples/>

G. Documentations des widgets de LPFExt

Deux cas à prendre en compte :

1. les widgets LPFExt qui correspondent à un composant de ExtJS
2. les widgets LPFExt qui ne correspondent pas à un composant de ExtJS

Cas 1 :

Lisez la documentation du widget ExtJS correspondant.

Exemple : la documentation du widget Button LPFExt,

- Pour ExtJS 6 : <http://docs.sencha.com/extjs/6.0/6.0.2-classic/#!/api/Ext.button.Button>
- Pour ExtJS 3.4 : <http://docs.sencha.com/extjs/3.4.0/#!/api/Ext.Button>

Cas 2 :

Lisez la documentation de ExtJS du composant parent de LPFExt qui correspond au composant de ExtJS.

Exemple :_documentation du widget de FloatField LPFExt.

FloatField n'a pas d'équivalent ExtJS. Le parent de FloatField est NumberField, qui possède un équivalent ExtJS. Lisez donc la documentation de NumberField ExtJS.

- Pour ExtJS 5.0.1 :
 - <http://docs.sencha.com/extjs/6.0/6.0.2-classic/#!/api/Ext.form.field.Number>
- Pour ExtJS 3.4 :
 - <http://docs.sencha.com/extjs/3.4.0/#!/api/Ext.form.NumberField>

Pour finir, vous pouvez observer le fichier xml de l'écran afin de savoir comment un widget est configuré. Il y a de nombreux exemples d'utilisation.

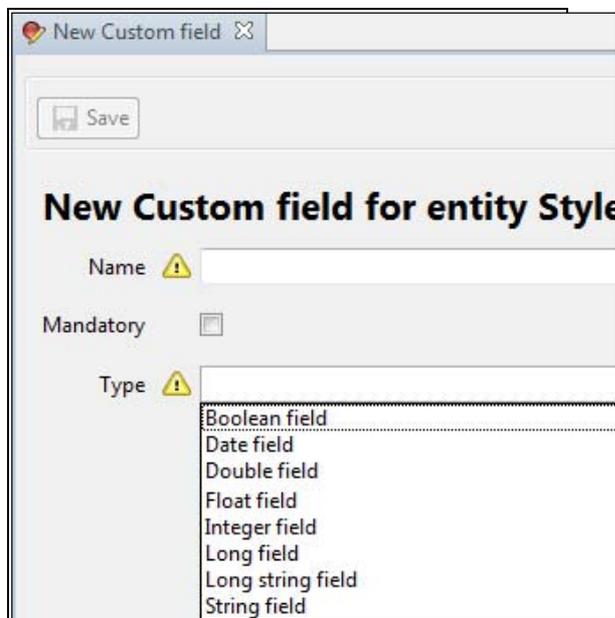
H. Changement d'emplacement du répertoire de LPFExt

Dans PLM V5R1, l'emplacement des fichiers LPFExt est sous lpf/ext3/nom du répertoire.

I. Rappel de l'application Administration et configuration

Pour compléter les nouveaux champs à ajouter à un écran avec les valeurs, il est nécessaire de créer les valeurs correspondantes dans la base de données.

Les champs du type suivant peuvent être ajoutés:

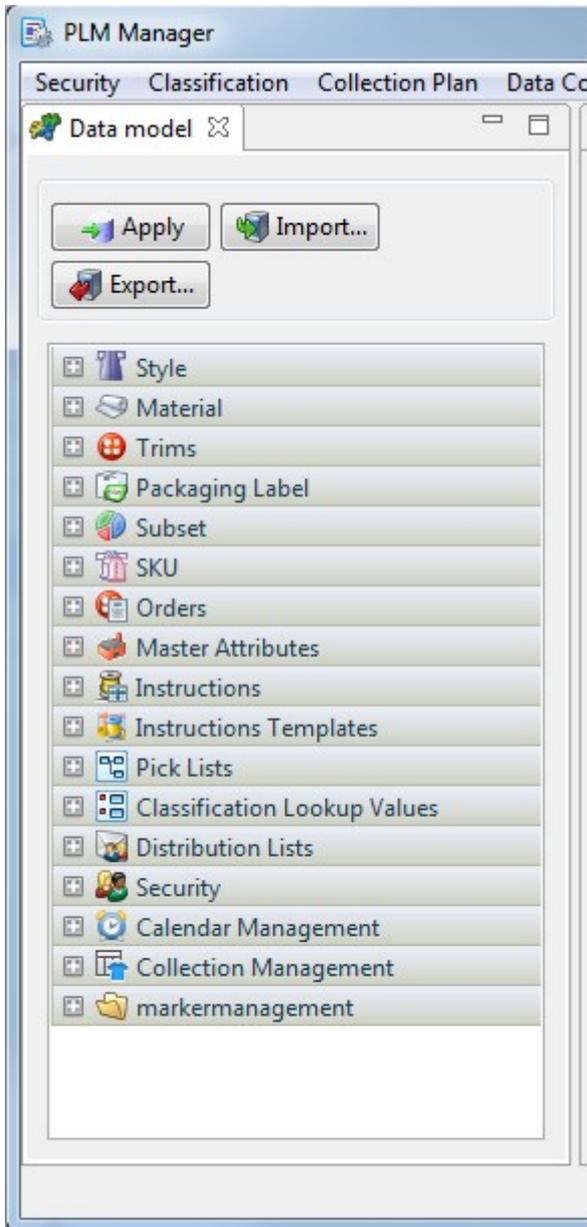


Par exemple : « MyCustomStringField » sur « Style » doit déclencher la création des données « MyCustomStringField » sur le type « String Field » dans la base de données.

Pour cela :

Ouvrez le module Administration et configuration

et le menu **Configuration des données/ Etendre modèle de données** pour afficher le panneau qui décrit toutes les données :



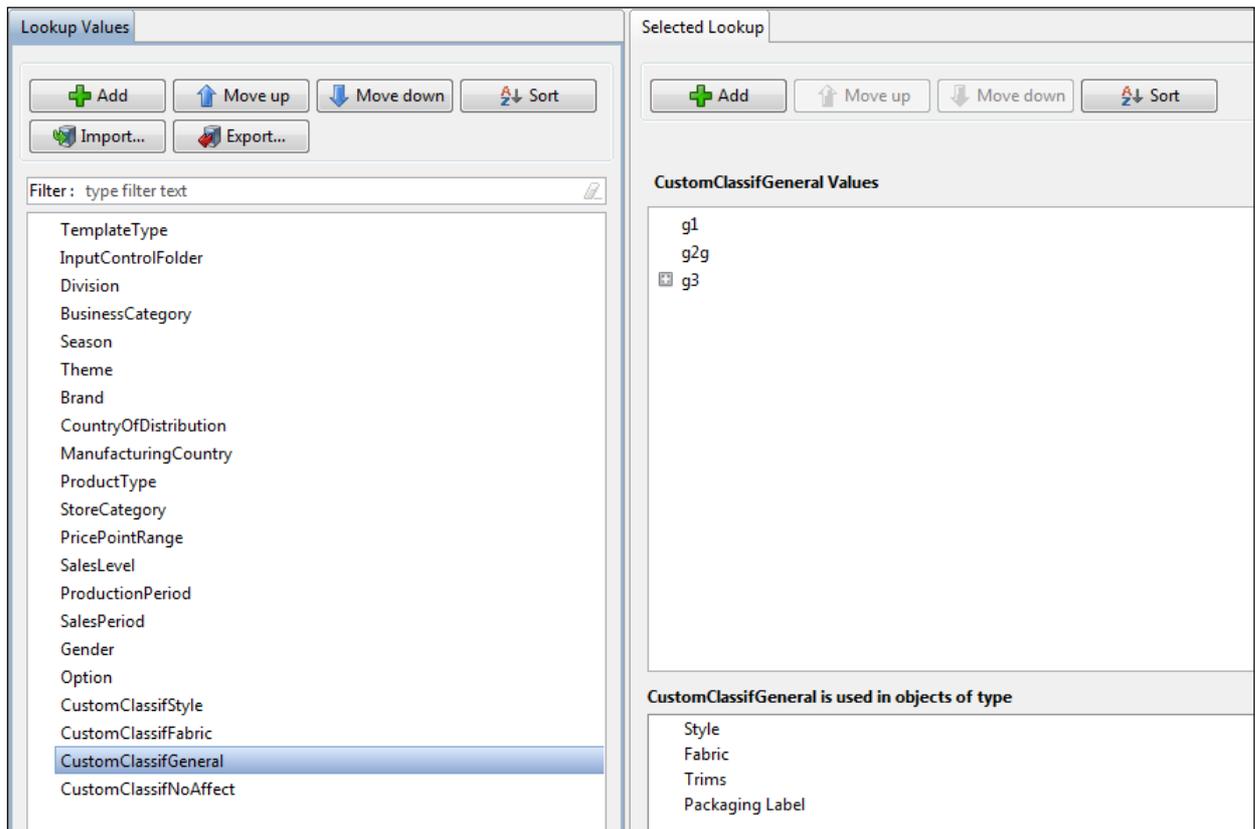
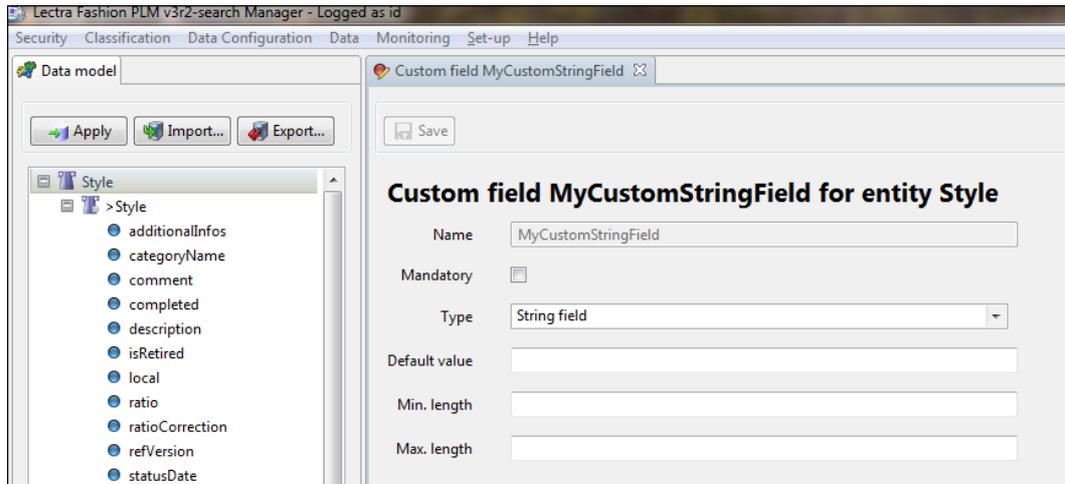
Faites un clic droit sur Style (Modèle).

Dans le menu contextuel : cliquez sur Add Custom Field (Créer une configuration).

Nommez la valeur MyCustomStringField.

Choisissez son type : String Field (Champ de chaîne).

Cliquez sur Enregistrer et appliquer la config à toutes les sous-catégories si vous avez besoin d'étendre la définition aux sous-catégories.



J. Ajout d'autres types de champs simples dans xml

Tous les différents types de champs définissables dans le module Administration et configuration possèdent une définition correspondante dans le fichier XML, conformément aux associations suivantes :

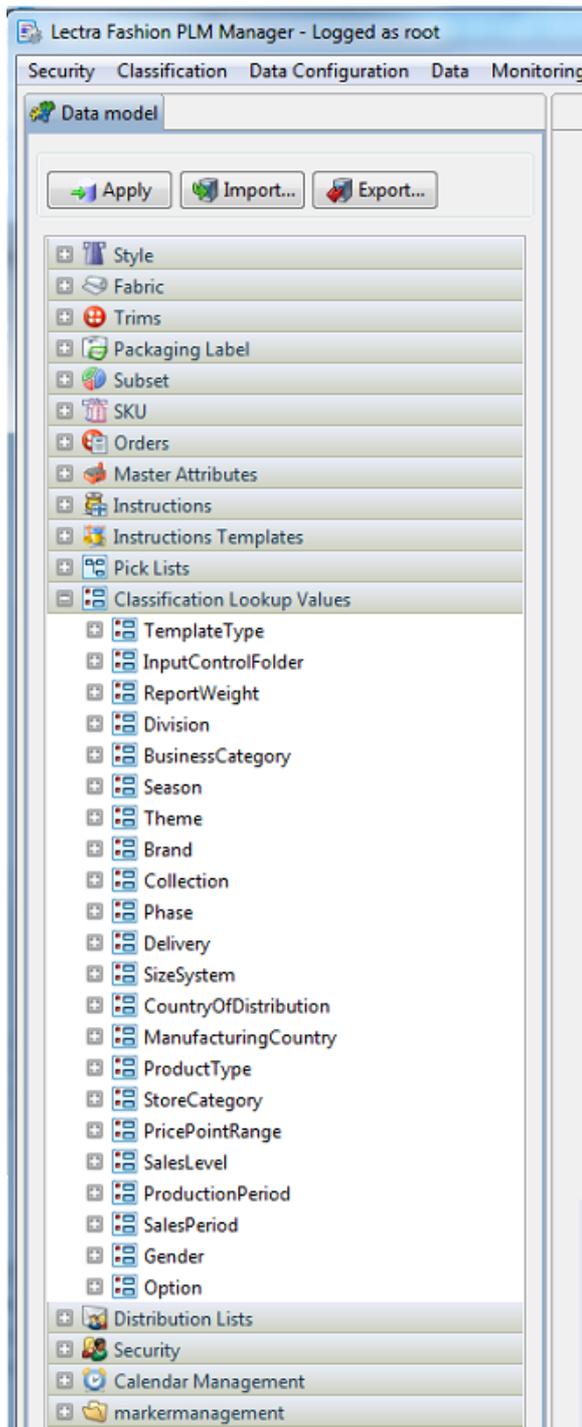
Boolean field	Twincombo / checkbox
Date field	datefield
Double field	doublefield
Float field	floatfield
Integer field	integerfield
Long field	longfield
Long string field	textarea
String field	textfield



L'insertion de différents types de champs dans le fichier xml de description doit être réalisée dans le fichier [namespace/custom/Lectra.PDM.User.xml](#).

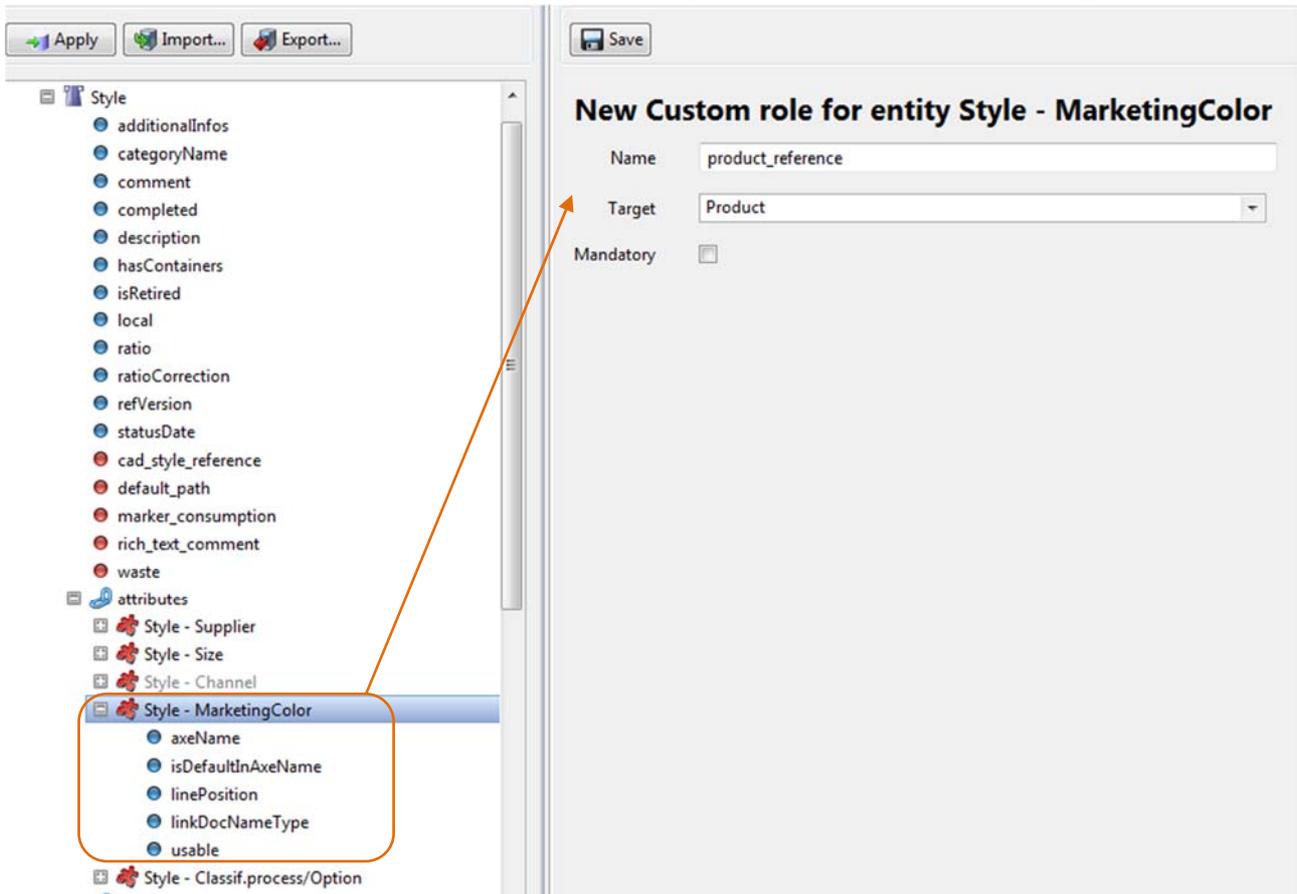
K. Ajout de « Pick Lists (Listes de sélection) » et de « Classification Lookup Values (Valeurs de classification) »

En observant le modèle de données du module Administration et configuration, vous pouvez voir ce qui a été créé :



L. Ajout du « Product Picker (Sélecteur de produits) » sur les attributs des produits

En observant le modèle de données du module Administration et configuration, vous pouvez créer un rôle personnalisé faisant référence au modèle de données de produit sur un de vos attributs de produits :



Changement de votre screensCusto.xml :

```
<screen name="Axis.Style" extends="Axis">
  .../...
  <tab name="colorsTab"...>
    <table name="colors"...>
      .../...
      <column name="product_reference"
i18n="product_reference" editable="true"/>
    </table>
  </tab>
  .../...
```

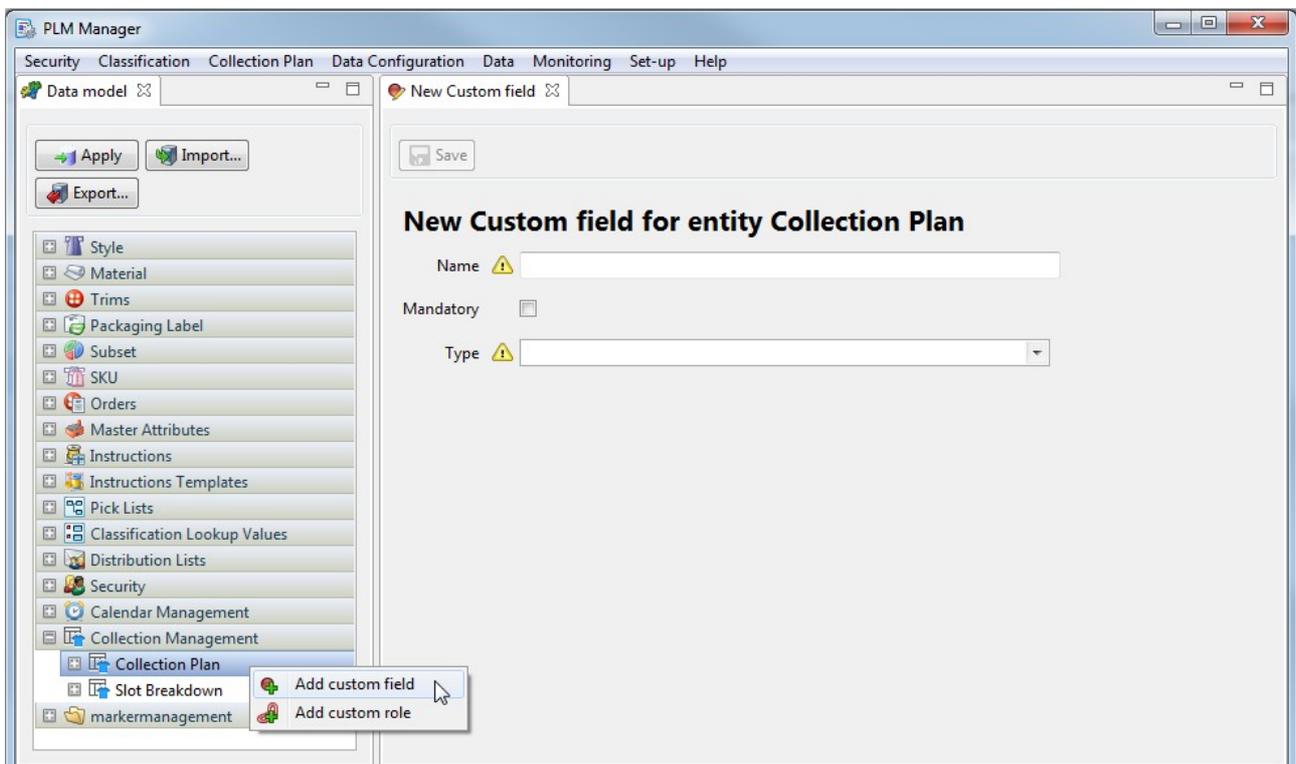
</screen>

Résultat dans l'écran Attributs du Style:



M. Ajout de champs et de liens personnalisés aux données générales d'un plan de collection

Faites un clic droit sur l'entrée "Collection Plan", puis sélectionnez « **Ajouter un champ personnalisé** » ou « **Ajouter un lien personnalisé** ».



N. Ajout de champs et de liens personnalisés à la grille des déclinaisons d'un plan de collection

Faites un clic droit sur l'entrée « Slot Breakdown (Déclinaison) », puis sélectionnez « Ajouter un champ personnalisé » ou « Ajouter un lien personnalisé ».