LECTRA FASHION PLM
PLM REPORT DESIGNER

User Guide

Date of last update: June, 2014

Contents

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

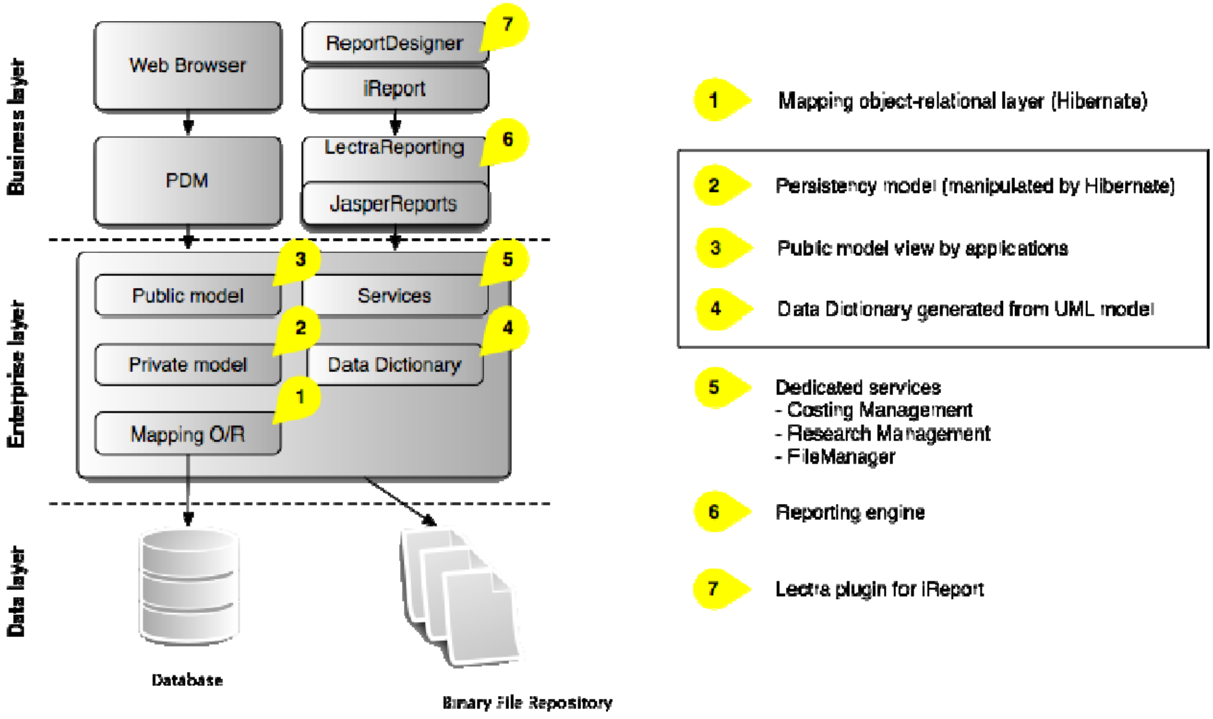
1. INTRODUCTION

This document aims at describing the architecture and main functionalities of the Report Designer plug-in.

In particular it describes the following functionalities:

- the connection to the PLM platform,
- the file manager integration in IReport,
- the query builder tools (construction, execution, profiling),
- the library elements available to construct reports,
- the report parameter management,
- the report internationalization.

2. ARCHITECTURE



1 Mapping object-relational layer (Hibernate)

2 Persistency model (manipulated by Hibernate)

3 Public model view by applications

4 Data Dictionary generated from UML model

5 Dedicated services
- Costing Management
- Research Management
- FileManager

6 Reporting engine

7 Lectra plugin for iReport

The reporting solution is an application plugged on the PLM platform like the PDM, and based on 2 modules:

- The reporting engine **LectraReporting** (based on Jasper Reports) process report templates deployed on the File Manager

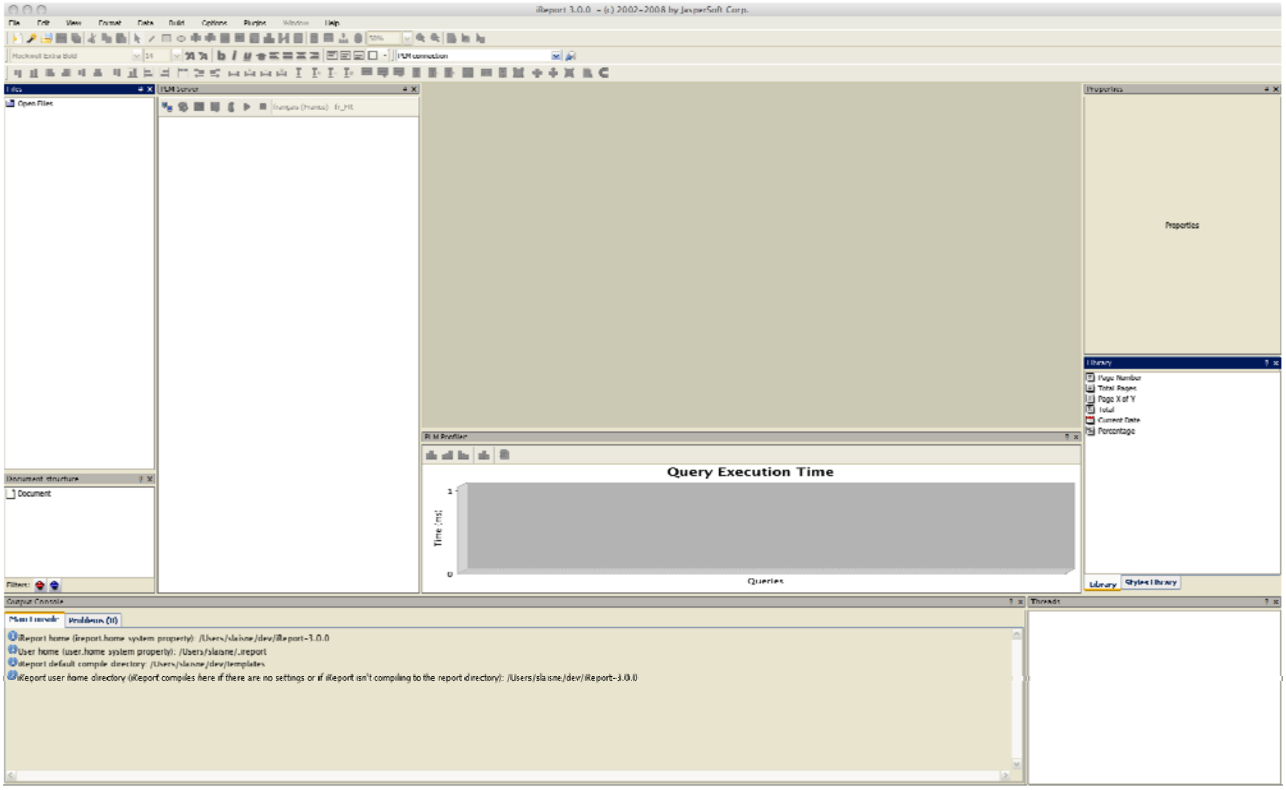
- The template editor named **ReportDesigner** (plug-in for iReport) offers the connection to the “Enterprise layer” in order to design, preview and deploy templates.

3. INSTALLATION

Refers to the PLM Manager Installer user guide for installation.

4. INTEGRATION OF REPORTDESIGNER IN IREPORT

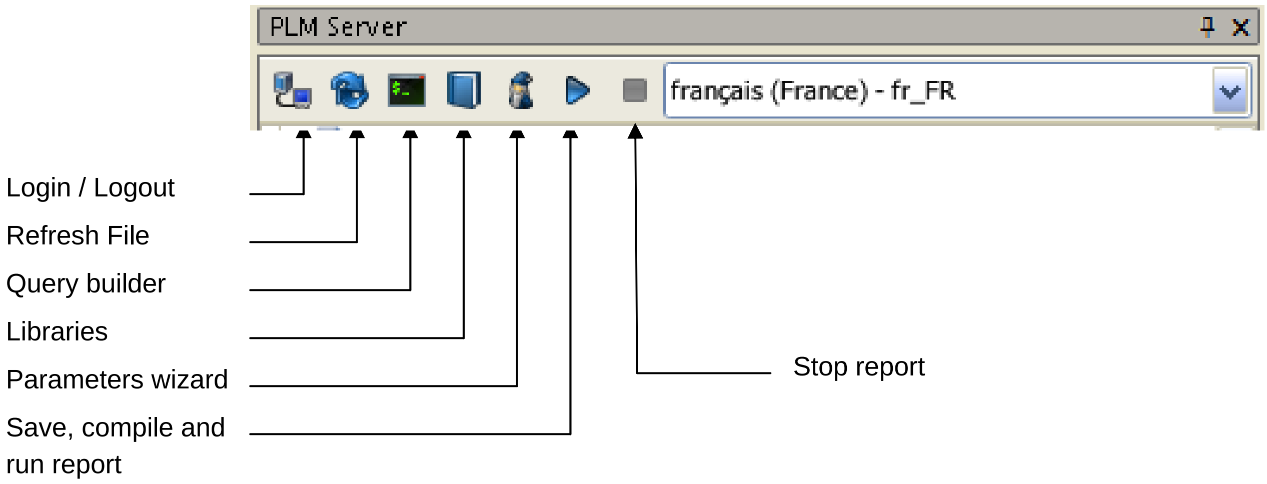
A main window of iReport is:



iReport home (ireport.home system property): /Users/slaisne/dev/iReport-3.0.0
User home (user.home system property): /Users/slaisne/.ireport
iReport default compile directory: /Users/slaisne/dev/templates
iReport user home directory (iReport compiles here if there are no settings or if iReport isn't compiling to the report directory): /Users/slaisne/dev/iReport-3.0.0

We are going to describe ReportDesigner elements.

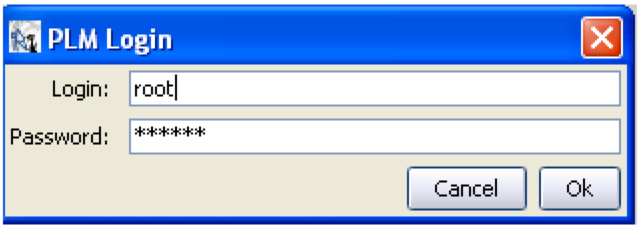
4.1 Toolbar



4.2 Login/logout

4.2.1 Login

1. In order to log into iReport, click on the login button: .
2. This connection wizard appears.



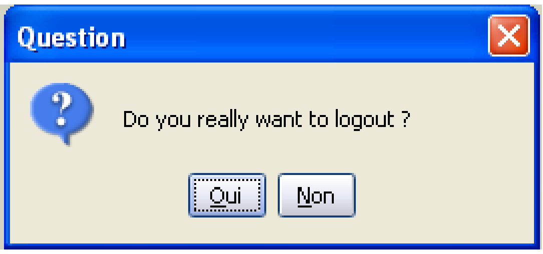
Login: root

Password: ******

3. Set your login and password, click on Ok.
4. If login and password are right, the “login successful” message appears, else wizard still wait for valid login and/or password. When you are logged, file manager tree is displayed and is replaced by

4.2.2 Logout

1. Click on the logout button .
2. Confirmation box appears.



Do you really want to logout ?

3. If you click on “No”, you stay connected, else you are logged out. File Manager Tree is hidden and is replaced by .

4.3 Refresh

is refresh button. It read elements included in the local template working directory and the PLM File Manager. For example, add a template in the template working directory. Click on refresh. The added template should be displayed in the File Manager Tree.

4.4 Query builder

displays query builder window. Query builder help to design HQL request with FQN naming.

4.5 PLM librairies

displays a window with tools that simplified access to technical information such as costing service, image renderer, technical color.

4.6 Parameter wizard

executes wizard that help to add java type or PLM type parameter.

4.7 Run Report with parameter

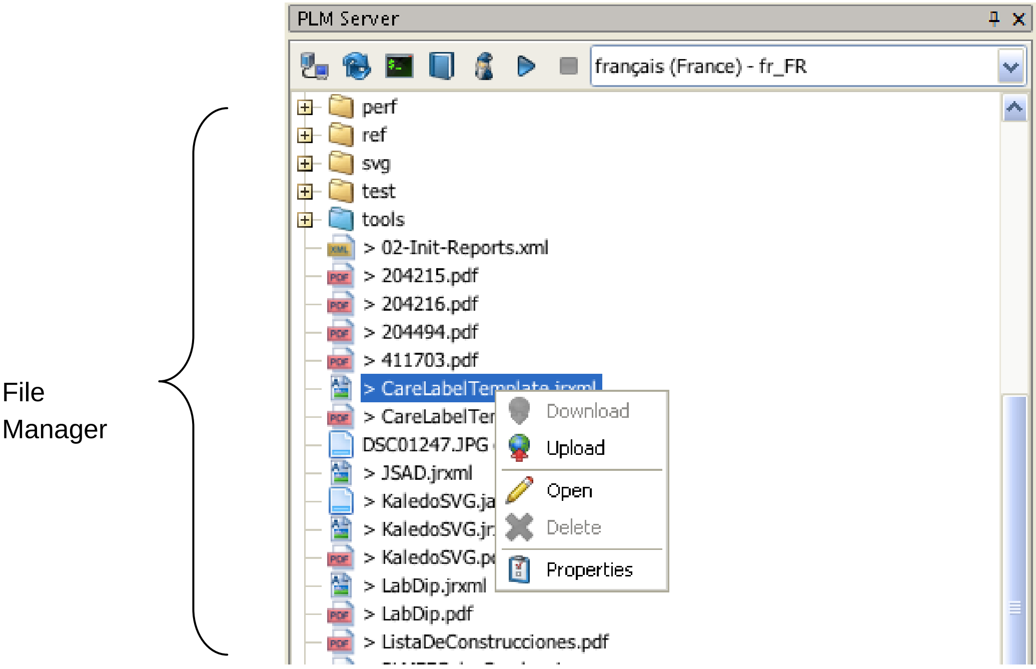
print, preview, parameter value selection for prompted parameter.

4.8 Stop Report

stop report execution

5. FILE MANAGER INTEGRATION

We suppose that you are now logged. File Manager Tree is displayed. The picture bellowed is an example. Mouse cursor point on “CollectionChildren.jrxml” and right mouse button click is done to have contextual menu.



File Manager is an interface between local files and PLM file system. With it, you can:

- Allows upload, download, update reports directly in iReport
- Distinguish file type with file icon
- Distinguish local or remote folder with color of folder icon:
  - 📁 for local folder
  - 📁 for remote folder
  - 📁 for remote and local folder
- Distinguish local file and downloaded file
  - “>” before file name for downloaded file
  - nothing before name for local file
- View file version

A file has different states:

- LOCAL_FILE: File is store in template working directory.
- UPLOADING: upload running
- DOWNLOADING: download running
- PROCESSED: final state.

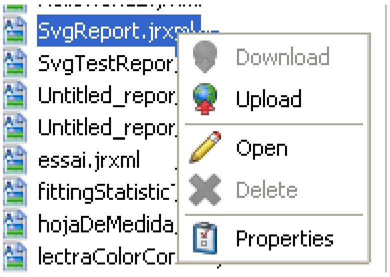
IN_PROCESS, TO_BE_PROCESSED, CHECK_TO_BE_PROCESSED, EXTRACTION_FAILED, IMPORT_CALL_FAILED, NO_OPERATOR, and NO_PROCESS_REQUIRED: some treatments are done on PLM File Manager side. Read its documentation to know their meaning.

☹ There is no concurrent version system. A same version file can be downloaded and modified by two persons. Only last uploaded file will have its modifications saved.

5.1 Upload file

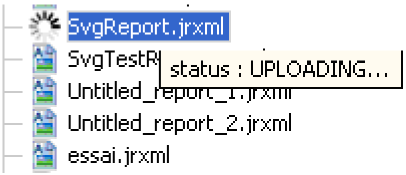
The upload functionality is available on local file (file included in the template working directory) and provides to upload a file on the File Manager.

1. Right click on the file name, and select upload.

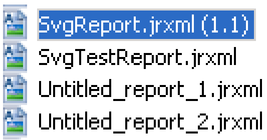


2. The “in progress” icon is displayed in place of icon type file. Tool tip could appear if you let mouse icon on name file. It indicates status of file.

Upload is running

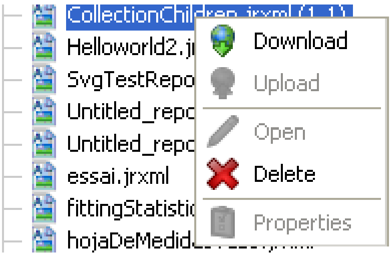


3. When the upload is done, the version number appears just after the filename :



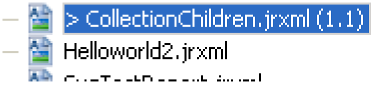
5.2 Download file

1. Right click on the file name, and select download.



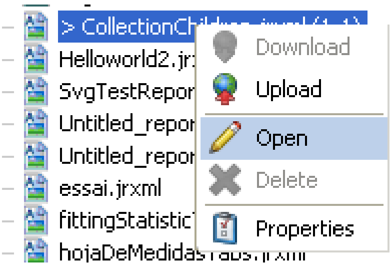
2. An icon display in place of icon type file. Tool tip could appear if you let mouse icon on name file. It indicates status of file.

3. When the download is done, the “>” character appears before the filename. It indicates that file is stored in the Template working directory.



5.3 Edit

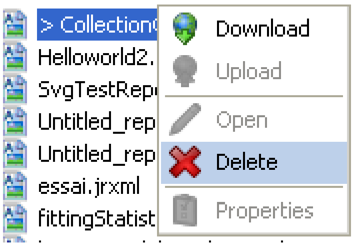
1. Right click on the filename, and select download.



2. Or you can double click on the report name file.

5.4 Delete

This function suppresses **only** remote files or directory in PLM File Manager. For directory, only if directory is empty.



5.5 Template classification

In order to access to templates from the PDM context, templates must be classified from an HVL named “TemplateType”.

- This HVL is defined by the enterprise

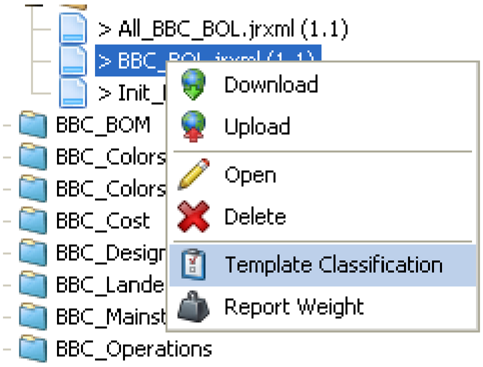
<propertydef name=”TemplateType” family=”process” arborescent=”true”/>.

Values must be added from XML init file or PDM.

- Classification applied on templates from iReport and XML init file

We suppose that file had been added to PLM File Manager and downloaded. Then, to define classification you should:

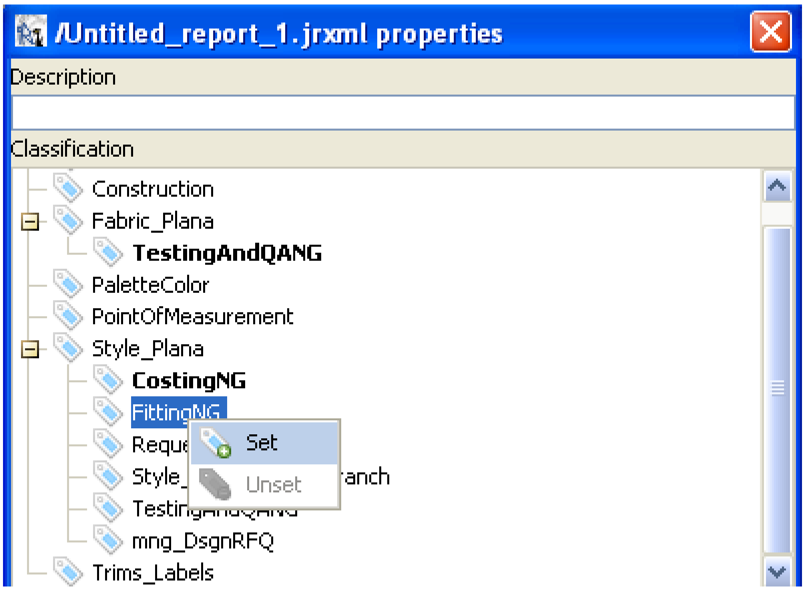
1. Click right on file to classify, and select Template Classification.



2. Classification hierarchy appeared :

   – To select a classification, click right on selected one and click on “Set”.

   – To unselect a classification, click right on element, and click on “Unset”.

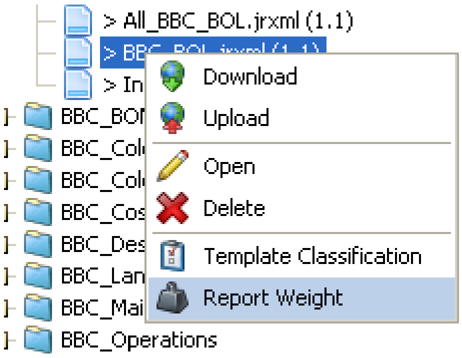


5.6 Report Weight Classification

In order to define the report weight, templates must be classified from a property named “ReportWeight”.

We suppose that the template file has been added to PLM File Manager and downloaded. Then, to define the report weight, you should:

3. Click right on the template file and select Report Weight.



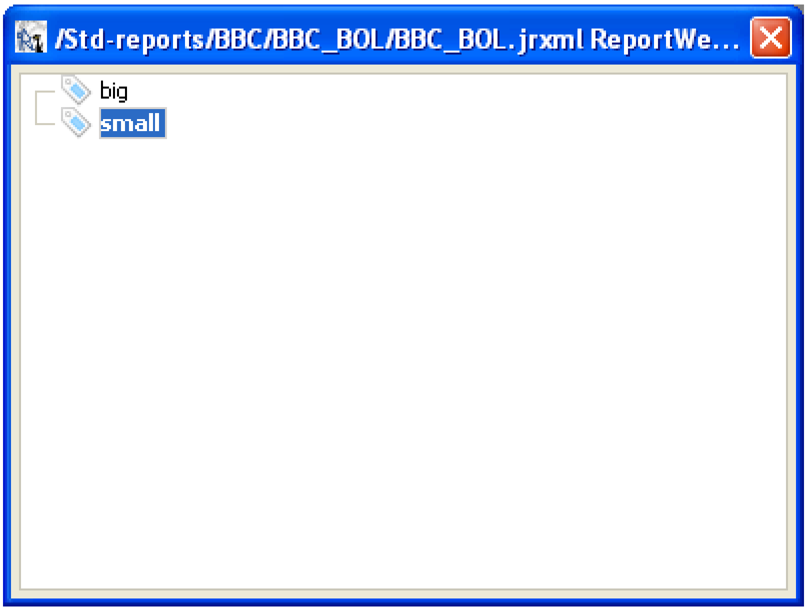
4. The available Report weights appear :

   a. Select the value “small” in order to use the generation queue dedicated for small reports (the report generation is less than 15 minutes).

   b. Select the value “big” in order to use the generation queue dedicated for the big reports (the report generation is more than 15 minutes).

      – To select a value, click right on selected one and click on “Set”.

      – To unselect a value, click right on element, and click on “Unset”.



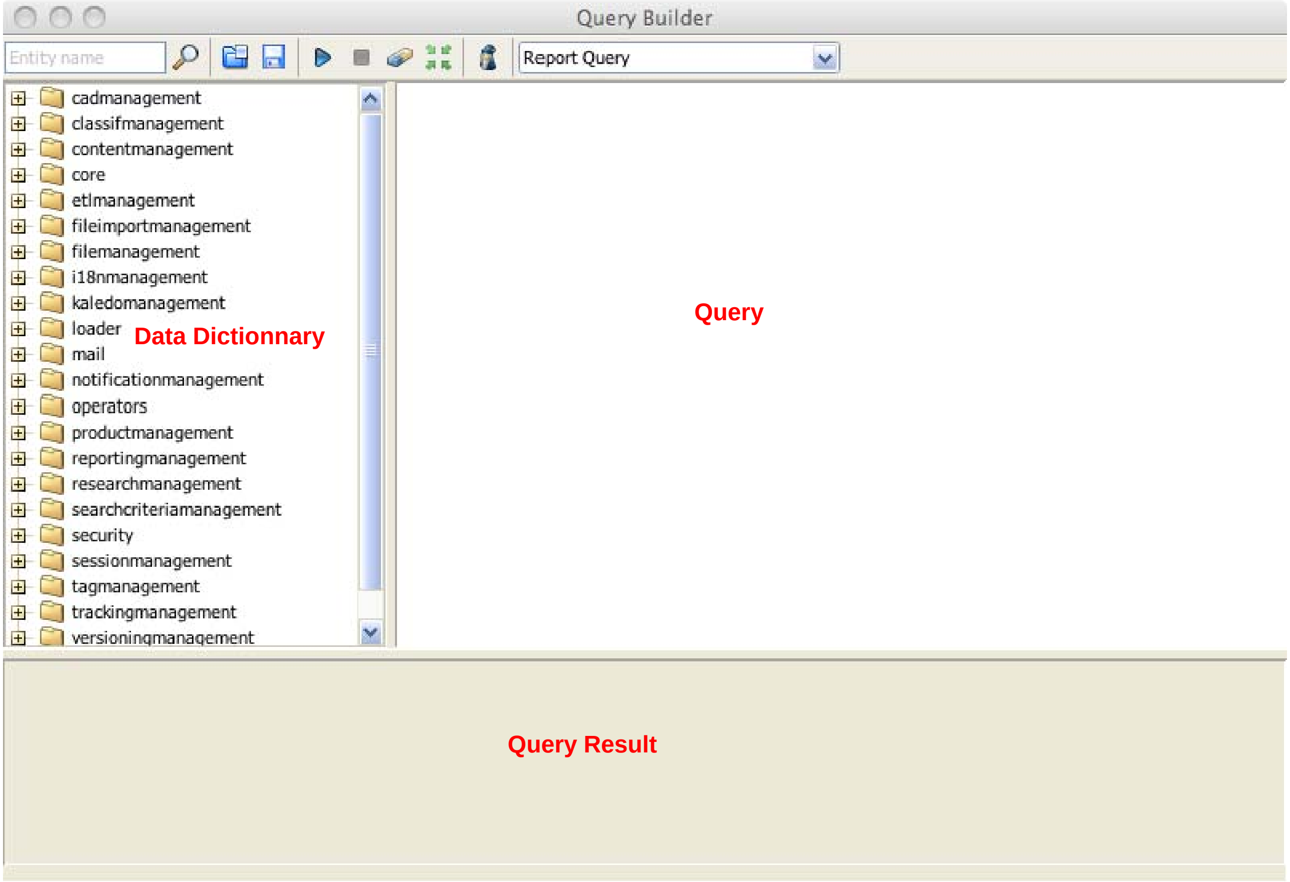
NB : you can also apply the report weight by a Report init file launched in the PLM Console :

```
<file path="/Std-reports/LabDip/LabDip_Sheet/LabDip_Sheet.jrxml" description="LabDip sheet"
content="Std-reports/LabDip/LabDip_Sheet/LabDip_Sheet.jrxml">
        <property name="ReportWeight"  family="config" path="big"/>
 </file>
```

6. QUERY BUILDER

Query builder is use to:

- Create query and fields for main dataset and sub dataset
- View Data Dictionary entities (UML model view)
- Prompt parameters if necessary
- Execute the query and view result
- Get values from custom data
- Create report fields directly



6.1 Query Language selection

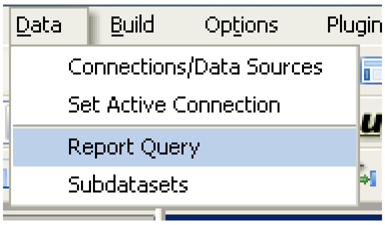
You can select the language for queries: **HQL** or **SQL**.

By default, the language is **HQL**.

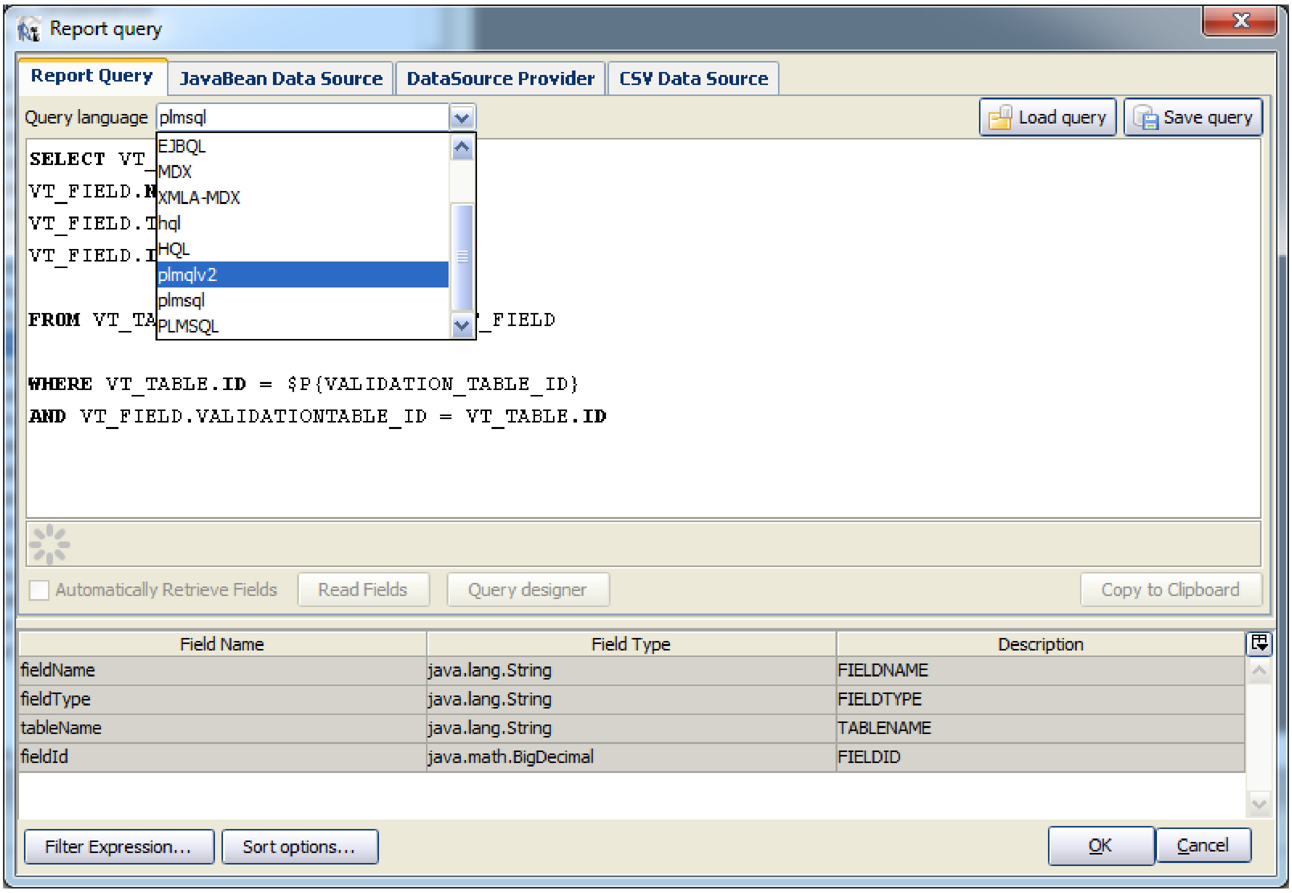
The query language is not available in the Query Builder window.

To select the language :

- go to the following menu :



- In the query language list,
  - Select **plmqlv2** for **HQL**.
  - Select **plmsql** for **SQL**.



Query language: plmsql

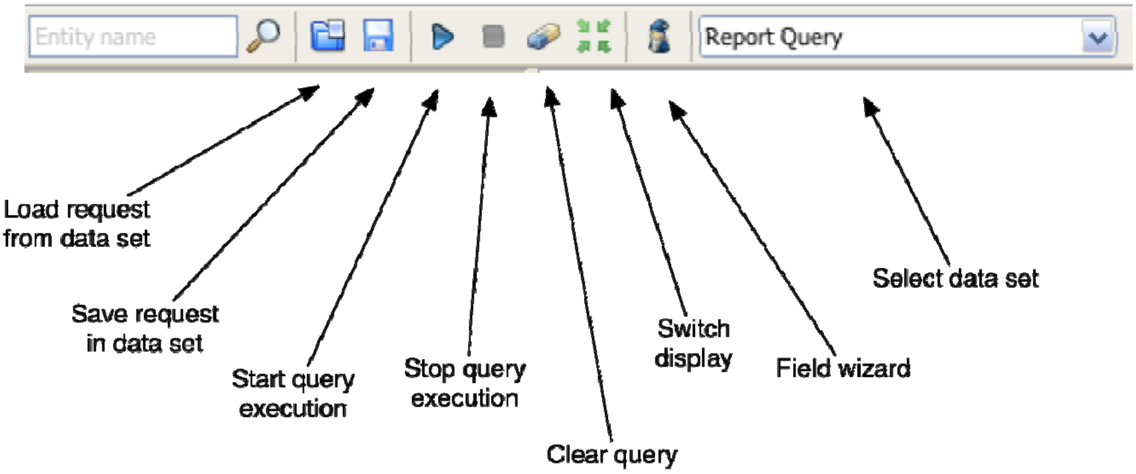
```
SELECT VT_
VT_FIELD.N
VT_FIELD.T
VT_FIELD.I

FROM VT_TA                               _FIELD

WHERE VT_TABLE.ID = $P{VALIDATION_TABLE_ID}
AND VT_FIELD.VALIDATIONTABLE_ID = VT_TABLE.ID
```

Field Name	Field Type	Description
fieldName	java.lang.String	FIELDNAME
fieldType	java.lang.String	FIELDTYPE
tableName	java.lang.String	TABLENAME
fieldId	java.math.BigDecimal	FIELDID

6.2 Query Builder tool bar



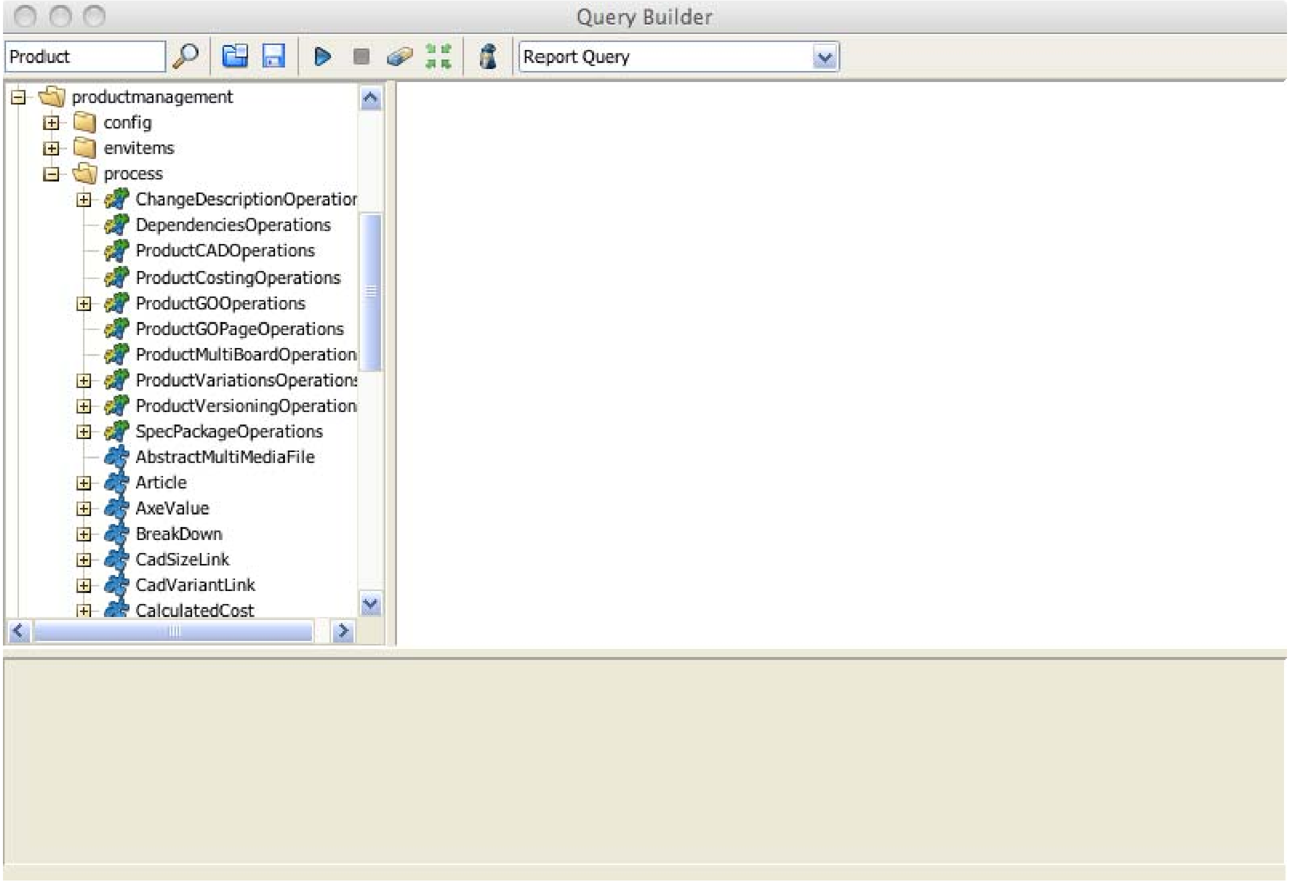
6.3 Toolbar elements

Icon	Description
📂	Load request from dataset: Reports contains a main dataset and eventually sub dataset (Only used for crosstab and chart tools). Each dataset have an associated request. To display request in query view, you can click on this button.
▶	Query execution: request is executed by par enterprise server. Result is displayed in query results view.
■	Stop request execution: execution of request may be to long. You can stop it by click on this button. It's active only when a request is run.
⌫	Clear request: Clear query view.
🧙	Field wizard: Create a report field from select part of request.
💾	Save request to dataset: associate request form view query to selected data set
⇄	Switch display: display instance id of object return or instance id of all object content in object return
Report Query ▾	Combo box to select current data set. Main data set name is Report Query; sub data set name is defined during its creation. If you create new sub data set in iReport, it is displayed in combo box.

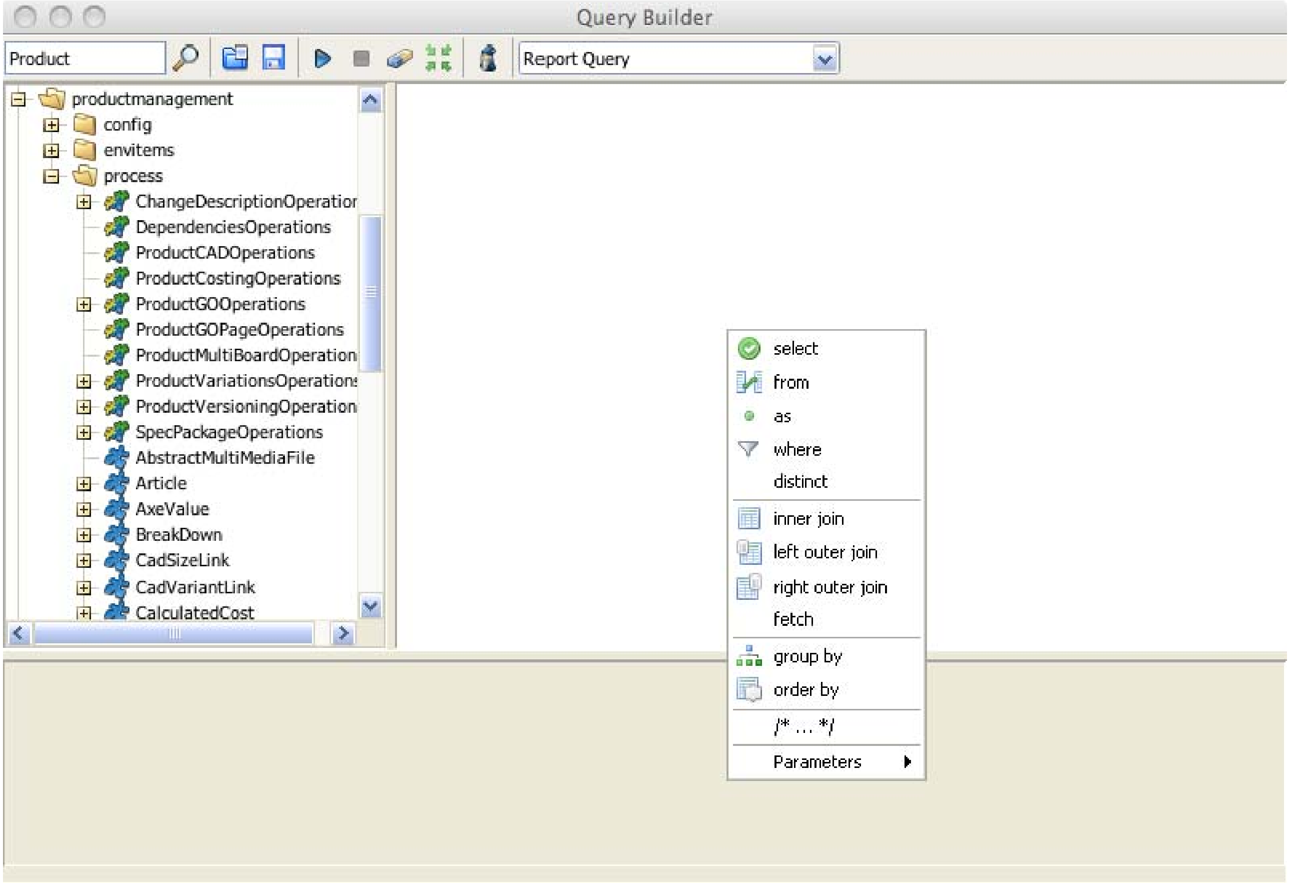
6.4 Create Query

We are going to give you fundamental element to write simple request.

1. Open query builder. Choose data set you want to create. Browse data dictionary to Product by open tree branch productmanagement, process.

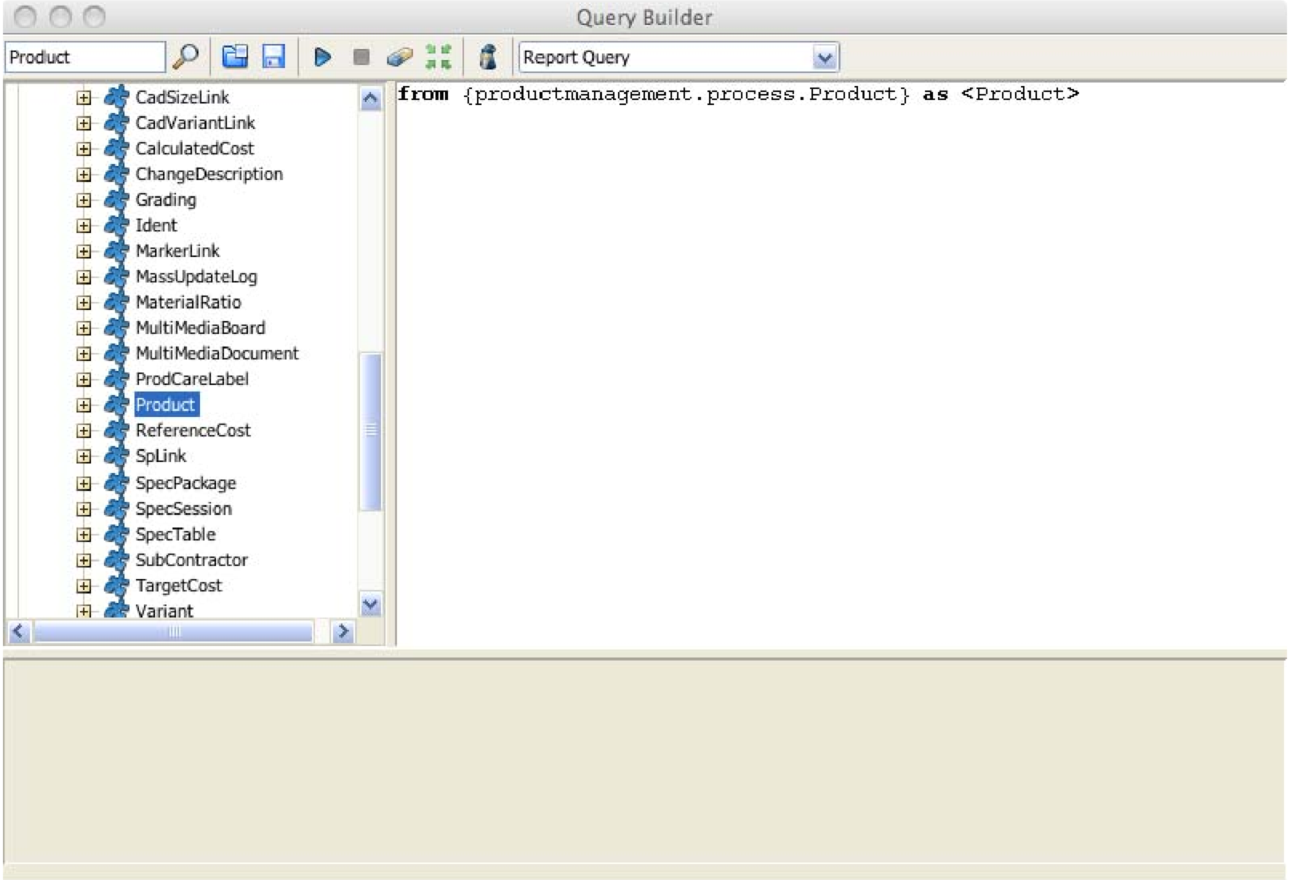


2. In query view, click right and select from keyword:



☺ You can see that all HQL request keyword can be acceded by right click in query views. Also, you can select parameter passed to report in “Parameter” menu.

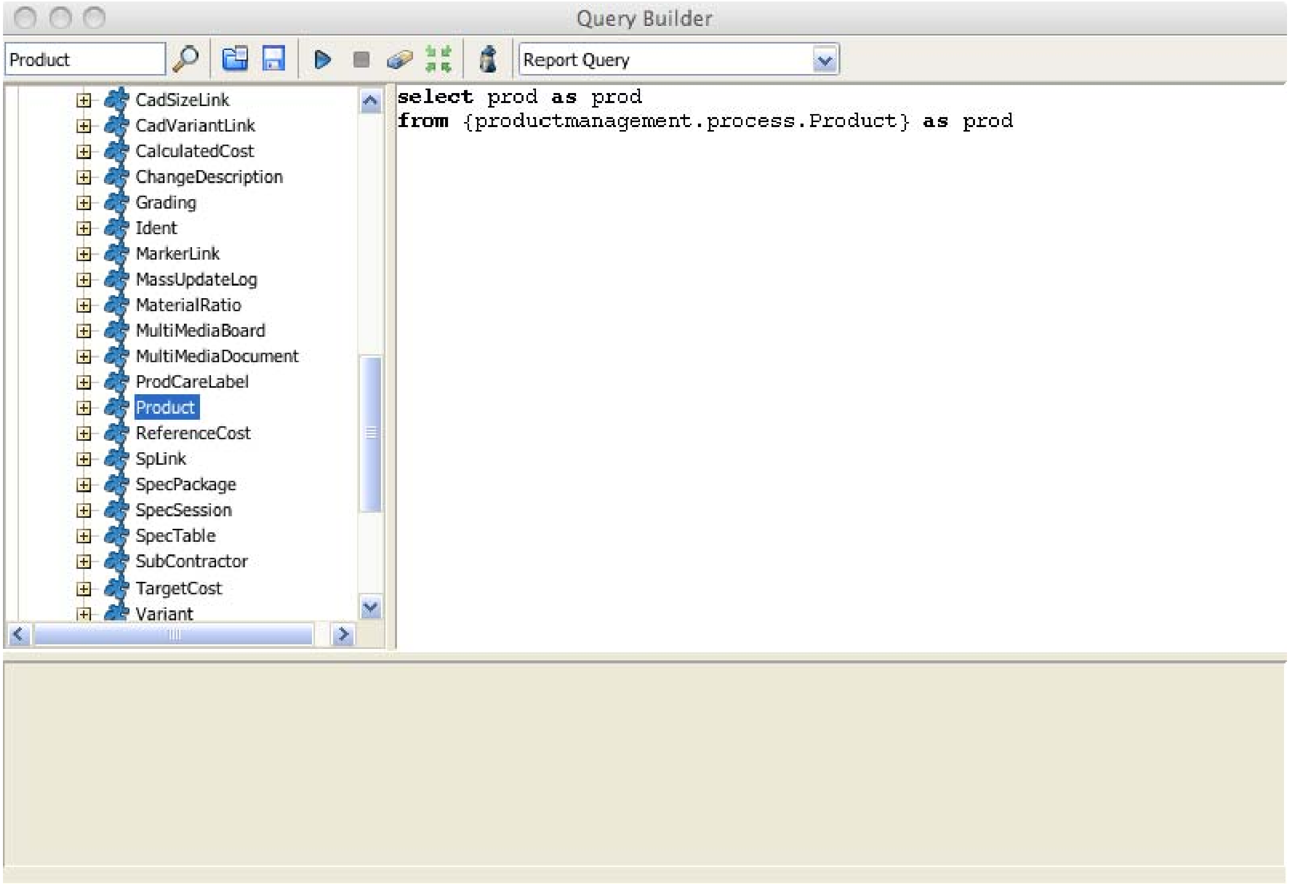
3. Drag and drop ⊞ Product to query view:



```
from {productmanagement.process.Product} as <Product>
```

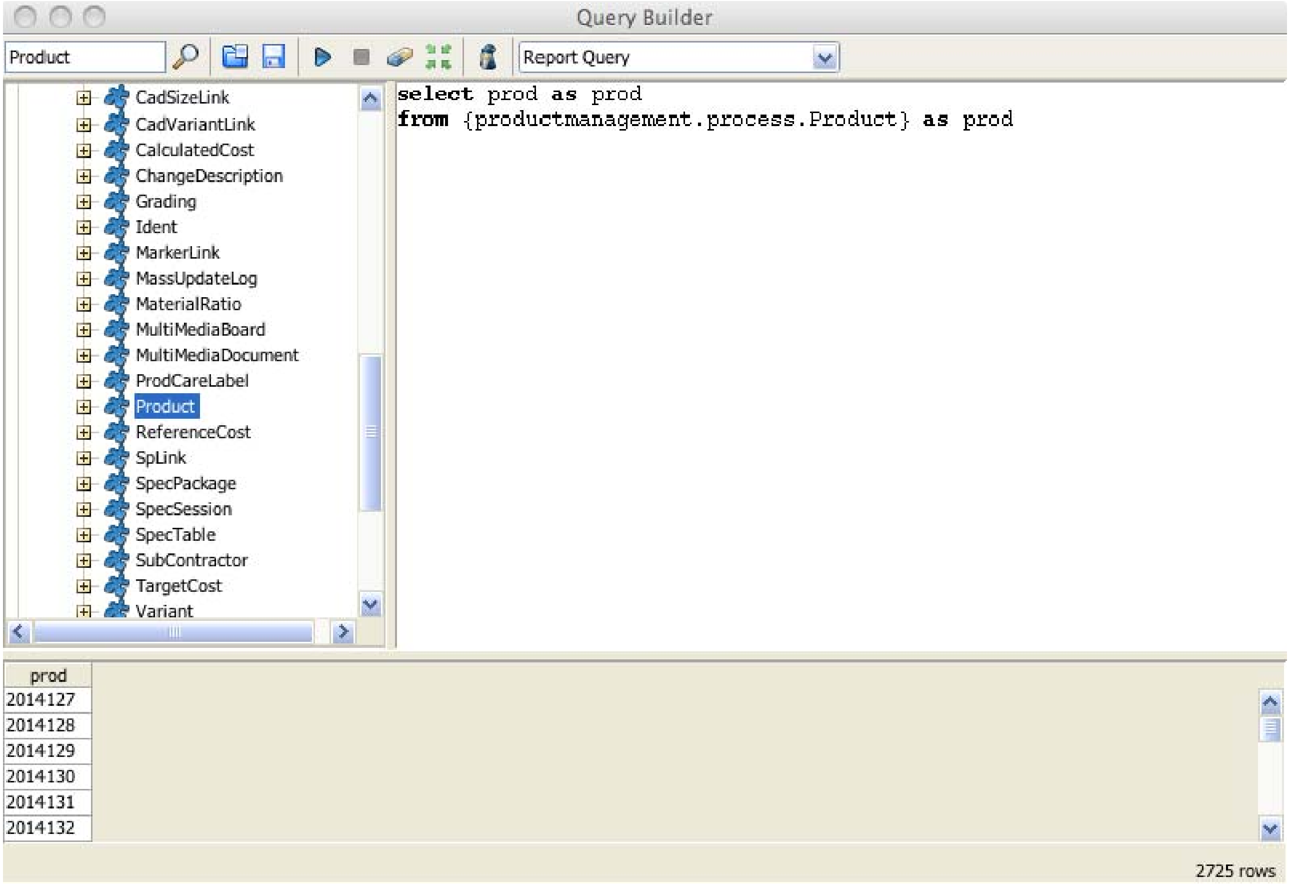
FQN value appears after ‘from”.

4. Complete query viewer to obtain this screen.



```
select prod as prod
from {productmanagement.process.Product} as prod
```

5. Click on ▶ and you obtain:



```
select prod as prod
from {productmanagement.process.Product} as prod
```

prod
2014127
2014128
2014129
2014130
2014131
2014132

2725 rows

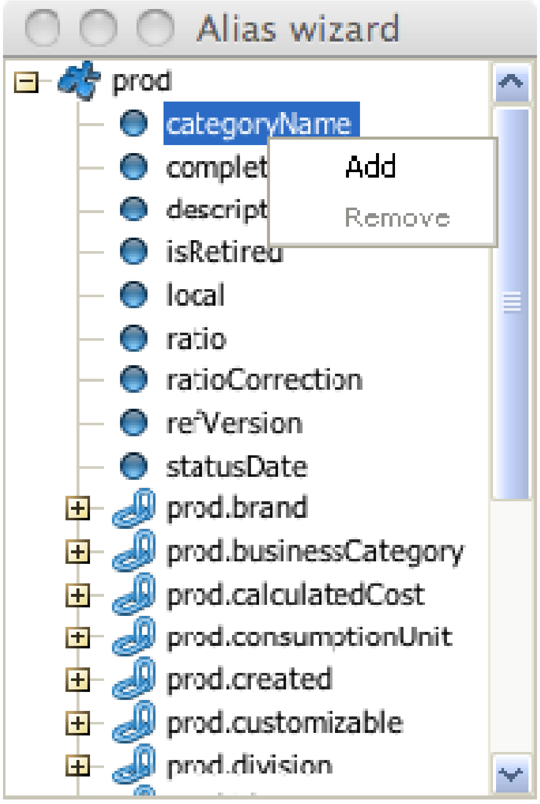
In Query Result view you have all element return by request execution. It is instance of object store in database. Now, we can select fields to display in report by click on .

😐 Always execute request before select fields.

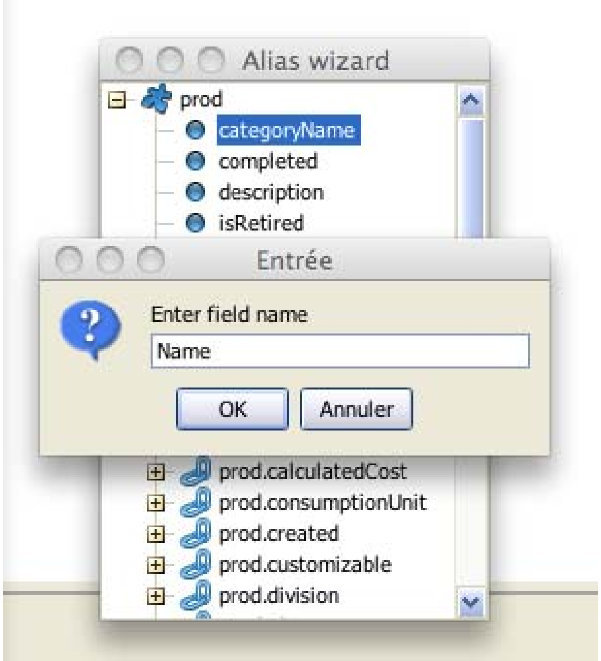
6. Select Fields: click on . Browse in product tree. We are going to display category name and description.

   a. Browse in product tree.

   b. Click right on field to display and select add. If you add wrong field you can select remove.



c. Give a name to field

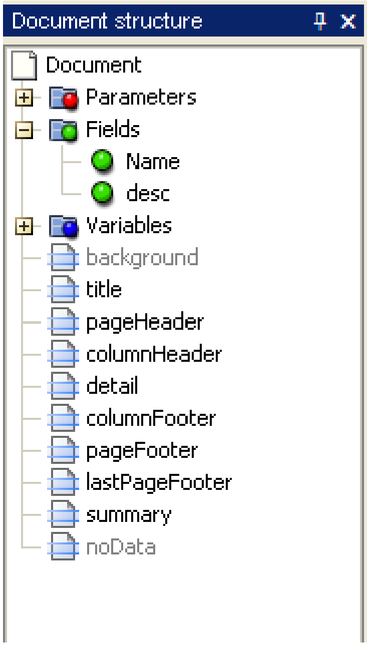


Enter field name

Name

d. Do same for description, use “desc” as field name

e. Automatically two fields are added in document structure



7. Now your request is ok. You have defined field. Click to export query in order to save request and link it with a dataset. By default it is main data set.

☺ If you choose another data set or quit without export query, a confirmation message box ask if you want to save you request, only if you have modified it.

7. QUERY LANGUAGE

In a global view, a report template is composed by a query to get data from the persistency layer, and a layout in order to display the results of the query execution. The “Mapping Object/Relational” module use “Hibernate” and the research service use its HQL (Hibernate Query Language) query language syntax.

This language is:

- Extremely powerful
- Full object-oriented
- Written on the Enterprise Data Dictionary
- returns Enterprise Model entities

7.1 The clause “from”

7.1.1 Description

Returns all instances of a class referenced by its FQN (Full Qualify Name)

7.1.2 Sample

from {productmanagement.process.Product} as p

7.2 The clause “select”

7.2.1 Description

Picks which objects and properties to return in the query result set. Alias is mandatory

7.2.2 Sample

select p as product

from {productmanagement.process.Product} as p

7.3 The clause “where”

7.3.1 Description

Allows to narrow the list of the instances returned

7.3.2 Sample

```
select p as product
from {productmanagement.process.Product} as p
where p.categoryName = ‘Style’
```

7.4 The clause “order by”

7.4.1 Description

The list returned by a query may be ordered by any property. The optional ascend or descend indicate ascending or descending order respectively.

7.4.2 Sample

```
select p as product
from {productmanagement.process.Product} as p
order by p.categoryName
```

7.5 Comment

7.5.1 Description

Comments are delimited by /* ... */

7.5.2 Sample

```
select p as product
/* get Product */
from {productmanagement.process.Product} as p
/* fetch techCode and studyCode */
inner join fetch p.ident
```

7.6 Joins

7.6.1 Inner Joins

7.6.1.1 Description

An inner join essentially combines the records from two tables (A and B) based on a given join-predicate

The inner join constructs may be abbreviated

7.6.1.2 Sample

select p as product, i as ident

from {productmanagement.process.Product} as p

inner join p.ident as i

Abbreviated version:

select p as product, **p.ident** as ident

from {productmanagement.process.Product} as p

7.6.2 Left outer join, right outer join and full join

7.6.2.1 Description

The result of a **left outer join** for list of object A and B always contains all records of the "left" list of object (A), even if the join-condition does not find any matching record in the "right" list of object (B)..

A **right outer join** closely resembles a left outer join, except with the lists reversed. Every record from the "right" list (B) will appear in the joined list at least once. If no matching row from the "left" list of object (A) exists, NULL will appear in columns from A for those records that have no match in A.

A **full join** combines the results of both left and right outer joins. The joined table will contain all records from both list, and fill in NULL for missing matches on either side. **This join is not usually useful**.

7.6.2.2 Sample

Left outer join:

select c

from {productmanagement.envitems.MarketingColor} as c

left outer join c.axeValues as axe

where axe.isDefaultInAxeName = true

and axe in elements (p.axes))

7.6.3 Fetch

7.6.3.1 Description

In addition, a “fetch” join allows associations or collections of values to be

initialized along with their parent objects, using a single select. Navigation beetween object is possible.

7.6.3.2 Sample

```
select p as product
from {productmanagement.process.Product} as p
inner join fetch p.ident
```

7.7 Sub query

7.7.1 Description

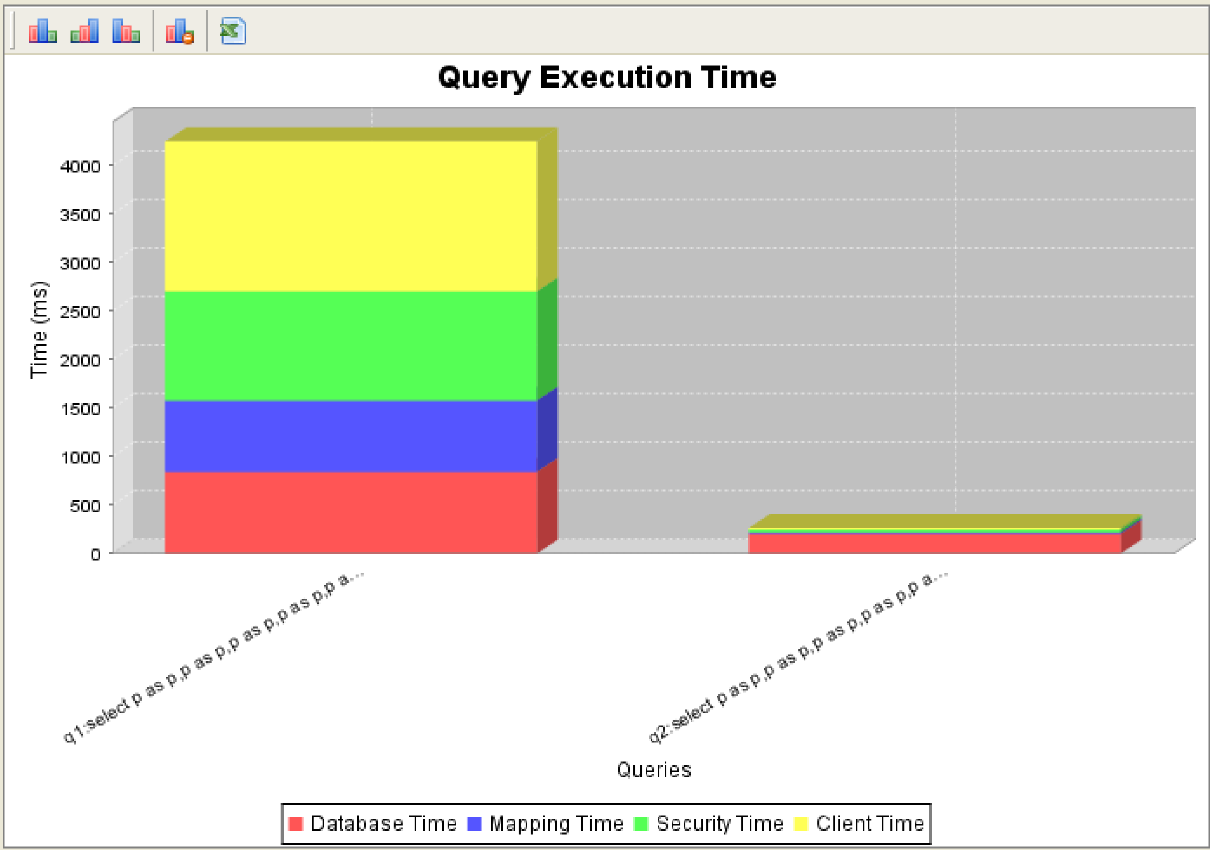
Allows selecting one element without impact the main query

7.7.2 Sample

```
select p as product,
(         select c
from {productmanagement.envitems.MarketingColor} as c
left outer join c.axeValues as axe
where axe.isDefaultInAxeName = true
and axe in elements (p.axes)
 ) as color
 from {productmanagement.process.Product} as p
```

8. QUERY PROFILER

This tool describe with a graph execution time of a query. Query is executed on a database (Database time), then result is transformed in object (Mapping time), user right are applied on object (security time), and finally time of data transfer between client and server.



Original order

Order by shortest to longest time

Order by longest to shortest time

Clear result

Export to excel file.

9. PLM LIBRARY

Access to technical information simply

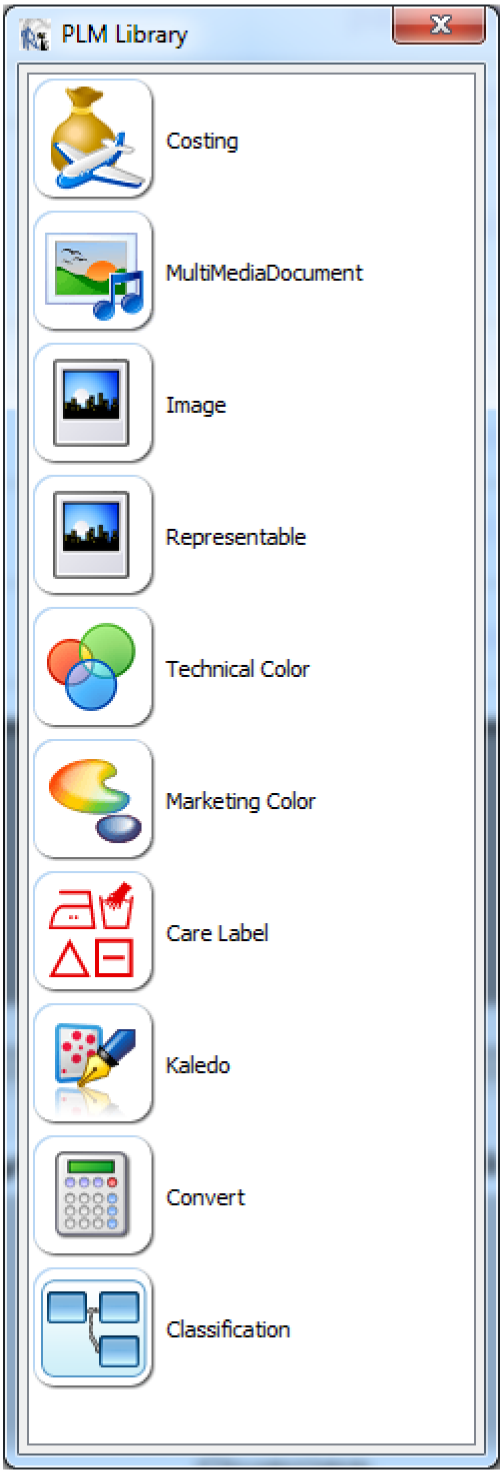
costing service

image renderer (path, file, content)

sub-report (path)

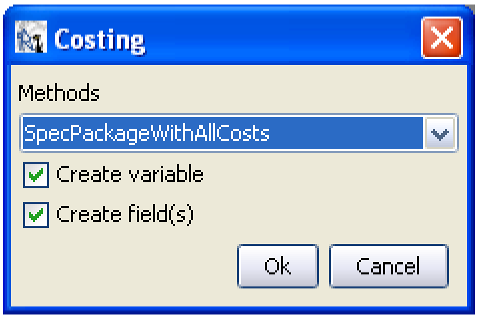
Technical color

All these functionalities are based on drag and Drop.



9.1 Costing Helper

1. Click on costing picture and drag and drop it into destination band in report.
2. Automatically, this box appeared :



Costing

Methods

SpecPackageWithAllCosts

☑ Create variable

☑ Create field(s)

Available product methods :

Method	Description	Description
SpecPackageWithAllCosts	This service enrichs the spec package with all calculated costs.	– SpecPackage specPackage (represent the specPackage owner and must contain the spec package id)
TotalSimulationCost	This service returns the sum of simulation costs for BOMs and BOL spec tables.	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)*
TotalSimulationCostWithLanded	This service returns the sum of simulation costs for BOMs and BOL spec tables with the landed.	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)*
LandedMargin	This service computes the landed margin	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)*
CalculatedIMU	This service computes the IMU %	– SpecPackage the spec package

		with all calculated costs (the return value of the *SpecPackageWithAllCosts* service)
CalculatedGrossProfit	This service computes the gross profit	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service)
TotalSimulationCostForSpecTable	This service returns the sum of simulation costs for a BOM or BOL spec table entered in parameter	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service) – SpecTable a BOL or BOM spec table
SimulationCostForBOMBOLLine	This service returns the simulation cost for a BOM or BOL line (spLink)	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service) – SpecTable a BOL or BOM spec table – SpLink the BOL or BOM line (spLink)
SimulationMarkerConsumption	This service returns the simulation consumption calculated from the selected marker	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service) – SpecTable a BOM spec table – SpLink the BOM line (spLink)
BOLLineRealCost	This service returns the real cost of a BOL line (spLink)	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service) – SpecTable a BOL spec table

		– SpLink the BOL line (spLink)
BOMDefaultPurchasePrice	The service returns the default price for a component in a BOM line (spLink)	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service) – SpecTable a BOM spec table – SpLink the BOM line (spLink)
LandedLineSimuApplyToValue	The service returns the simulation apply to value for a landed line	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service) – SpecTable a Landed spec table – SpLink the Landed line (spLink)
LandedLineSimuFormulaResult	The service returns the simulation formula result for a landed line	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service) – SpecTable a Landed spec table – SpLink the Landed line (spLink)
LandedLineSimuIntermediateTotalCost	The service returns the simulation intermediate total cost for a landed line	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service) – SpecTable a Landed spec table – SpLink the Landed line (spLink)
LandedLineSimuTotalCost	The service returns the simulation total cost for a landed line	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service) – SpecTable a Landed spec table – SpLink the Landed line (spLink)

BOMColorForPriceCalculationByColorSize	This service returns selected color for real cost calculation for a BOM line (spLink) according to the color and size (breakdown)	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)* – SpecTable a BOM spec table – SpLink the BOM line (spLink) – Color the selected color (MarketingColor) – Size the selected size (FinalSize)
BOMReferenceUnitPriceByColorSize	This service returns reference unit price for a BOM line (spLink) according to the color and size (breakdown)	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)* – SpecTable a BOM spec table – SpLink the BOM line (spLink) – Color the selected color (MarketingColor) – Size the selected size (FinalSize)
BOMLineRealCostByColorSize	This service returns the real cost of a BOM line (spLink) according to the color and size (breakdown)	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)* – SpecTable a BOM spec table – SpLink the BOM line (spLink) – Color the selected color (MarketingColor) – Size the selected size (FinalSize)
BOMBOLSpecTableRealCostByColorSize	This service returns the real cost of a BOM or BOL spec table according to the color and size (breakdown)	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)* – SpecTable a BOL or BOM spec table

		– Color the selected color (MarketingColor) – Size the selected size (FinalSize)
LandedLineApplyToCost	This service returns the landed apply to cost for a Landed line according to the color and size	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)* – SpecTable a Landed spec table – SpLink the Landed line (spLink) – Color the selected color (MarketingColor) – Size the selected size (FinalSize)
LandedLineValueCost	This service returns the landed value cost for a Landed line according to the color and size	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)* – SpecTable a Landed spec table – SpLink the Landed line (spLink) – Color the selected color (MarketingColor) – Size the selected size (FinalSize)
LandedLineIntermediateCost	This service returns the landed intermediate cost for a Landed line according to the color and size	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)* – SpecTable a Landed spec table – SpLink the Landed line (spLink) – Color the selected color (MarketingColor) – Size the selected size (FinalSize)
LandedLineTotalCost	This service returns the landed total cost for a Landed line according to the color and size	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts*

		service) – SpecTable a Landed spec table – SpLink the Landed line (spLink) – Color the selected color (MarketingColor) – Size the selected size (FinalSize)

TotalRealCostWithoutLandedByColorSize	This service returns the total real cost of a specPackage without the landed for a color and a size	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)* – SpecTable a Landed spec table – Color the selected color (MarketingColor) – Size the selected size (FinalSize)
TotalRealCostWithLandedByColorSize	This service returns the total real cost of a specPackage with the landed for a color and a size	– SpecPackage the spec package with all calculated costs (the return value of the *SpecPackageWithAllCosts* service*)* – SpecTable a Landed spec table – Color the selected color (MarketingColor) – Size the selected size (FinalSize)
TargetCostRetail	This service computes the target cost retail	– Double retailPricePoint (the retail price point) – Double targetIMU (the target IMU %)
TargetCostWholesale	This service computes the target cost wholesale	– Double sellingPrice (the selling price) – Double targetGrossProfit (the target gross profit)

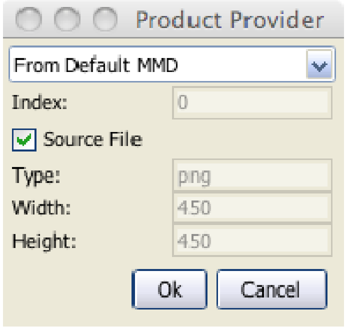
Method parameters :

- Create Fields: Automatically create field fill by request to be pass to the method.
- Create Variable: Automatically create variable that contain result of service Create Fields : Automatically create field fill by request to be pass to the method

3. Create request in Query Builder. The request must return generated field necessary for the costing services.

9.2 MultiMediaDocument Helper

1. Click on MultiMediaDocument picture and drag and drop it into destination band in report.
2. Automatically, this box appeared :



Product Provider

From Default MMD

Index: 0

☑ Source File

Type: png

Width: 450

Height: 450

Available product methods:

Method	Description
From Default MMD	Get the product default multimedia board image.
From MMD	Get a product multimedia board image according the line position
From Spec package default MMD	Get the spec package default multimedia board image.
From Spec package MMD	Get a spec package multimedia board image according to the line position.
From Spec table default MMD	Get the spec table default multimedia board image.
From Spec table MMD	Get a spec table multimedia board image according to the line position.
From Change Description	Get the change description image.

Method parameters:

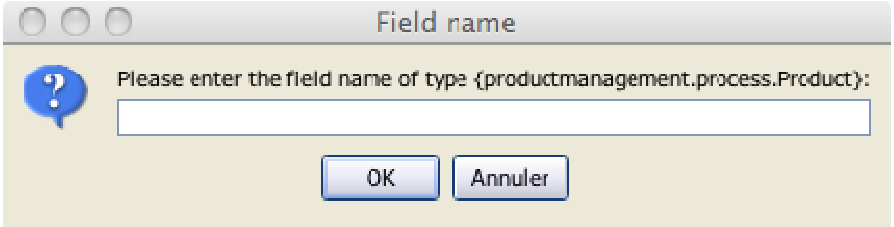
– Source File checkbox: the image renderer is made from the source file. Otherwise, the image renderer is made from the available thumbnail.

– Index *(available only for the methods: From MMD, From Spec package MMD)*. Set the line position of the multimedia board image.

- Type *(available only if the source file option is unchecked):* set the type of the thumbnail (by default “png”)
- Width *(available only if the source file option is unchecked)*: set the width of the thumbnail.
- Height *(available only if the source file option is unchecked)*: set the height of the thumbnail.

The return thumbnail is the one that have the closer size selected by the user. Set width = -1 and height = -1 to get the bigger one.

3. Click on the OK button. The following box automatically appears:



Field name

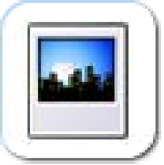
Please enter the field name of type {productmanagement.process.Product}:

4. Enter the field name returned by the template query that corresponds to :

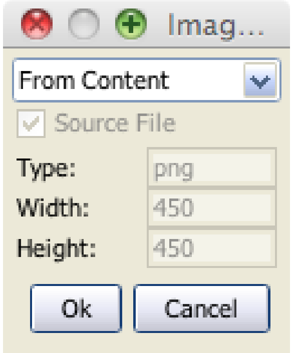
- The product object with the type {productmanagement.process.Product} for the method “*From Default MMD” and “From MMD”.*
- The spec package object with the type {productmanagement.process.SpecPackage} for the method “*From Spec package default MMD” and “From Spec package MMD”.*
- The change description object with the type {productmanagement.process.ChangeDescription} for the method “*From Change Description”.*

NB: The field is not created automatically and should exist.

9.3 Image Helper

1. Click on image picture  and drag and drop it into destination band in report.

2. Automatically, this box appeared :



Available product methods:

Method	Description
From Content	Get the image renderer from a file manager content.
From File	Get the image renderer from a file contained on the file manager.
From Resource	Get the image renderer from a resource defined in the classpath

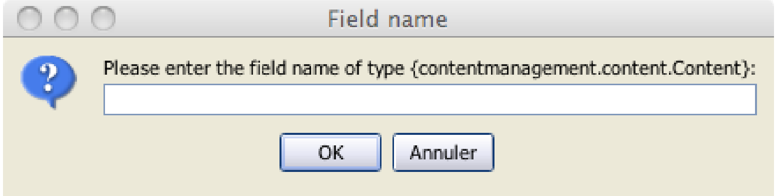
Method parameters (only for the From File method)

- Source File checkbox: the image renderer is made from the source file. Otherwise, the image renderer is made from the available thumbnail.
- Type *(available only if the source file option is unchecked):* set the type of the thumbnail (by default “png”)
- Width *(available only if the source file option is unchecked)*: set the width of the thumbnail.
- Height *(available only if the source file option is unchecked)*: set the height of the thumbnail.

The return thumbnail is the one that have the closer size selected by the user. Set width = -1 and height = -1 to get the bigger one.

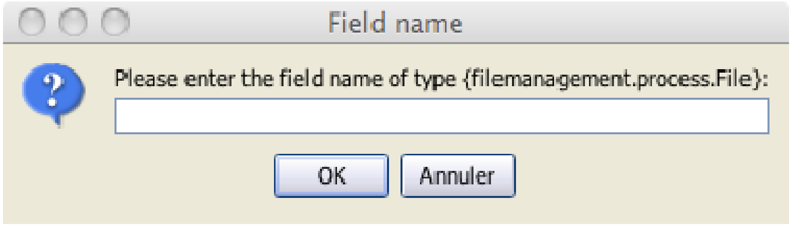
3. Click on the OK button.

   a. The following box automatically appears for “From content” methods:



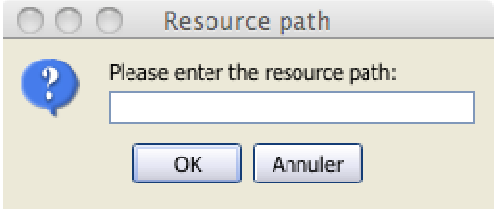
Please enter the field name of type {contentmanagement.content.Content}:

2. The following box automatically appears for “From file” methods:



Please enter the field name of type {filemanagement.process.File}:

b. The following box automatically appears for “From resource” :



Please enter the resource path:

Enter the path of the resource file.

NB: The field is not created automatically and should exist.

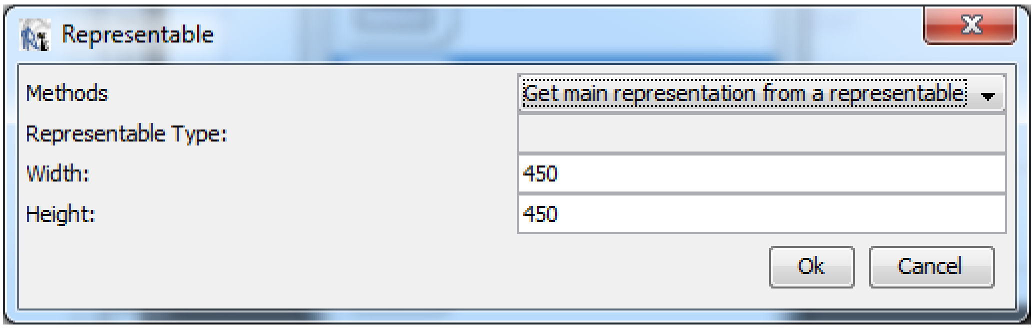
<u>Necessary area according to thumbnail size</u>

The following table permits to define the image area in report in order to have a good display.

Number of images displayed in a A4 page	Area size (in pixels) of the image in Report	Dimensions to setup for the Tiff image
16	112x124	Tiff450:450
16	267x225	Tiff800:800
4	267x350	Tiff1000:1000
1	535x560	Tiff1500:1500

9.4 Representable Helper

4. Click on image 🖼 and drag and drop it into destination band in report.
5. Automatically, this box appears :



Representable

Methods: Get main representation from a representable

Representable Type:

Width: 450

Height: 450

Available product methods:

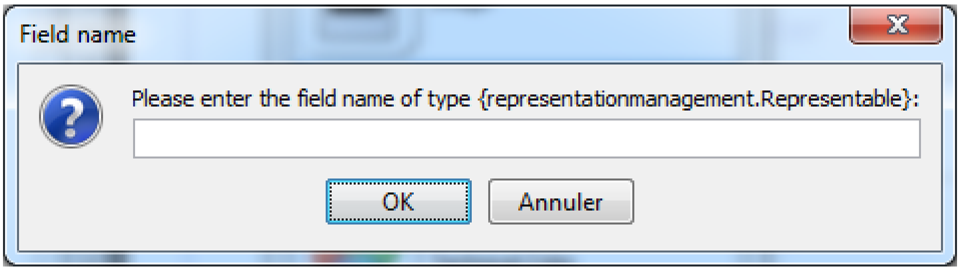
Method	Description
Get main representation from a representable	Get the image renderer for the main representation of a representable entity (example Prototype or Marker result entities).
Get a representation from a representable	Get the image renderer for the representation of a representable entity (example Prototype or Marker result entities) according to the type (example front ...).

Method parameters (only for the From File method)

- Type *(available only if the 'get main representation from a representable' method is selected):* set the type of the representation (for example "front")
- Width: set the width of the representation.
- Height*:* set the height of the representation.

The return representation is the one that has the closer size selected by the user.

6. Click on the OK button.
   a. The following box automatically appears for methods:



Field name

Please enter the field name of type {representationmanagement.Representable}:

Enter the field that has the representable type.

NB: The field is not created automatically and should exist.

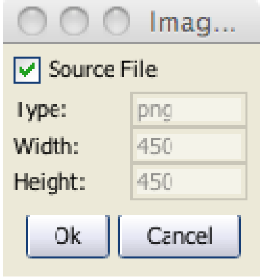
<u>Necessary area according to thumbnail size</u>

The following table permits to define the image area in report in order to have a good display.

Number of images displayed in a A4 page	Area size (in pixels) of the image in Report	Dimensions to setup the Tiff image
16	112x124	Tiff450:450
16	267x225	Tiff800:800
4	267x350	Tiff1000:1000
1	535x560	Tiff1500:1500

9.5 Technical color Helper

1. Click on technical color picture and drag and drop it into destination band in report.
2. Automatically, this box appeared :



☑ Source File

Type: png

Width: 450

Height: 450

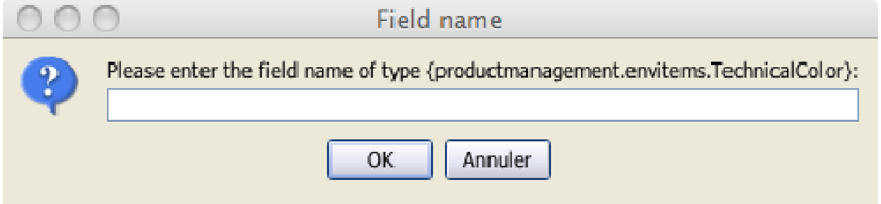
Method parameters

- <u>Source File checkbox:</u> the image renderer is made from the source file. Otherwise, the image renderer is made from the available thumbnail.
- <u>Type</u> *(available only if the source file option is unchecked):* set the type of the thumbnail (by default “png”)
- <u>Width</u> *(available only if the source file option is unchecked)*: set the width of the thumbnail.
- <u>Height</u> *(available only if the source file option is unchecked)*: set the height of the thumbnail.

The return thumbnail is the one that have the closer size selected by the user. Set width = -1 and height = -1 to get the bigger one.

3. Click on the OK button.

The following box automatically appears for “From File” and “From content” methods:



Please enter the field name of type {productmanagement.envitems.TechnicalColor}:

4. Enter the field name returned by the template query that corresponds to:

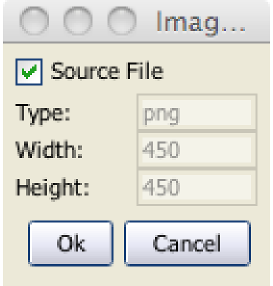
5.

- The technical color object with the type {productmanagement.envitems.TechnicalColor}.

NB: The field is not created automatically and should exist.

9.6 Marketing color Helper

1. Click on marketing color picture and drag and drop it into destination band in report.
2. Automatically, this box appeared :



☑ Source File

Type: png

Width: 450

Height: 450

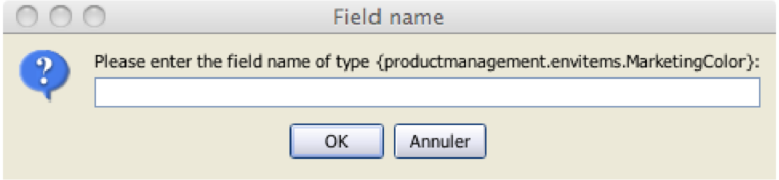
Method parameters

- Source File checkbox: the image renderer is made from the source file. Otherwise, the image renderer is made from the available thumbnail.
- Type *(available only if the source file option is unchecked):* set the type of the thumbnail (by default “png”)
- Width *(available only if the source file option is unchecked)*: set the width of the thumbnail.
- Height *(available only if the source file option is unchecked)*: set the height of the thumbnail.

The return thumbnail is the one that have the closer size selected by the user. Set width = -1 and height = -1 to get the bigger one.

3. Click on the OK button.

The following box automatically appears for “From File” and “From content” methods:



Please enter the field name of type {productmanagement.envitems.MarketingColor}:

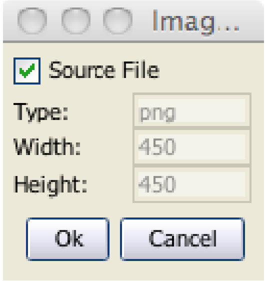
4. Enter the field name returned by the template query that corresponds to:

   – The marketing color object with the type {productmanagement.envitems.MarketingColor}.

NB: The field is not created automatically and should exist.

9.7 Care label Helper

1. Click on care label picture and drag and drop it into destination band in report.
2. Automatically, this box appeared :



☑ Source File

Type: png

Width: 450

Height: 450

Method parameters

– Source File checkbox: the image renderer is made from the source file. Otherwise, the image renderer is made from the available thumbnail.

– Type *(available only if the source file option is unchecked):* set the type of the thumbnail (by default “png”)

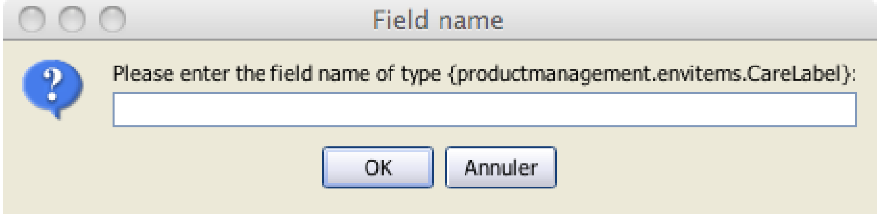
– Width *(available only if the source file option is unchecked)*: set the width of the thumbnail.

– Height *(available only if the source file option is unchecked)*: set the height of the thumbnail.

The return thumbnail is the one that have the closer size selected by the user. Set width = -1 and height = -1 to get the bigger one.

3. Click on the OK button.

The following box automatically appears for “From File” and “From content” methods:



Field name

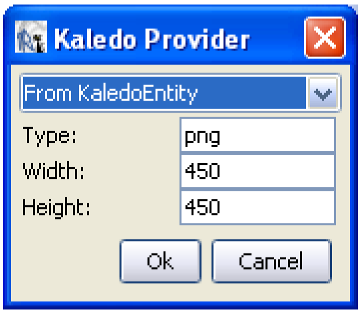
Please enter the field name of type {productmanagement.envitems.CareLabel}:

4. Enter the field name returned by the template query that corresponds to:
   - The care label object with the type {productmanagement.envitems.CareLabel}.

NB: The field is not created automatically and should exist.

9.8 Kaledo helper

1. Click on kaledo picture and drag and drop it into destination band in report.
2. Automatically, this box appeared :



Kaledo Provider

From KaledoEntity

Type: png
Width: 450
Height: 450

Available product methods:

Method	Description
From Kaledo Entity	Display a Kaledo entity
Get Palette Data	Display the palette data from a fabric entity

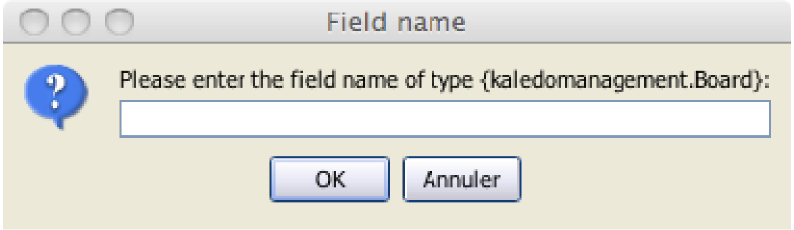
Method parameters:

For Kaledo object, the source file cannot be displayed directly. That's why; the Kaledo methods need thumbnail properties:

- Type : set the type of the thumbnail (by default "png")
- Width : set the width of the thumbnail.
- Height: set the height of the thumbnail.

The return thumbnail is the one that have the closer size selected by the user. Set width = -1 and height = -1 to get the default size.

3. Click on the OK button.



Please enter the field name of type {kaledomanagement.Board}:

4. Enter the field name returned by the template query that corresponds to:

For the Kaledo Entity method: One of kaledo entities

- The board object with the type {kaledomanagement.Board}
- The fabric object with the type {kaledomanagement.Fabric}
- The generic fabric object with the type {kaledomanagement.GenericFabric}
- The kaledo product object with the type {kaledomanagement.KaledoProduct}
- The style object with the type {kaledomanagement.Style}
- The coloris object with the type {kaledomanagement.Coloris}

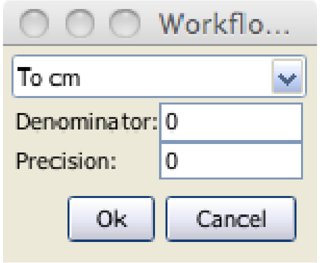
For the Get Palette Data method: One of fabric entities

- Basic, Print, Weave, Knit fabrics.

NB: The field is not created automatically and should exist.

9.9 Convert Helper

1. Click on convert picture and drag and drop it into destination band in report.
2. Automatically, this box appeared :



Workflo...

To cm

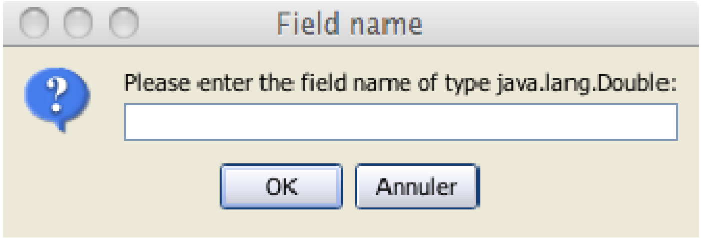
Denominator: 0

Precision: 0

Available product methods:

Method	Description
To cm	Convert to cm
To mm	Convert to mm
To inch	Convert to inch

3. Click on the OK button



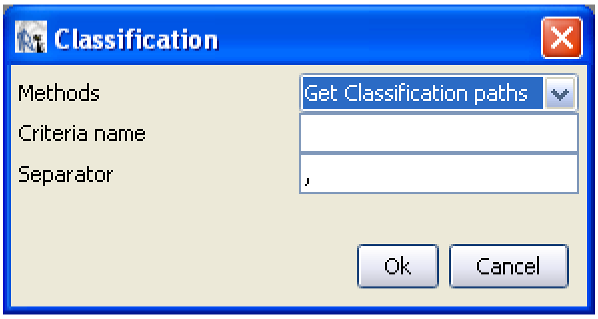
Field name

Please enter the field name of type java.lang.Double:

4. Enter the field name of the value

9.10 Classification Helper

3. Click on convert picture and drag and drop it into destination band in report.
4. Automatically, this box appeared :



Classification

Methods: Get Classification paths

Criteria name:

Separator: ,

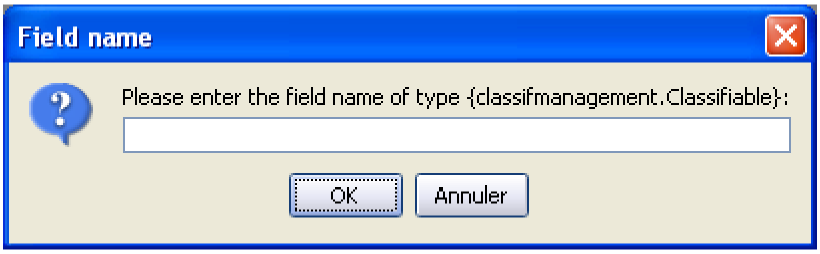
Available classification methods:

Method	Description
Get classification paths	Get paths of classification criterias applied on a entity
Get classification names	Get names of classification criterias applied on a entity

Method parameters:

- Criteria name : name of the criteria (ex Season, Brand …) **Mandatory**
- Separator: the string that separates the values. By default, the separator is ‘,’ **Mandatory**

5. Click on the OK button



Field name

Please enter the field name of type {classifmanagement.Classifiable}:

6. Enter the field name of the classifiable entity that already exists.

10. REPORT PARAMETER WIZARD

There are two types of parameters:

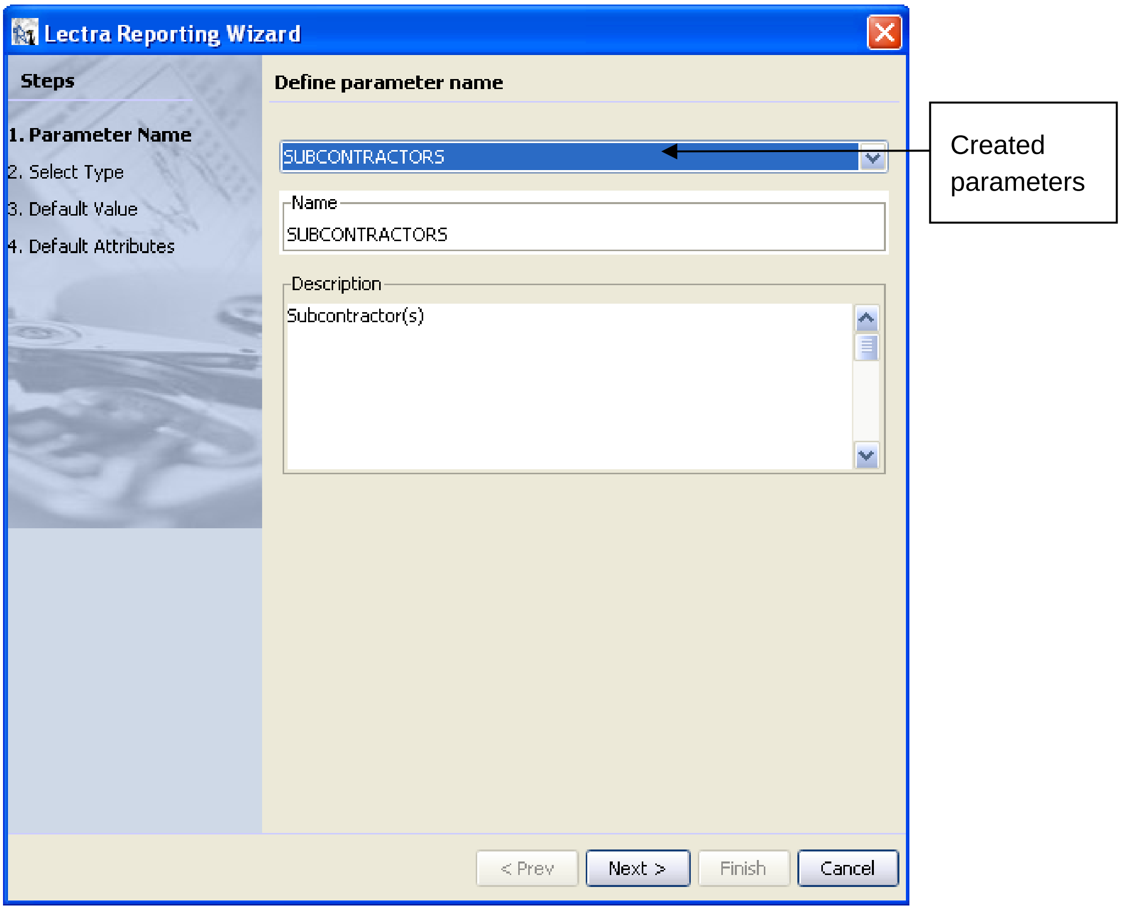
- The main parameter: It is the entry point to the report and is mandatory and cannot be prompted.
- The dependent parameters: These are prompted parameters seen in PDM report print popup.

To define the list of available values for a parameter, a set of keys and values should be filled.

We take the example of a report with 3 parameters :

- the main one is SPEC_PACKAGE, The report will be launched on a spec package.
- 2 others parameters depends of the main parameters :
  - The AXE_COLOR parameter: the available values are the color in variation axes.
  - The AXE_SIZE parameter: the available values are the size in variation axes.

Define parameter Name


Lectra Reporting Wizard

Steps

1. Parameter Name
2. Select Type
3. Default Value
4. Default Attributes

Define parameter name

SUBCONTRACTORS

Name: SUBCONTRACTORS

Description: Subcontractor(s)

Created parameters

Parameter list :

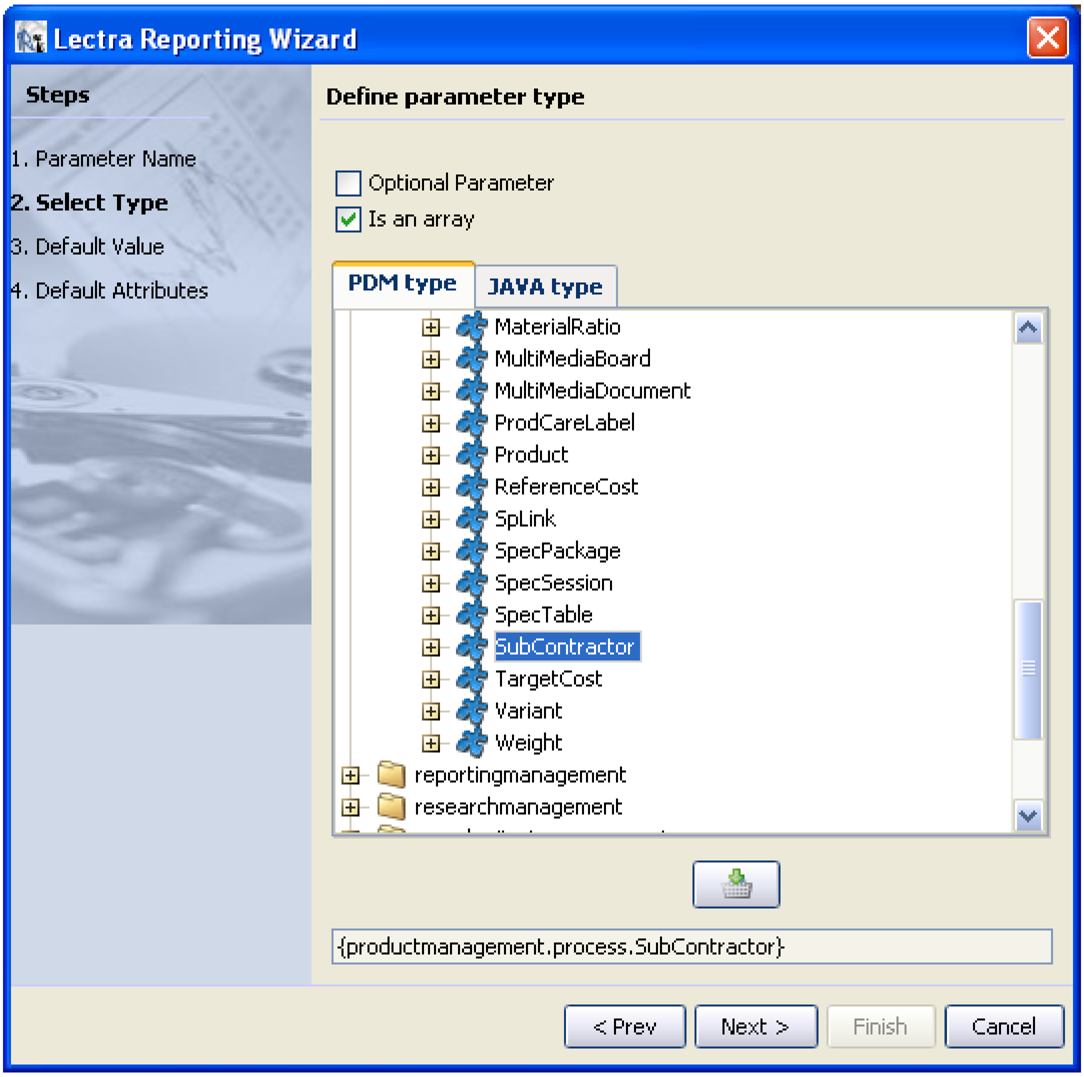
The combo box contains all existing parameters.

If you want to modify an existing parameter, select it in combo box. Else set new Name and eventually its description.

Fields :

- Name : name of the parameter
- Description : description of the parameter

10.1 Define parameter type



Lectra Reporting Wizard

Steps

1. Parameter Name
2. **Select Type**
3. Default Value
4. Default Attributes

Define parameter type

☐ Optional Parameter
☑ Is an array

PDM type | JAVA type

- MaterialRatio
- MultiMediaBoard
- MultiMediaDocument
- ProdCareLabel
- Product
- ReferenceCost
- SpLink
- SpecPackage
- SpecSession
- SpecTable
- SubContractor
- TargetCost
- Variant
- Weight
- reportingmanagement
- researchmanagement

{productmanagement.process.SubContractor}

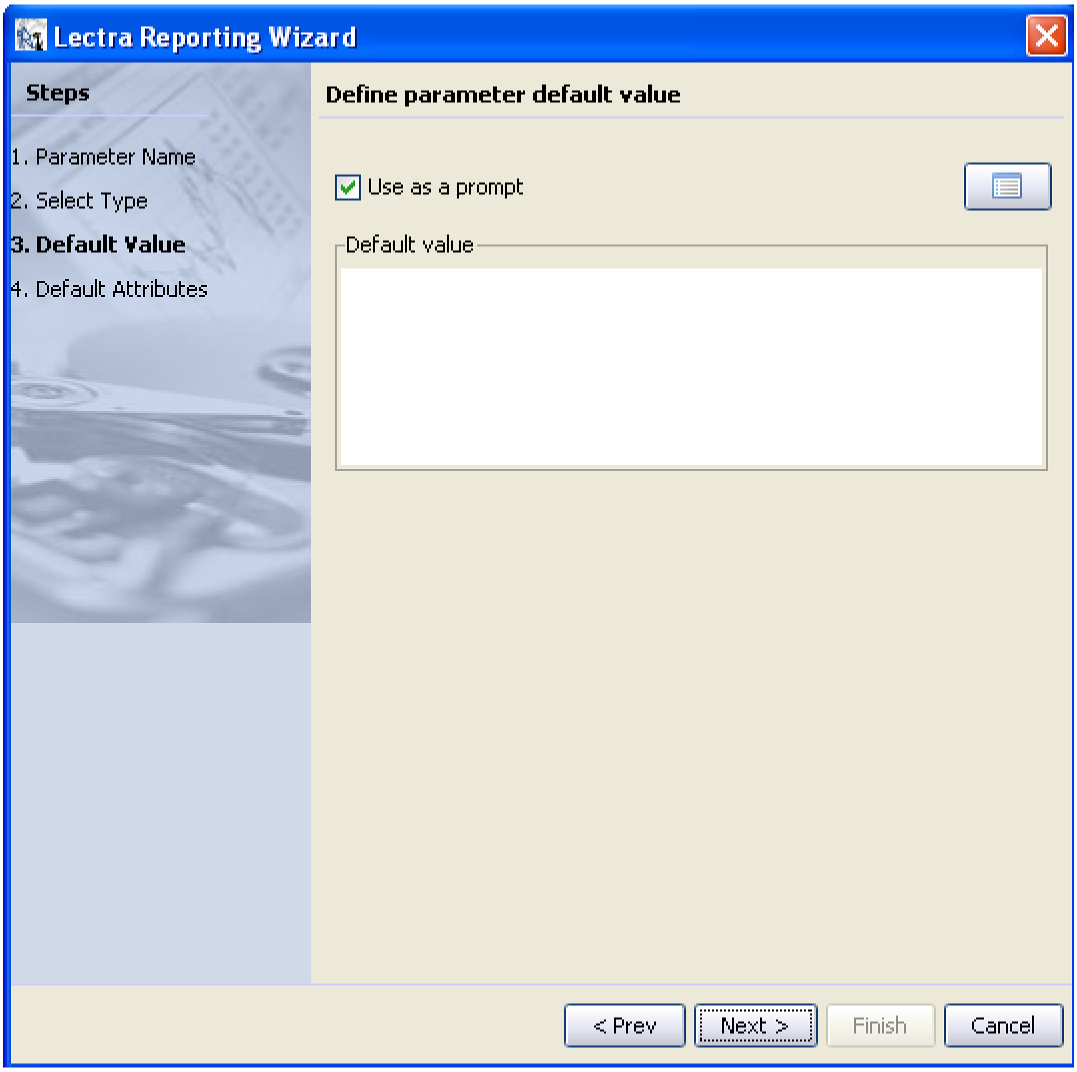
This page allows defining the type of the parameter.

Fields :

- Optional parameter : check this box in order to have an optional parameter (the parameter can have no values). Only prompted parameters can be optional.
- Is an array: if parameter is a set of values check this box.
- PDM type: All Available elements of FQN (PDM model).
- JAVA type : Integer, String, Date, Double, Float, Long, Short, BigDecimal, Time, Timestamp, InputStream

To select an element, you can drag and drop or click on .

10.2 Define parameter default value



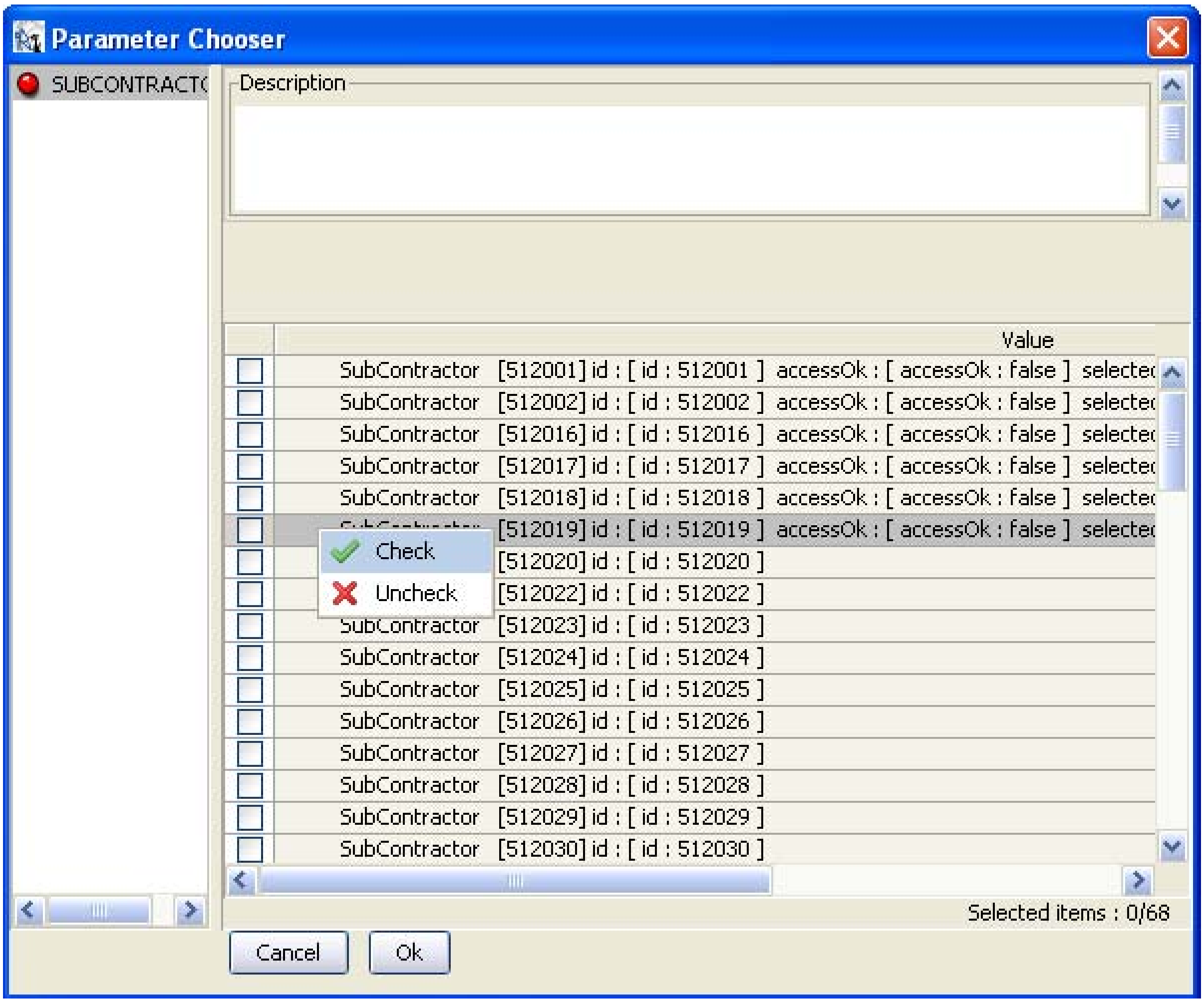
Define parameter default value

☑ Use as a prompt

Default value

Fields :

- Use as a prompt: if this box is checked, parameter value will be asked when report will be executed. For optional parameter, the “use as a prompted” field is automatically checked.
- Default value: value used if parameter is not prompted (separated with comma if the parameter is an array).
- Open instance selection or when executing a report with prompted parameters:



	Value
□	SubContractor [512001] id : [id : 512001] accessOk : [accessOk : false] selectec
□	SubContractor [512002] id : [id : 512002] accessOk : [accessOk : false] selectec
□	SubContractor [512016] id : [id : 512016] accessOk : [accessOk : false] selectec
□	SubContractor [512017] id : [id : 512017] accessOk : [accessOk : false] selectec
□	SubContractor [512018] id : [id : 512018] accessOk : [accessOk : false] selectec
□	[512019] id : [id : 512019] accessOk : [accessOk : false] selectec
□	[512020] id : [id : 512020]
□	[512022] id : [id : 512022]
□	SubContractor [512023] id : [id : 512023]
□	SubContractor [512024] id : [id : 512024]
□	SubContractor [512025] id : [id : 512025]
□	SubContractor [512026] id : [id : 512026]
□	SubContractor [512027] id : [id : 512027]
□	SubContractor [512028] id : [id : 512028]
□	SubContractor [512029] id : [id : 512029]
□	SubContractor [512030] id : [id : 512030]

Selected items : 0/68

☺If parameter is an array you can select several lines and with right click select ‘check’ or ‘uncheck’ options.

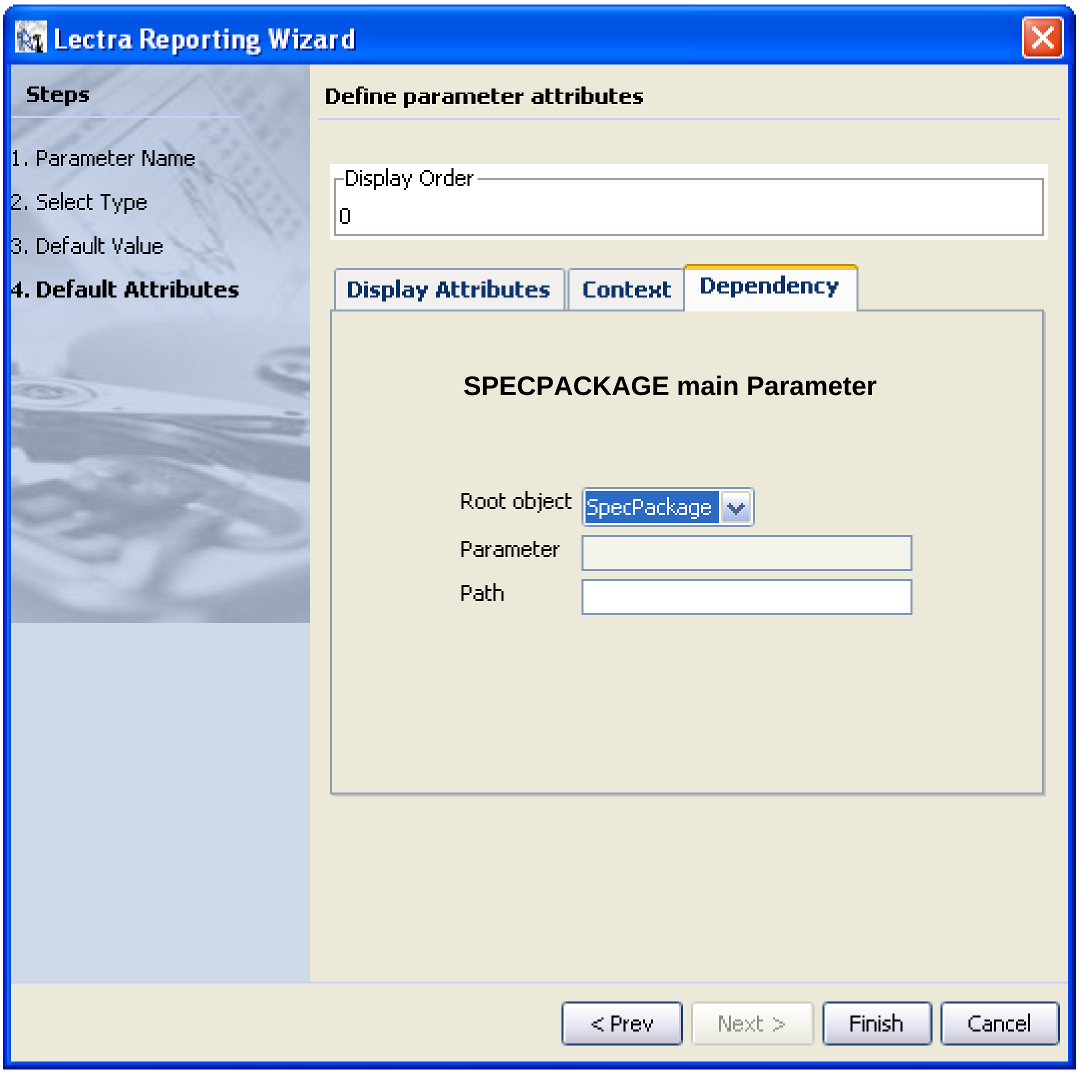
If the parameter is optional, you can select no values.

10.3 Define Parameter Attributes

This part of wizard is used in order to define general behaviour of the parameter:

- Dependency with context or to other parameters.
- Display rules.

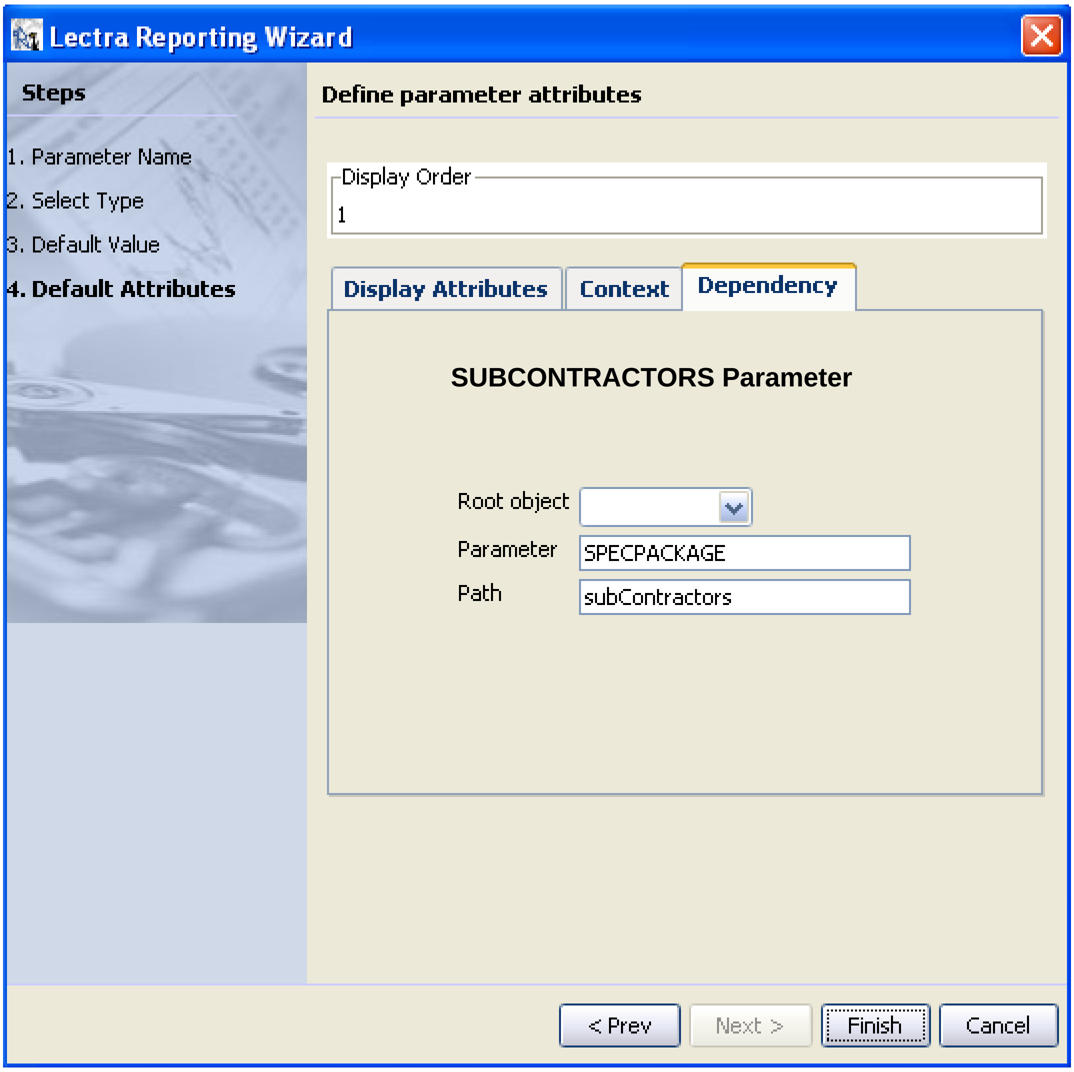
10.3.1Dependency



Display Order
0

SPECPACKAGE main Parameter

Root object: SpecPackage
Parameter:
Path:



Define parameter attributes

Display Order: 1

SUBCONTRACTORS Parameter

Root object:

Parameter: SPECPACKAGE

Path: subContractors

Fields:

– Root objects dependency:

The root object dependency is used only for main parameters (no prompted parameters). It permits selecting the type of the root object.

If this field is filled, the other one are empty.

– Parameter: Use for dependent parameter.

If the parameter depends on another one, set the name of the main parameter.

– Path: Use for dependent parameter

Enter the path to access from the root object to the parameter object.

Path syntax:

role1.role2.role3

It is the path to navigate for the root object to the parameter object.

If you have to filter on a role because it returns several elements, you can use this syntax:

role:fieldname=fieldvalue

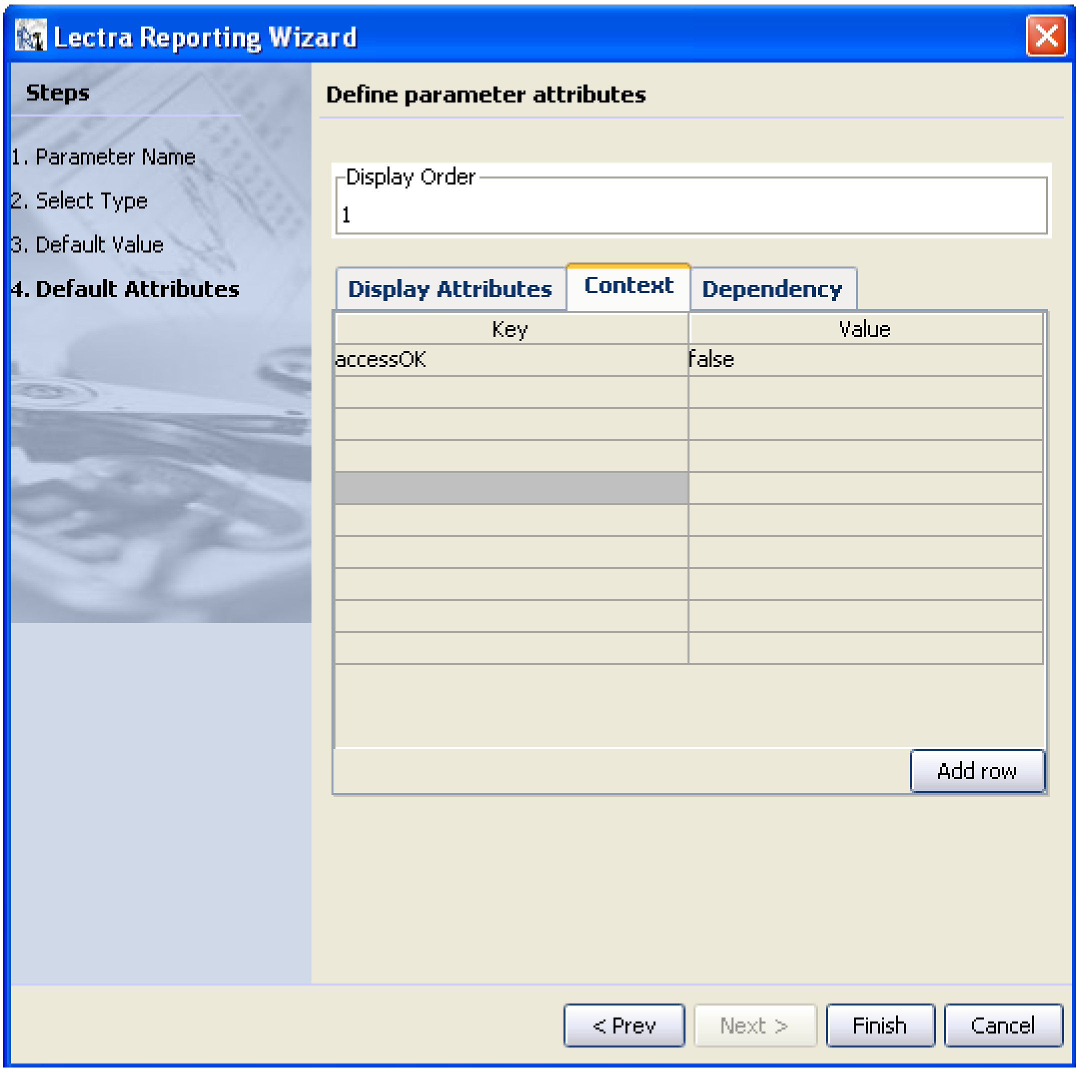
for example : In a spec package, we have many specTables and you parameter focuses only on the specTable named “StyleCostingProductionColorSize”. So you path element can be written like:

specTable:specTableDefName=StyleCostingProductionColorSize.spLinks

NB: If no parameter is entered, the possible value of the parameter is all objects with the parameter type.

10.3.2Context

The context tab permits applying filters on elements returned by dependency.



Display Order: 1

Key	Value
accessOK	false

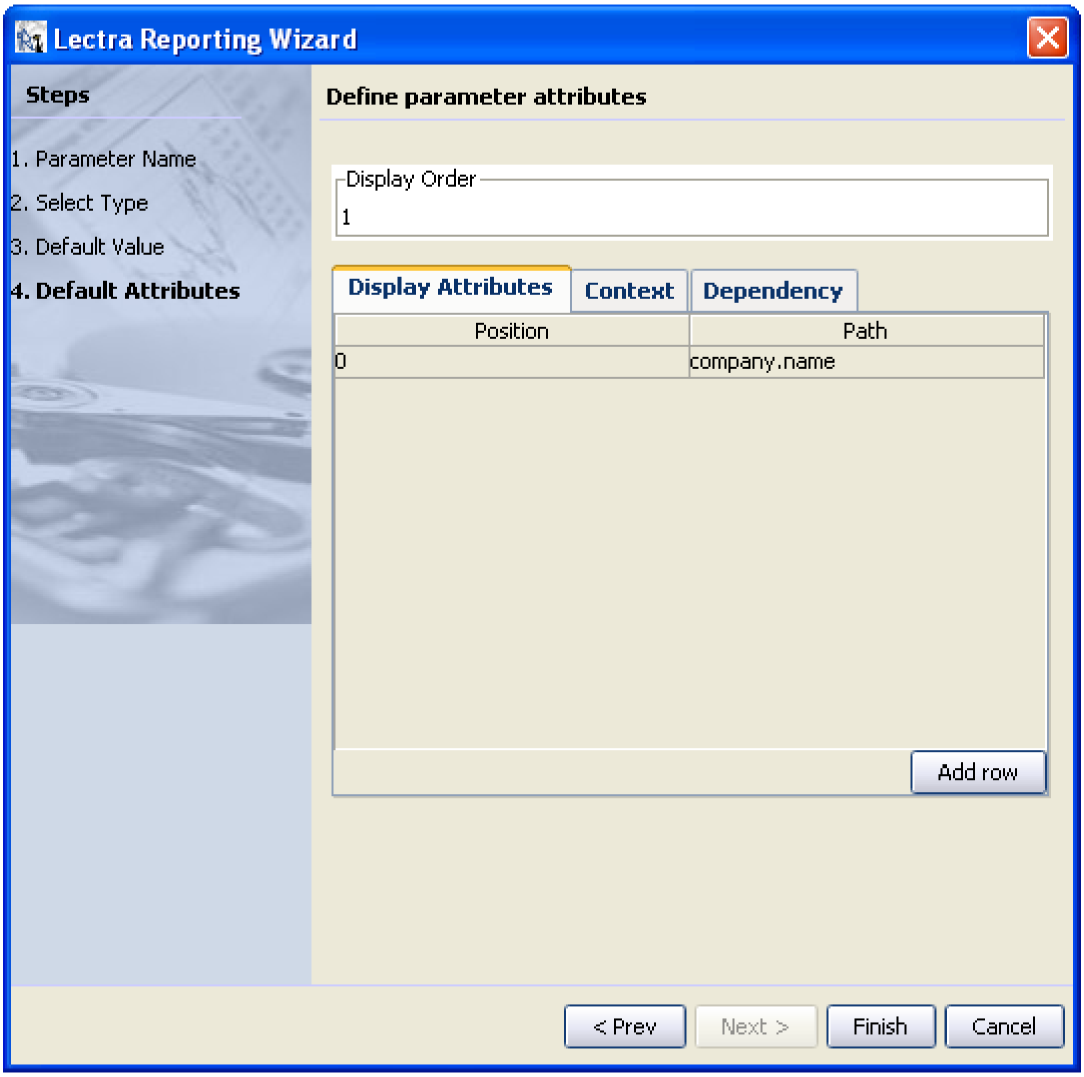
Fields:

- Key: Attribute name on which the filter is applied.
- Value: selected value on this field.

NB: Several filters can be applied with an “AND” condition between them.

10.3.3Display attributes

The context tab permits to define what to display to the user for parameter possible values.



Display Order: 1

Position	Path
0	company.name

Fields:

- Position: display position of attribute/ Integer from 0, 1, 2, etc.
- Path : path to the field to display

At the end of parameter configuration, click on finish.

10.4 Important Points

- Prompted parameters must be in the context.
- If a parameter is prompted, then dependency properties are mandatory.
- You cannot print multiple items reports with parameters if the root parameter type different of [].

11. REPORT PARAMETER EXAMPLES

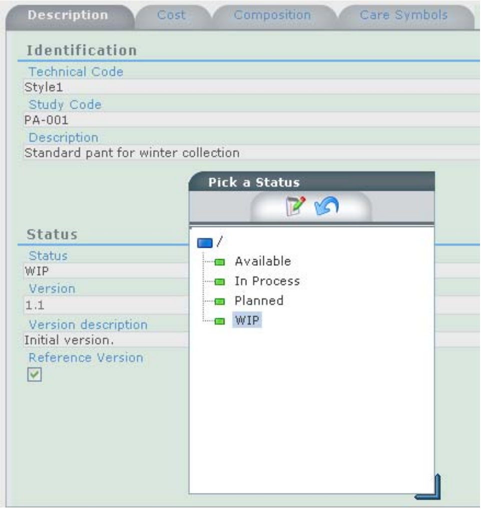
This part aims at giving parameter configuration examples.

11.1 Display all style status

The report needs 2 parameters:

- A selected style,
- The status of the style.

When the user wants to generate this report, we want to give him all available status.



Identification

Technical Code
Style1

Study Code
PA-001

Description
Standard pant for winter collection

Status

Status
WIP

Version
1.1

Version description
Initial version.

Reference Version
☑

PRODUCT parameter		
Parameter name step	Name	PRODUCT
	Description	Style
Select type step	Is an array	yes
	Parameter type	{productmanagement.process.Product}
Default value step	Use as a prompt	no

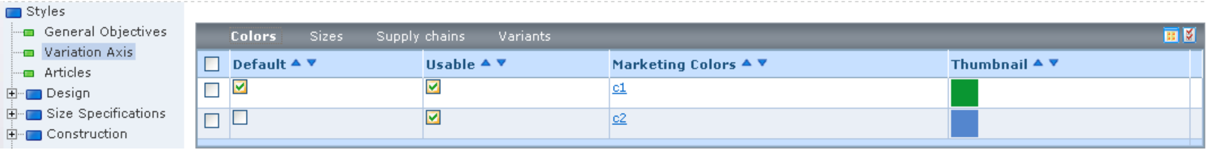
		Default value	You can put a list of default values
Default attributes step	Common	Display order	0
	Display attributes	Position	▨
		Path	▨
	Context	Key	▨
		Value	▨
	Dependency	Root object	Product
		Parameter	▨
		Path	▨
STATUS parameter			
Parameter name step		Name	STATUS
		Description	Status of the style
Select type step		Is an array	no
		Parameter type	{core.property}
Default value step		Use as a prompt	yes
		Default value	▨
Default attributes step	Common	Display order	1
	Display attributes	Position	0
		Path	path
	Context	Key	propertyDefName
		Value	DevStatus
		Key	propertyDefFamilly
		Value	process
	Dependency	Root object	Property
		Parameter	▨
		Path	▨

11.2 Display marketing colors in a product variation axes

The report needs 2 parameters:

- A selected product,
- The marketing colors that depends from the selected product.

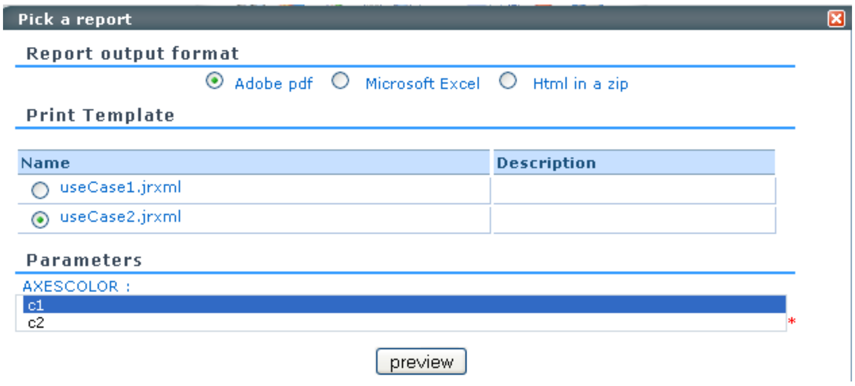
When the user wants to generate this report, we want to give him all only marketing colors available in the product axes.



☐	Default	Usable	Marketing Colors	Thumbnail
☐	☑	☑	c1	🟩
☐	☐	☑	c2	🟦

PRODUCT parameter			
Parameter name step		Name	PRODUCT
		Description	Style
Select type step		Is an array	yes
		Parameter type	{productmanagement.process.Product}
Default value step		Use as a prompt	no
		Default value	You can put a list of default values
Default attributes step	Common	Display order	0
	Display attributes	Position	▨
		Path	▨
	Context	Key	▨
		Value	▨
	Dependency	Root object	Product
		Parameter	▨
		Path	▨
AXE_COLOR parameter			
Parameter name step		Name	AXE_COLOR
		Description	Marketing colors in the product

Select type step		Is an array	yes
		Parameter type	{productmanagement.process.AxeValue}
Default value step		Use as a prompt	yes
		Default value	▨
Default attributes step	Common	Display order	1
	Display attributes	Position	0
		Path	Axifyable.name
	Context	Key	axeName
		Value	Style|||MarketingColor
	Dependency	Root object	▨
		Parameter	PRODUCT
		Path	axes



Pick a report

Report output format

● Adobe pdf ○ Microsoft Excel ○ Html in a zip

Print Template

Name	Description
○ useCase1.jrxml	
● useCase2.jrxml	

Parameters

AXESCOLOR :

c1

c2

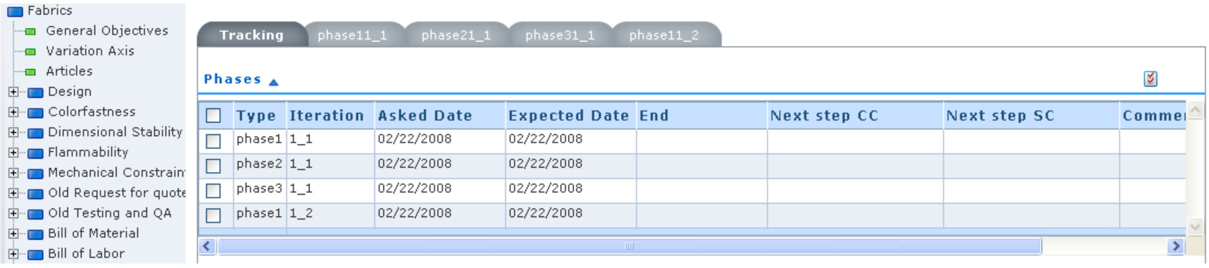
*

11.3 Display all Fabric testing and Qa phases

We want to display in parameters pop up all created phase with phase name and phase iteration (phase name is a CustomRoles and iteration a customFields).

The report needs 2 parameters:

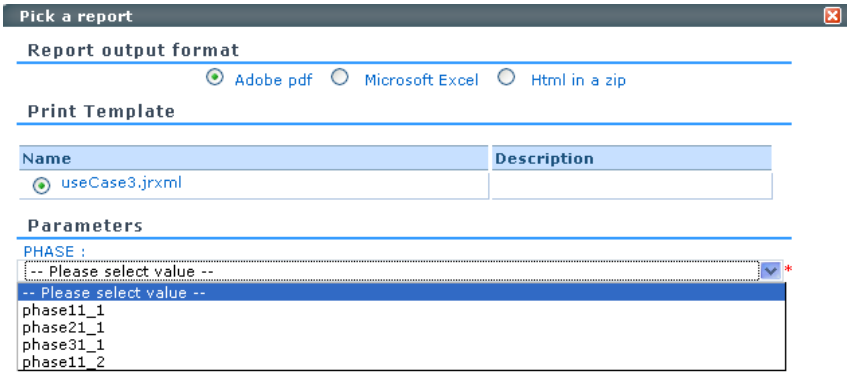
- A selected spec_package,
- A phase selected from all available in the selected spec_package.



□	Type	Iteration	Asked Date	Expected Date	End	Next step CC	Next step SC	Comme
□	phase1	1_1	02/22/2008	02/22/2008				
□	phase2	1_1	02/22/2008	02/22/2008				
□	phase3	1_1	02/22/2008	02/22/2008				
□	phase1	1_2	02/22/2008	02/22/2008				

SPECPACKAGE parameter			
Parameter name step		Name	SPECPACKAGE
		Description	Spec package
Select type step		Is an array	no
		Parameter type	{productmanagement.process.SpecPackage}
Default value step		Use as a prompt	no
		Default value	You can put a default value
Default attributes step	Common	Display order	0
	Display attributes	Position	▨
		Path	▨
	Context	Key	▨
		Value	▨
	Dependency	Root object	SpecPackage
		Parameter	▨
		Path	▨
PHASE parameter			
Parameter name step		Name	PHASE
		Description	phase
Select type step		Is an array	no
		Parameter type	{productmanagement.process.SpecTable}
Default value step		Use as a prompt	yes

		Default value	▨
Default attributes step	Common	Display order	1
	Display attributes	Position	0
		Path	customRoles:phase.targets.name
		Position	1
		Path	customFields:alias_iter
	Context	Key	specTableDefName
		Value	Fabric|||TestingAndQANG|||Stage
	Dependency	Root object	▨
		Parameter	SPECPACKAGE
		Path	specTables



Pick a report

Report output format

● Adobe pdf ○ Microsoft Excel ○ Html in a zip

Print Template

Name	Description
● useCase3.jrxml	

Parameters

PHASE :

-- Please select value -- *

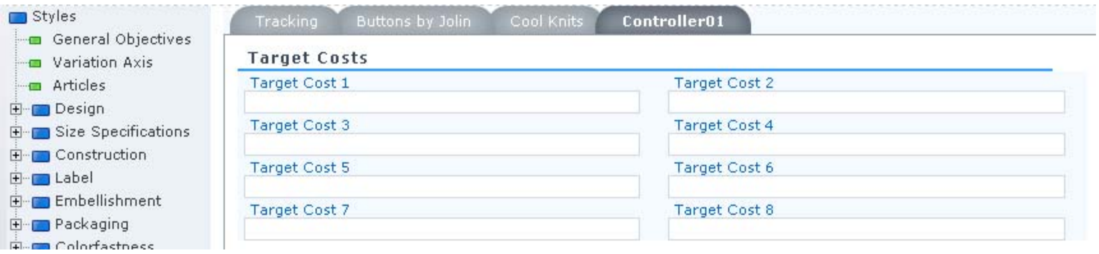
-- Please select value --
phase11_1
phase21_1
phase31_1
phase11_2

11.4 Display all RFQ suppliers

We want to display in parameters pop up the name of all companies, a company is associated to a specPackage and a subcontractor.

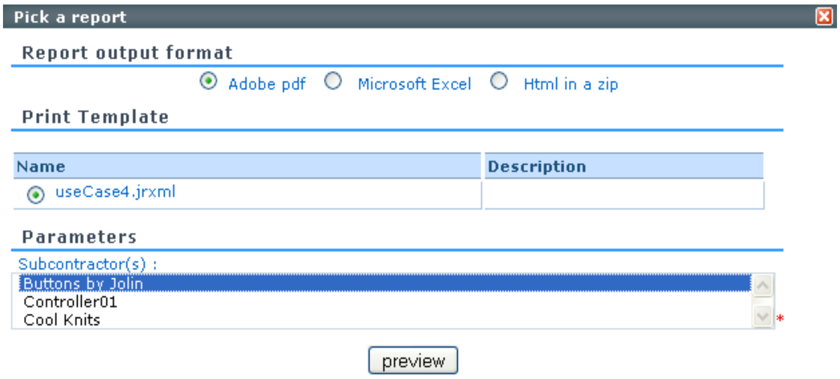
The report needs 2 parameters:

- A selected spec_package,
- All the companies linked in the subcontractor.



SPECPACKAGE parameter			
Parameter name step		Name	SPECPACKAGE
		Description	Spec package
Select type step		Is an array	no
		Parameter type	{productmanagement.process.SpecPackage}
Default value step		Use as a prompt	no
		Default value	You can put a default value
Default attributes step	Common	Display order	0
	Display attributes	Position	▨
		Path	▨
	Context	Key	▨
		Value	▨
	Dependency	Root object	SpecPackage
		Parameter	▨
		Path	▨
SUBCONTRACTORS parameter			
Parameter name step		Name	SUBCONTRACTORS
		Description	Suupliers available in the specpackage.
Select type step		Is an array	no
		Parameter type	{productmanagement.process.SubContractor}
Default value step		Use as a prompt	yes

		Default value	▨
Default attributes step	Common	Display order	1
	Display attributes	Position	0
		Path	company.name
	Context	Key	▨
		Value	▨
	Dependency	Root object	▨
		Parameter	SPECPACKAGE
		Path	subContractors



Pick a report

Report output format

● Adobe pdf ○ Microsoft Excel ○ Html in a zip

Print Template

Name	Description
● useCase4.jrxml	

Parameters

Subcontractor(s) :

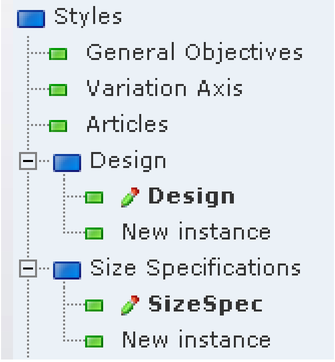
Buttons by Jolin
Controller01
Cool Knits

11.5 Display all specPackage from a style

We want to display in parameters pop up all the spec package instance associated to a Style.

The report needs 2 parameters:

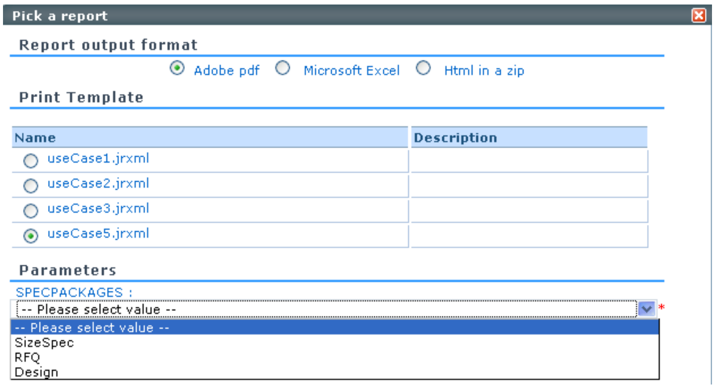
- A selected style,
- The spec package get from the list of style spec packages.



Styles
General Objectives
Variation Axis
Articles
Design
Design
New instance
Size Specifications
SizeSpec
New instance

PRODUCT parameter			
Parameter name step		Name	PRODUCT
		Description	Style
Select type step		Is an array	no
		Parameter type	{productmanagement.process.Product}
Default value step		Use as a prompt	no
		Default value	You can put a default value
Default attributes step	Common	Display order	0
	Display attributes	Position	▨
		Path	▨
	Context	Key	▨
		Value	▨
	Dependency	Root object	Product
		Parameter	▨
		Path	▨
SPECPACKAGE parameter			
Parameter name step		Name	SPECPACKAGE
		Description	Spec package.
Select type step		Is an array	no
		Parameter type	{productmanagement.process.SpecPackage}
Default value step		Use as a prompt	yes
		Default value	▨
Default attributes step	Common	Display order	1
	Display attributes	Position	0
		Path	name

	Context	Key	▨
		Value	▨
	Dependency	Root object	▨
		Parameter	PRODUCT
		Path	specifications



Pick a report

Report output format

● Adobe pdf ○ Microsoft Excel ○ Html in a zip

Print Template

Name	Description
○ useCase1.jrxml	
○ useCase2.jrxml	
○ useCase3.jrxml	
● useCase5.jrxml	

Parameters

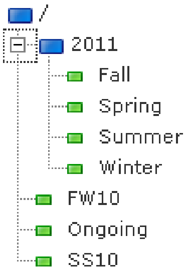
SPECPACKAGES :

-- Please select value -- *

-- Please select value --
SizeSpec
RFQ
Design

11.6 Display all seasons

We want to define a parameter that gets all available seasons.



SAISON parameter		
Parameter name step	Name	SAISON
	Description	Saison
Select type step	Is an array	no
	Parameter type	{core.Property}
Default value step	Use as a prompt	yes
	Default value	▨

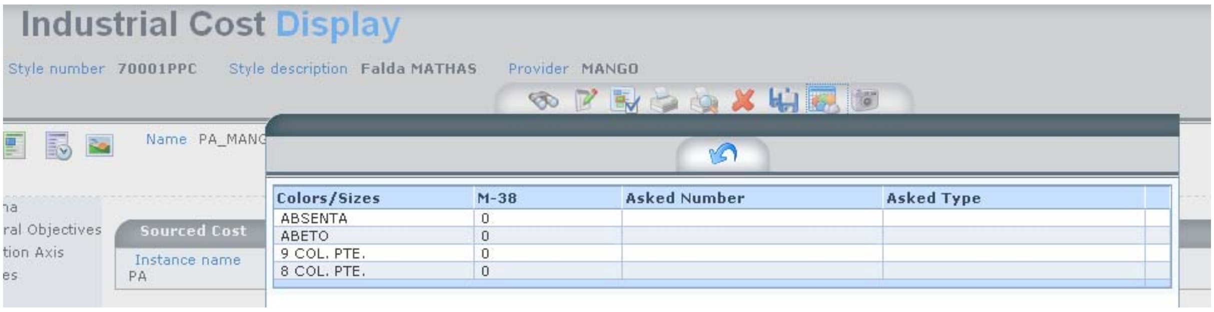
Default attributes step	Common	Display order	0
	Display attributes	Position	0
		Path	path
	Context	Key	propertyDefName
		Value	Season
		Key	propertyDefFamily
		Value	process
	Dependency	Root object	▨
		Parameter	▨
		Path	▨

11.7 Display all lines of a custom table

This is a custom example. We can have a custom table in the costing spec package and we want to define a parameter that takes a line of this custom table.

The report needs 2 parameters:

- A selected spec package,
- The lines available in this custom table.



Industrial Cost Display

Style number 70001PPC Style description Falda MATHAS Provider MANGO

Name PA_MANG

Sourced Cost

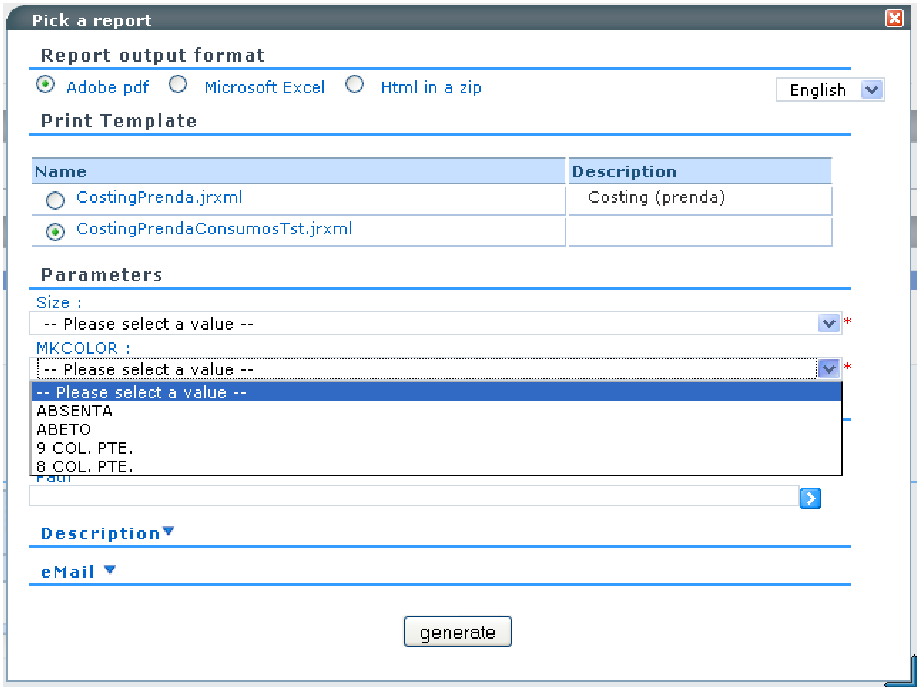
Instance name PA

Colors/Sizes	M-38	Asked Number	Asked Type
ABSENTA	0		
ABETO	0		
9 COL. PTE.	0		
8 COL. PTE.	0		

SPECPACKAGE parameter		
Parameter name step	Name	SPECPACKAGE
	Description	Spec package
Select type step	Is an array	no

		Parameter type	{productmanagement.processSpecPackage}
Default value step		Use as a prompt	no
		Default value	You can put a default value
Default attributes step	Common	Display order	0
	Display attributes	Position	▨
		Path	▨
	Context	Key	▨
		Value	▨
	Dependency	Root object	SpecPackage
		Parameter	▨
		Path	▨
CUSTP_LINE parameter			
Parameter name step		Name	CUSTO_LINE
		Description	▨
Select type step		Is an array	yes
		Parameter type	{productmanagement.process.SpLink}
Default value step		Use as a prompt	yes
		Default value	▨
Default attributes step	Common	Display order	1
	Display attributes	Position	0
		Path	customRoles:*CustomRoleName*.targets.name
		Position	1
		Path	customFields:*CustomFieldName*
	Context	Key	spLinkDefName
		Value	"Style|||CostingNG|||ProductionColorSize|||-" *Name of the spLink*

	Dependency	Root object	▨
		Parameter	SPECPACKAGE
		Path	specTables:specTableDefName=*specTableName*.spLinks



Pick a report

Report output format

● Adobe pdf ○ Microsoft Excel ○ Html in a zip English

Print Template

	Name	Description
○	CostingPrenda.jrxml	Costing (prenda)
●	CostingPrendaConsumosTst.jrxml	

Parameters

Size :
-- Please select a value -- *

MKCOLOR :
-- Please select a value -- *

-- Please select a value --
ABSENTA
ABETO
9 COL. PTE.
8 COL. PTE.

Path

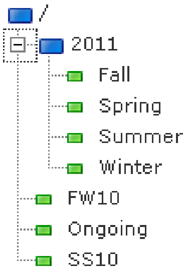
Description

eMail

generate

11.8 Optional parameter

We want to define an optional parameter on all available seasons.

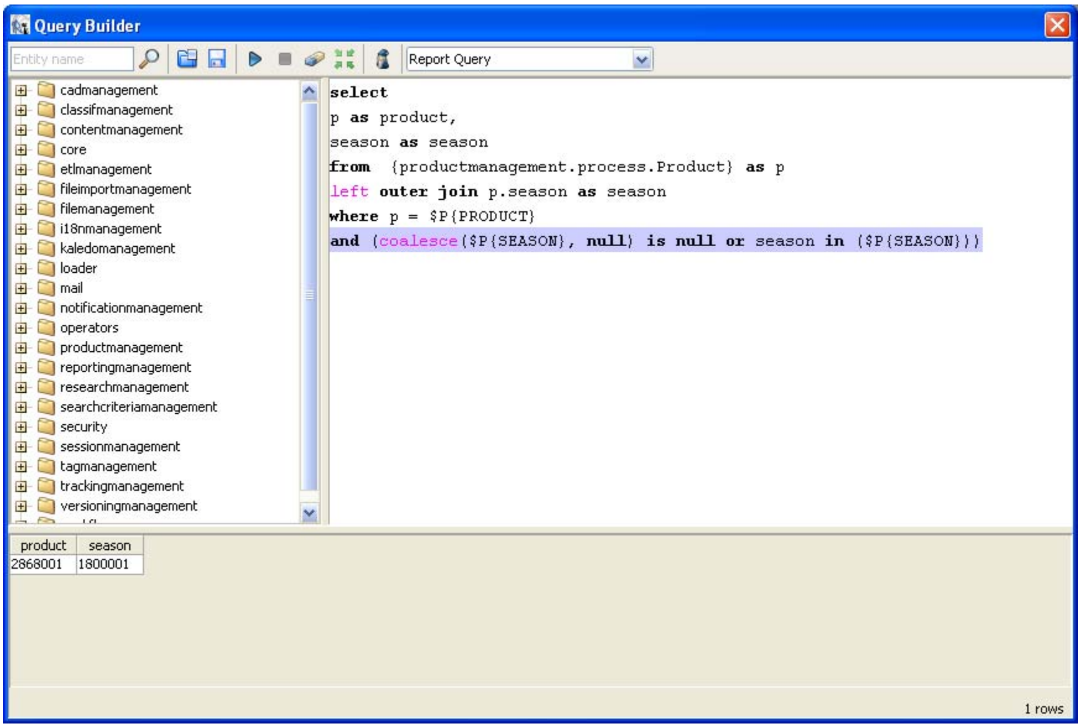


/
2011
Fall
Spring
Summer
Winter
FW10
Ongoing
SS10

SAISON parameter		
Parameter name step	Name	SAISON
	Description	Saison
Select type step	Is an array	no

		Optional parameter	**true**
		Parameter type	{core.Property}
Default value step		**Use as a prompt**	**yes**
		Default value	▨
Default attributes step	Common	Display order	0
	Display attributes	Position	0
		Path	path
	Context	Key	propertyDefName
		Value	Season
		Key	propertyDefFamily
		Value	process
	Dependency	Root object	▨
		Parameter	▨
		Path	▨

Impact on the report request :



```
select
p as product,
season as season
from  {productmanagement.process.Product} as p
left outer join p.season as season
where p = $P{PRODUCT}
and (coalesce($P{SEASON}, null) is null or season in ($P{SEASON}))
```

product	season
2868001	1800001

If the parameter is null, the request will return the product that corresponds to the selected product.

If the parameter has a value, the request will return the product that corresponds to the selected product and that refers to the selected saison.

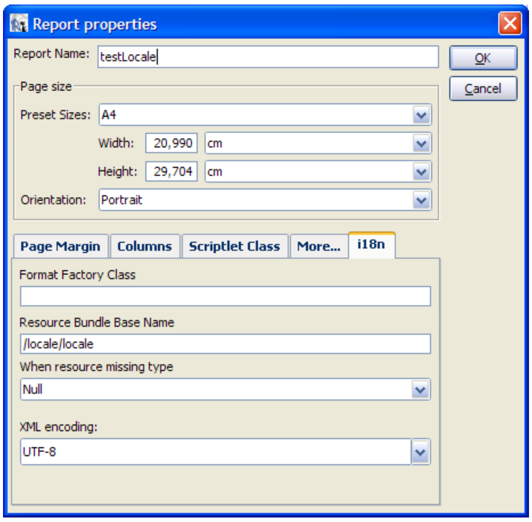
No value for an optional parameter -> no filter on the criteria.

12. LOCALIZATION

12.1 Report configuration for localization

Localization is compliant with Java standard. In order to use localization, user needs to define a properties file per language. Each file contains key-value pair, and this file is persisted in the FileManager.

In the report template, user declare only /<path>/<resource bundle name> in the report properties as follow.



Report Name: testLocale

Preset Sizes: A4

Width: 20,990 cm

Height: 29,704 cm

Orientation: Portrait

Format Factory Class

Resource Bundle Base Name: /locale/locale

When resource missing type: Null

XML encoding: UTF-8

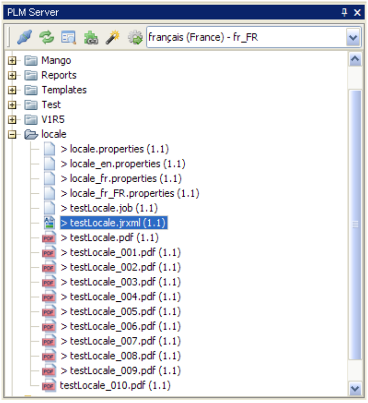
12.2 Localization files

Locale files are:

- /locale/locale.properties: default language
- /locale/locale_fr.properties: French language
- /locale/locale_en.properties: English language
- and so on …

At runtime, with the current locale, JasperReports found itself the localization file.

In iReport, the main toolbar of ReportDesigner plugin show a combo-box with all available locale. The format of the displayed name is <language name> - <language>_<country>_<variant>.


PLM Server
français (France) - fr_FR
Mango
Reports
Templates
Test
V1R5
locale
> locale.properties (1.1)
> locale_en.properties (1.1)
> locale_fr.properties (1.1)
> locale_fr_FR.properties (1.1)
> testLocale.job (1.1)
> testLocale.jrxml (1.1)
> testLocale.pdf (1.1)
> testLocale_001.pdf (1.1)
> testLocale_002.pdf (1.1)
> testLocale_003.pdf (1.1)
> testLocale_004.pdf (1.1)
> testLocale_005.pdf (1.1)
> testLocale_006.pdf (1.1)
> testLocale_007.pdf (1.1)
> testLocale_008.pdf (1.1)
> testLocale_009.pdf (1.1)
testLocale_010.pdf (1.1)

The resource bundle locale file name is

<resource bundle name>_<language>_<country>_<variant>.properties

Localization file example

<u>Locale_en.properties example :</u>

Collection = Collection

Color = Color

Season = Season

Style = Style

Target = Target

Supplier = Supplier

Fabric = Fabric

Description = Description

<u>Locale_fr.properties example :</u>

Collection = Collection

Color = Couleur

Season = Saison

Style = Modèle

Target = Objectif

Supplier = Fournisseur

Fabric = Tissus

Description = Description

12.3 Localization expression in reports

To use a localisation key, use an text field and insert in the field expression the $V{} tag.

Text field Example

Use the expression into the text field $R{Color} to print Color in english, Couleur in French.

12.4 Localization and pdf fonts

The European fonts like Arial are not compatible with Asiatic languages.

If reports should be compatible with European and Asiatic languages, the styles library provides styles that manage the font according to the selected language:

- font : for classic text
- fontBold : for bold text
- fontItalic : for italic text
- fontBoldItalic : for bold and italic text.

This 4 styles check the locale and select the corresponded font.

- For japanese locale the selected font is :

  fontName = Arial

  fontPdfName = HeiseiKakuGo-W5

  encoding = UniJIS-UCS2-H (Japanese)

- For traditional chinese the selected font is :

  fontName = Arial

  fontPdfName = Mhei-Medium

  encoding = UniCNS-UCS2-H (Chinese traditional)

- For simplified chinese the selected font is :

  fontName = Arial

  fontPdfName = STSong-Light

  encoding = UniGB-UCS2-H (Chinese Simplified)

- For korean the selected font is :

  fontName = Arial

  fontPdfName = HYGoThic-Medium

encoding = UniKS-UCS2-H (Korean)

- For all other languages the selected font is :

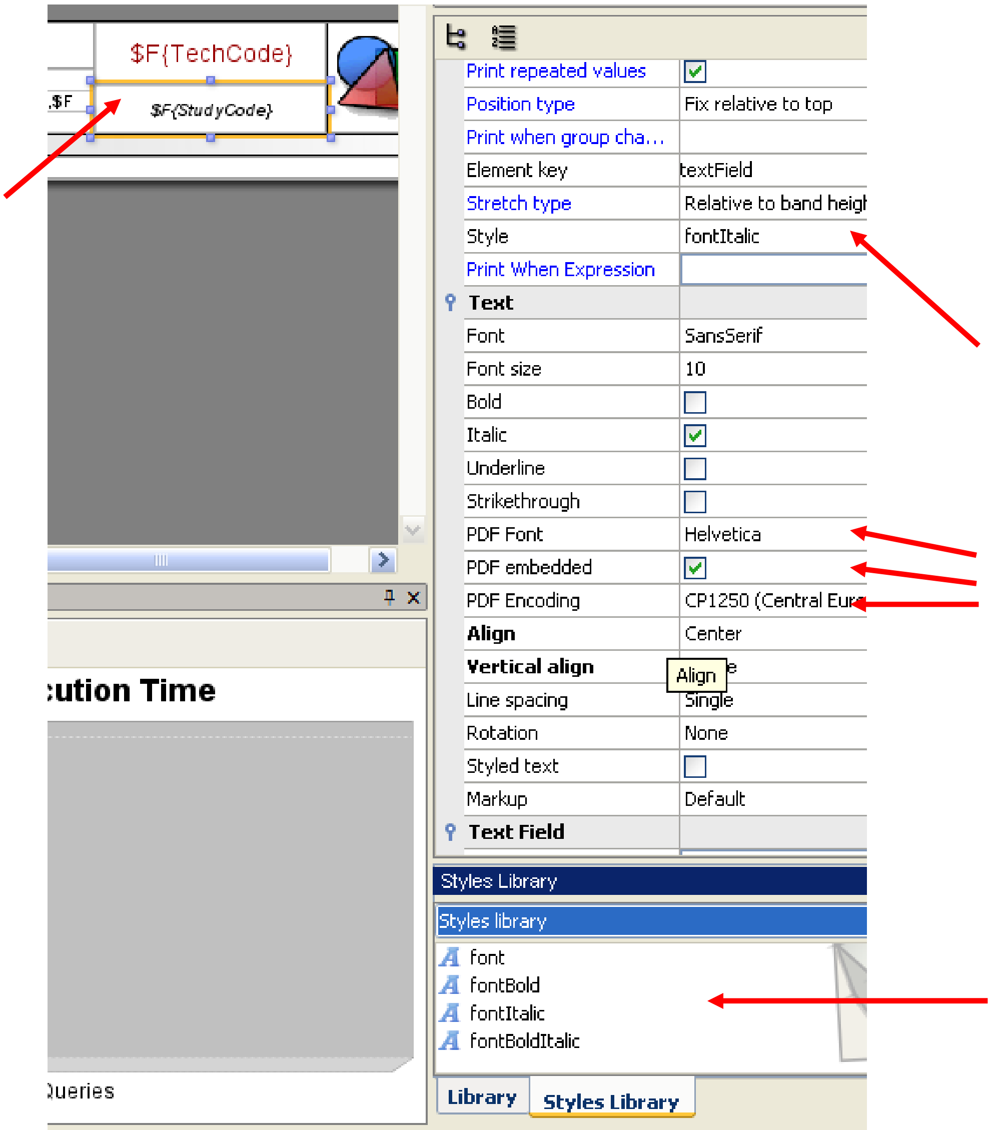
  fontName = Arial

  fontPdfName = Arial (arial.ttf)

  encoding = CP1252 (Western European ANSI aka WinAnsi)

For each text field or static text:

- Do not select the font with the fields : PDF font, PDF embedded and PDF encoding
- Click on the style field and select one of the 4 predefined styles available in the styles library.



Selected text field

Style for italic text

Font fields not used

Available font that manage european and asiatic languages

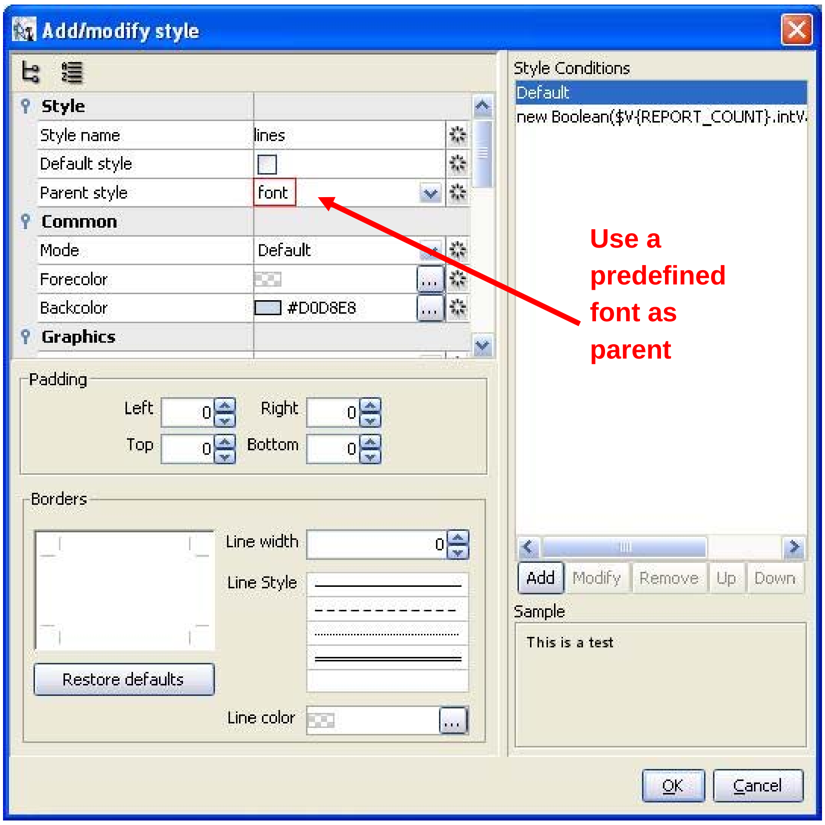
Print repeated values	☑
Position type	Fix relative to top
Print when group cha...	
Element key	textField
Stretch type	Relative to band heigl
Style	fontItalic
Print When Expression	
Text	
Font	SansSerif
Font size	10
Bold	☐
Italic	☑
Underline	☐
Strikethrough	☐
PDF Font	Helvetica
PDF embedded	☑
PDF Encoding	CP1250 (Central Eur
Align	Center
Vertical align	Align e
Line spacing	Single
Rotation	None
Styled text	☐
Markup	Default
Text Field	

Styles Library

Styles library

- font
- fontBold
- fontItalic
- fontBoldItalic

NB: If you need to create a style for text fields, you should use one of the 4 predefines styles as the parent.



Style name: lines

Default style: □

Parent style: font

Mode: Default

Forecolor:

Backcolor: #D0D8E8

Left 0 Right 0

Top 0 Bottom 0

Line width 0

Style Conditions:

Default

new Boolean($V{REPORT_COUNT}.intV.

This is a test

Use a predefined font as parent