

Auteur :	Cyril PESTANA
Mise en application :	29/09/2011



iReport

 JasperForge.org

sylob

Sommaire

I	INTRODUCTION.....	2
II	PROCESSUS DE PUBLICATION	3
III	ETAPES DE MODIFICATION D'UN ÉTAT STANDARD	4
1.	TÉLÉCHARGEMENT D'UN ETAT	4
2.	MODIFICATION DE L'ETAT	5
3.	CHARGEMENT D'UN ETAT	7
IV	NORME XML	9
V	PARAMÉTRAGE D'IREPORT	10
1.	INSTALLATION IREPORT : 2 PARAMÈTRAGES	10
2.	MODIFICATION D'UNE MAQUETTE : 2 PARAMÈTRAGES	12
VI	UTILISATION D'IREPORT	13
1.	STRUCTURE DU RAPPORT	13
2.	PROPRIÉTÉS DU RAPPORT	15
3.	ELEMENT DU RAPPORT	18
4.	CHAMP, PARAMÈTRE ET VARIABLE	25
5.	CLÉ MESSAGE	30
6.	OPTION DE GROUPE ET DE TRI	31
7.	EDITEUR D'EXPRESSION.....	33
VII	CRÉATION D'UNE ACTIVITÉ DE PUBLICATION	34
VIII	EXPRESSION JAVA.....	35
IX	BIBLIOTHÈQUE DE MÉTHODES.....	39
X	RÉFÉRENCES.....	40

I Introduction

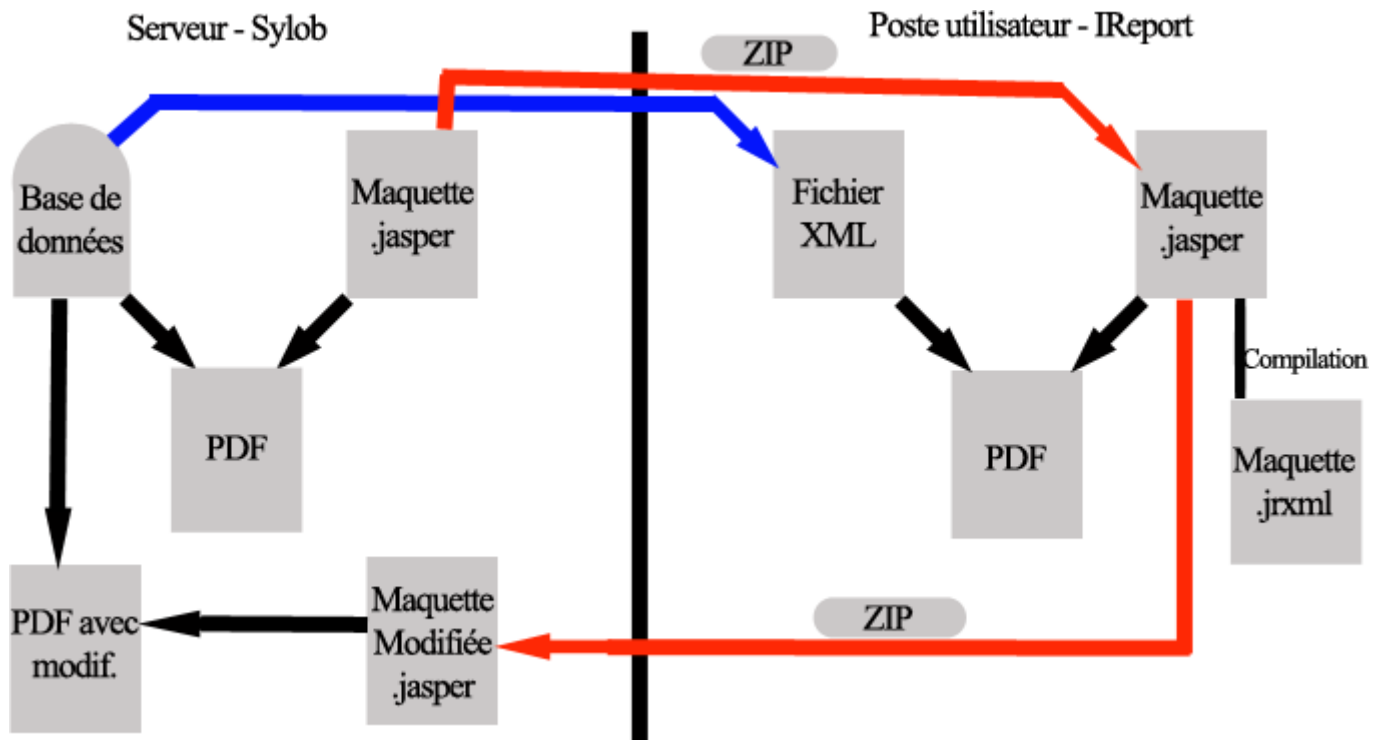
Sylob intègre plus de 130 états standards déjà paramétrés, qui couvrent tous les domaines de l'entreprise : Vendre, Produire, Acheter, ... Chacune de ces maquettes est personnalisable par modification d'états standards.

IReport est l'outil de génération d'états et de rapports qui permet de modifier et de finaliser les maquettes proposées par Sylob. Cet outil ne demande pas d'être un pro du langage SQL. La construction et la modification d'états se font par le biais de fichiers XML. La navigation se fait dans un graphe structuré dans lequel les éléments sont clairement définies et parlants comme dans l'explorateur Windows.

De plus avec IReport, toutes les informations statiques et dynamiques apparaitront dans la langue du client, avec le formatage des données spécifiques aux particularités des pays.

Enfin, IReport est un outil gratuit que Sylob a personnalisé afin qu'il soit facile d'utilisation pour les personnes n'ayant pas de connaissances profondes en informatique.

II Processus de publication



Le processus de publication sur le serveur Sylob et lors de l'utilisation d'IReport est semblable.

Serveur-Sylob : Lorsque l'utilisateur clique sur un bouton « Publier », le système récupère de la base de données l'élément sélectionné. Il retourne ensuite la maquette correspondante à la publication. L'élément de la base de données joint à la maquette permet le remplissage du PDF.

Poste utilisateur – IReport : Lors de la modification d'une maquette avec IReport, les éléments proviennent d'un fichier de type XML contenant toutes les données nécessaires à la publication. Ce fichier doit être téléchargé de Sylob sur le poste de l'utilisateur. L'utilisateur doit aussi récupérer la maquette correspondante stockée sur le serveur. Quand ces téléchargements sont effectués, les données présentes dans le fichier XML joints à la maquette permettent le remplissage du PDF.

Le téléchargement et le chargement de la maquette s'effectue par l'intermédiaire d'un fichier ZIP.

L'extension d'une maquette est de type jasper. Le fichier avec l'extension jrxml est la retranscription du fichier jasper dans IReport. Les modifications sur la maquette avec IReport s'effectuent sur le fichier jrxml. Pour transformer le fichier jrxml en jasper, une étape de compilation est nécessaire.

III Etapes de modification d'un état standard

Cette partie est un tutoriel qui détaille précisément toutes les étapes pour télécharger un état standard de Sylob, effectuer une modification sur la maquette avec IReport et charger la nouvelle maquette sur Sylob.

1. Téléchargement d'un état

Exemple : Modification de la maquette des factures immédiates :

1. Ouvrir l'activité « Gérer les paramètres de publications » qui se trouve dans le menu « Gérer Entreprise – Personnalisation - Document ». Cette activité propose la liste des activités qui propose une édition. Chaque édition est personnalisable.
 2. Modifier le paramétrage état de l'activité « Publier une facture client immédiate ».
- Voici l'écran obtenu.

Paramètres des publications

Gérer les paramètres des publications

Modifier le paramétrage de la publication pour l'activité "Publier une facture client immédiate"

Format de sortie : XML

Domaine : dm2 - Gérer les ventes

Liste des groupes de feuilles de style

Ajouter un groupe...

Libellé	Personnalisé	Action
Par défaut	Non	

[1 - 1 sur 1]

Informations à intégrer dans la source de données XML

Pour un niveau d'arborescence de : quatre


Facture immédiate client : Choisir un élément

Publier au format XML ...

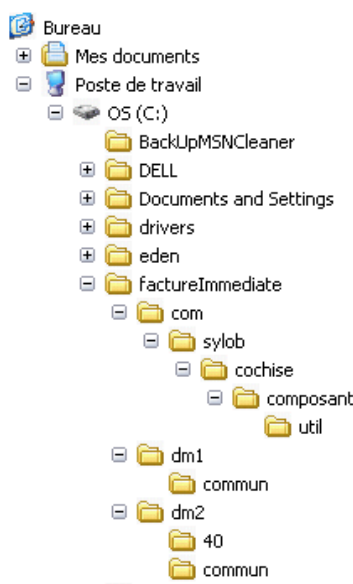
Valider Annuler

Choisir XML comme type de sortie

Choisir quatre ou cinq comme niveau d'arborescence

3. Sélectionner « XML » comme format de sortie.
4. Sélectionner « quatre » ou « cinq » comme niveau d'arborescence. Cette option permet de définir la taille du fichier XML. Plus le niveau est élevé, plus il y aura d'information dans le fichier XML et plus le temps de génération du fichier XML sera long.
Le bouton « Afficher l'arborescence » propose l'ensemble des attributs présents dans le fichier XML. Cette option est très utile afin de retrouver les informations dans le fichier XML.
5. Cliquer sur l'icône  pour télécharger l'ensemble des fichiers utilisés pour la publication de la facture immédiate. Tous ces fichiers sont regroupés dans un seul fichier compressé.
6. Enregistrer le fichier sur « C : » avec le nom pré-chargé « factureImmédiate.zip ».
7. Cliquer sur le bouton « Valider » pour enregistrer les modifications concernant ce paramétrage de publication.

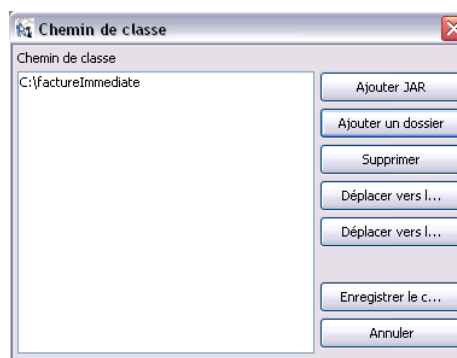
8. Ouvrir l'activité « Publier une facture client immédiate », sélectionner une facture immédiate et cliquer sur « Publier ». Une fenêtre du navigateur s'ouvre avec les données sélectionnées au format XML de votre facture immédiate. Ce traitement peut prendre quelques instants si le niveau d'arborescence sélectionné est élevé.
9. Cliquer sur « Fichier », « Enregistrer sous », donner le nom de « FactureImmédiateVente.XML » et enregistrer ce document sur « C : ».
10. Dézipper le contenu du fichier « factureImmédiate.zip » qui se trouve sur « C : » dans le répertoire suivant « C:/factureImmédiate ». Le créer si nécessaire. Ci-dessous l'architecture obtenue.
Sylob conseille d'utiliser l'utilitaire gratuit de compression et de décompression de fichiers « WinRar ».



2. Modification de l'etat

Sylob a personnalisé l'utilitaire IReport afin qu'il soit facile d'utilisation. Contacter le support de Sylob support@sylob.com afin d'obtenir la dernière version d'IReport. La dernière version se trouve également sur le dernier CD de mise à jour.

1. Cliquer sur « IReport-2.0.5 » et cliquer sur « iReport.exe ». IReport se lance.
2. Lors de la première utilisation, deux paramètres sont à renseigner
 - Cliquer sur « Créer » et sélectionner « Aperçu JRViewver ».
 - Cliquer sur « Options », « Paramètres », « Compiler ».Cocher la case « utiliser le répertoire de rapport à compiler »
Décocher la case « conserver le fichier .java ».
3. Lors d'une modification d'un nouvel état, deux paramètres sont à renseigner
 - Cliquer sur « Options », « Chemin de classe », « Ajouter un dossier ». Sélectionner le dossier suivant « C:/factureImmédiate ». Cliquer sur « Enregistrer le chemin de classe ». Vérifier que seul le dossier figure dans la liste comme indiqué sur l'écran ci-dessous. Supprimer les autres si nécessaire.



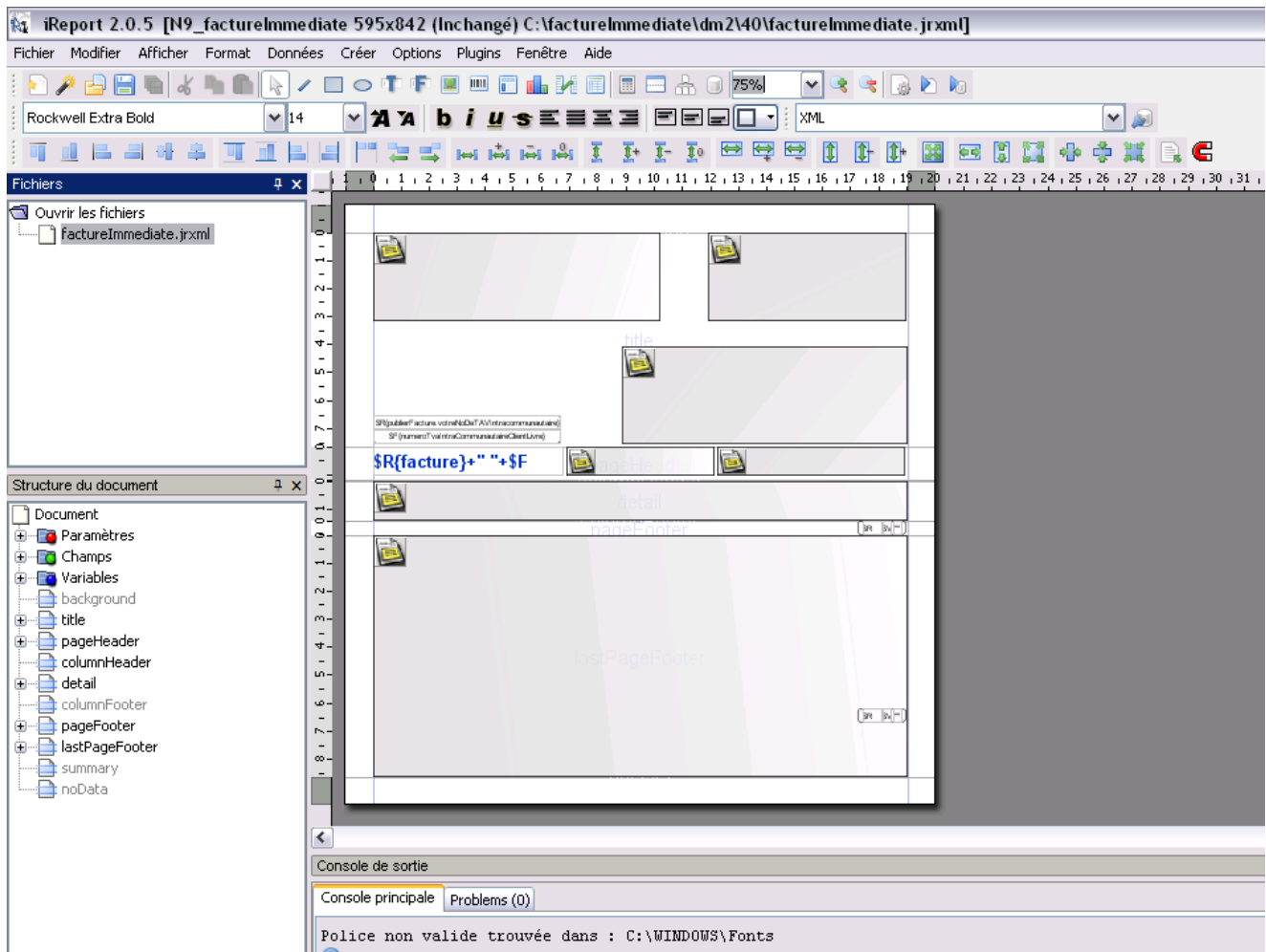
- Cliquer sur « Données », « Connections », « Nouveau » et sélectionner « Sylob datasource ». Saisir l'écran ci-dessous et cliquer sur « Enregistrer ». L'expression de base pour IReport, ici « root » est spécifiée sur Sylob dans l'activité « Gérer les paramètres état ». L'expression de base varie selon les états. L'expression « root » est utilisée pour la plupart des états proposés par Sylob.



The screenshot shows a Java Swing dialog box titled "Propriétés de la connexion" (Connection Properties) with a close button (X) in the top right corner. The dialog is for a "Sylob Datasource", indicated by a database icon and the title. It contains several input fields and buttons:

- Nom**: A text field containing "monXMLFactureImmediateVente".
- Fichier XML**: A text field containing "C:\\FactureImmediateVente.XML" with a "Parcourir..." (Browse...) button to its right.
- Sélectionner l'e...**: A text field containing "root".
- Serveur**: An empty text field.
- Utilisateur**: An empty text field.
- Mot de passe**: An empty text field.
- Société**: An empty text field.
- Locale**: A dropdown menu showing "Par défaut" (Default) with a "Select..." button to its right.

At the bottom right of the dialog, there are three buttons: "Test", "Enregistrer" (Save), and "Annuler" (Cancel).

4. Fermer et relancer IReport car c'est au lancement d'IReport que ce dernier charge ces paramètres.
5. Cliquer sur « Ouvrir », sélectionner le fichier « factureImmediate.jrxml » qui se trouve dans un sous-répertoire de « "C:/factureImmediate" ». Le fichier avec l'extension jrxml est la maquette en cours de modification. Le fichier avec l'extension jasper est la maquette compilé qui sera exécutée. Ce fichier factureImmediate.jrxml est le support principal de la publication des factures immédiates. Voici l'écran obtenu.



6. Effectuer les modifications souhaitées.
7. Enregistrer votre rapport avec le nom suivant « factureImmediateModify.jrxml ». Ce nouveau nom permettra le chargement sur le serveur. Il est impossible de charger un état avec le même nom qu'un état réalisé par Sylob.
8. Compiler votre rapport en cliquant sur l'icône .
9. Lancer le rapport en cliquant sur l'icône . Une fenêtre s'ouvre et la facture apparaît avec les modifications effectuées.

3. Chargement d'un état

1. Ouvrir « C:/factureImmediate »,
2. Sélectionner l'ensemble des fichiers et dossiers. Ils contiennent tous les états « .jasper » de notre publication. Les fichiers qui commencent par « **specif-etat** » regroupent l'ensemble des clés d'internationalisation utiles pour notre publication. Si une nouvelle clé doit être ajoutée, la positionner en fin de fichier.
3. Créer un fichier Zip, pour cela cliquer sur « Add archive to », donner le nom du nouveau rapport principal « factureImmediateModify.zip ». **Le nom du fichier Zip doit impérativement être le même que celui du rapport principal.** Ce nom doit absolument être **unique pour l'ensemble des sociétés** sinon le système écrasera l'existant. Les états conçus par Sylob ne peuvent pas être écrasés.
4. Ouvrir l'activité « Gérer les paramètres de publications ».
5. Modifier l'état pour l'activité « Publier Facture Client Immédiate ».
6. Cliquer sur le bouton « Ajouter un groupe ».
7. Cliquer sur le bouton « Ajouter une feuille de style ». L'écran ci-dessous apparaît.

Gérer les paramètres des publications

Générer le code sql

Ajouter une feuille de style

Activité	Publier une facture client immédiate
Libellé du groupe :	
Nom de la feuille de style :	
Expression de base pour iReport :	/root *
Fichier (*.zip) :	<input type="text"/> Parcourir...

Liste des langues

Langues affectées

FRANCAIS (fr)

<< Affecter les langues

Retirer les langues >>

Langues non affectées

ALLEMAND (de) ▲
 ANGLAIS (en)
 ARABE (ar)
 CATALAN (ca)
 CHINOIS (zh)
 ESPAGNOL (sp)
 ITALIEN (IT)
 ITALIEN (it) ▼

Liste des paramétrages entité associés


Clé message	Libellé	Type de données	Action
Aucun élément trouvé.			

Valider

Annuler

- Cliquer sur « Parcourir » et aller chercher le fichier « factureImmediateModify.zip » nouvellement créé qui se trouve dans « C:/ factureImmediate ».
- Sélectionner les langues utilisateurs qui serviront de sélection lorsque l'utilisateur effectuera une publication. Si l'utilisateur a comme langue le Français, le système choisira la publication dont la langue affectée est le Français.
- Cliquer sur « Valider ».

Gérer les paramètres des publications

 L'élément "factureImmediate_Modify.jasper" a été ajouté avec succès.





Modifier le paramétrage de la publication pour l'activité "Publier une facture client immédiate"

Ajouter un groupe de feuille de style pour l'activité "Publier une facture client immédiate"

Activité :	Publier une facture client immédiate
Libellé :	Spécifique *
Type d'état :	Par élément ▼ *

Liste des feuilles de style

Ajouter une feuille de style...

Nom de la feuille de style	Personnalisé	Expression de base pour iReport	Langues affectées	Action
factureImmediate_Modify.jasper	Oui	/root	FRANCAIS	   

[1 - 1 sur 1]

Valider

Annuler

- Saisir un libellé pour le groupe.
- Sélectionner « Par élément » pour le type d'état.
- Cliquer sur « Valider ».
- Sélectionner le nouveau groupe dans la liste des groupes de feuille de style.
- Sélectionner le type de sortie « PDF ».

Gérer les paramètres des publications

Modifier le paramétrage de la publication pour l'activité "Publier une facture client immédiate"

Format de sortie : PDF

Domaine : dm2 - Gérer les ventes

Liste des groupes de feuilles de style

Ajouter un groupe...

Libellé	Personnalisé	Action
<input type="radio"/> Par défaut	Non	
<input checked="" type="radio"/> Spécifique	Oui	

[1 - 2 sur 2]

Informations à intégrer dans la source de données XML

Pour un niveau d'arborescence de : quatre

Afficher l'arborescence

Facture immédiate client :

Choisissez un élément

Publier au format XML ...

Valider

Annuler

16. Cliquer sur « Valider ».

17. La publication des factures immédiates fonctionne alors avec l'état nouvellement chargé.

IV Norme XML

La norme XML sert essentiellement à stocker et transférer des données structurées en arborescence. IReport utilise la source de donnée XML qui associé à la maquette JASPER donne la représentation finale de la publication qui peut être affichée au format PDF.

Ci-dessous, un exemple de fichier XML utilisé pour IReport :

```
<root>
  <factureSurLivraisonVente>
    <numeroFacture>00000025</numeroFacture>
    <dateFacture>23/12/2008</dateFacture>
    <reference>Référence de la facture 000000025</reference>
    <client>
      <code>00000040</code>
      <raisonSociale>ETS LESCORT</raisonSociale>
      <adresse>
        <numero>155</numero>
        <voie>rue</voie>
        <libelle>Jean Jaurès</libelle>
        <ville>ALBI</ville>
      </adresse>
    </client>
    <listeLigneDocumentCommercial>
      <numero>1</numero>
      <article>
        <code>ART1</code>
        <designation>Vis ART1</designation>
        <quantite>8.00</quantite>
      </article>
    </listeLigneDocumentCommercial>
    <listeLigneDocumentCommercial>
      <numero>2</numero>
      <article>
        <code>ART2</code>
        <designation>Ecrou ART2</designation>
      </article>
    </listeLigneDocumentCommercial>
  </factureSurLivraisonVente>
</root>
```

```
<quantite>4568</quantite>
</article>
</listeLigneDocumentCommercial>
</factureSurLivraisonVente>
</root>
```

Nœuds :

La hiérarchie représentée dans l'exemple ci-dessus met en évidence les balises « <root> » et « <factureSurLivraisonVente> ». Ces balises sont définies en XML par le terme nœuds car la hiérarchie continue entre la balise d'ouverture et la balise de fermeture. On dit alors que ces balises ont des enfants. L'exemple XML ci-dessous démontre bien qu'il existe des balises comme « <dateFacture> » ou des nœuds tel « <client> » entre l'ouverture du nœud « <factureSurLivraisonVente> » et la fermeture de ce nœud « </factureSurLivraisonVente> ».

Nœuds texte :

Les balises qui n'ont pas d'enfant comme « <reference> » par exemple sont nommées en XML « nœud texte ». Ces nœuds texte contiennent une valeur soit de type texte, soit numérique, soit booléenne, etc.

Pour accéder à une des valeurs d'un nœud texte, il faut partir de la racine « root » et descendre dans l'arborescence. Les différents nœuds croisés indiquent le chemin d'accès à la donnée. Par exemple, pour récupérer la valeur de la « dateFacture », Il faut passer par le nœud « <root> », puis le nœud « <factureSurLivraisonVente> » et enfin par le nœud texte « <dateFacture> ». Il faut donc renseigner le chemin suivant où chaque nœud est séparé par le caractère « / ». « /root/factureSurLivraisonVente/dateFacture ».

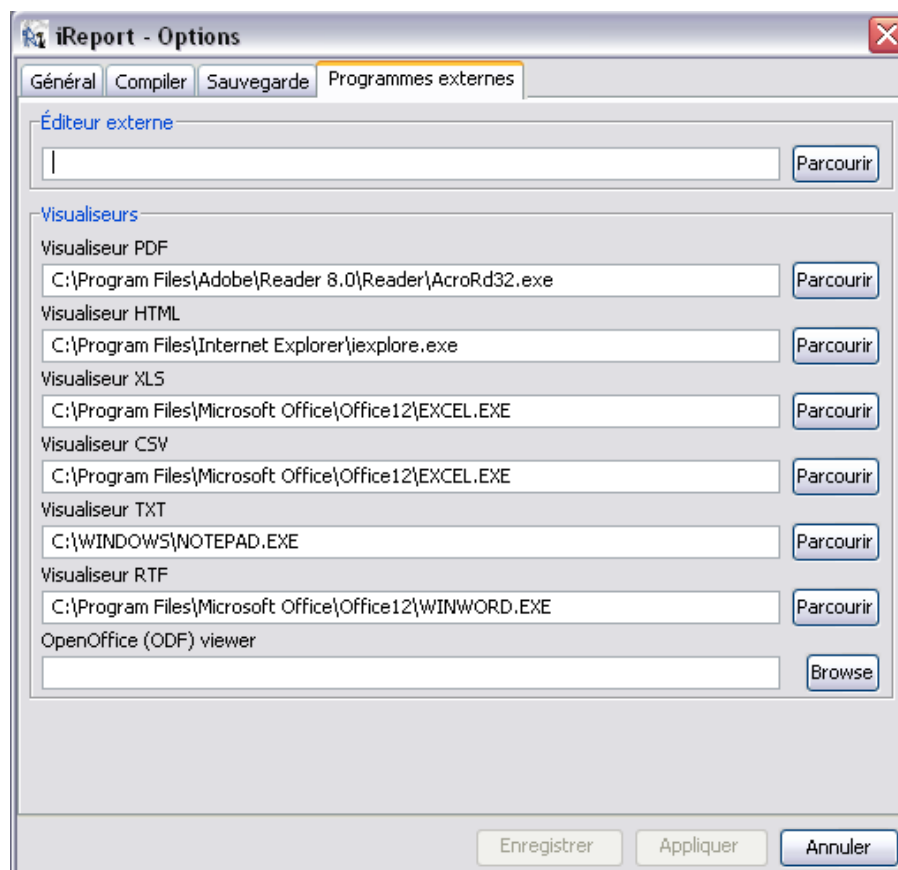
V Paramétrage d'IReport

1. Installation IReport : 2 paramétrages

1. Sélection du viewer

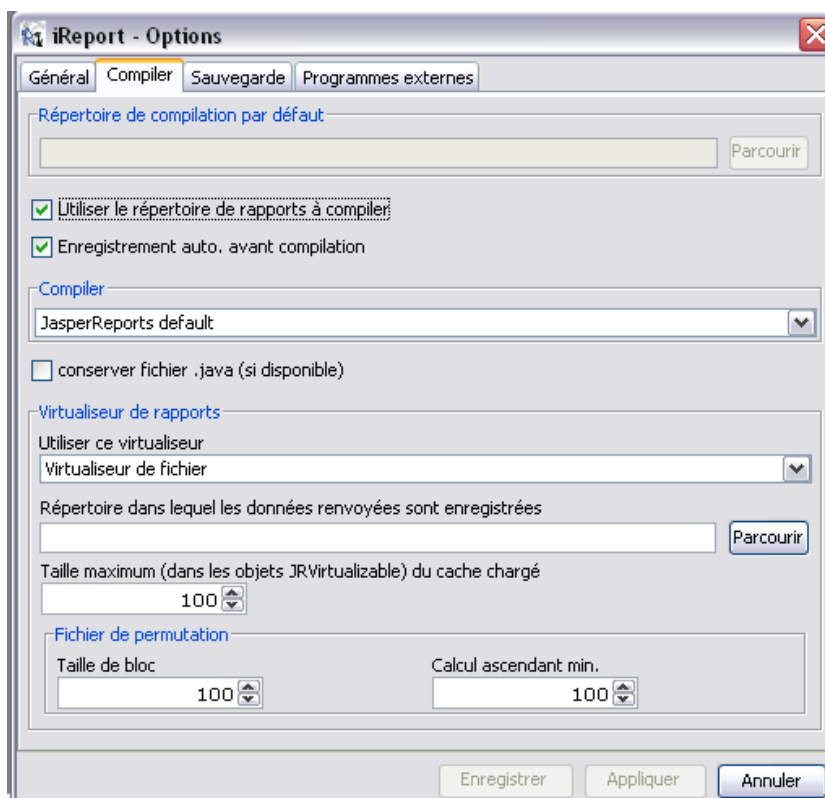
Il est possible de visualiser le résultat des modifications effectuées sur la maquette au format PDF, HTML, CSV, EXCEL, etc. IReport propose un outil de visualisation par défaut qui correspond parfaitement aux besoins des états Sylob.
Cliquer sur « Créer » et sélectionner « Aperçu JRViewer ».

Pour choisir un autre logiciel de visualisation de PDF comme ACROBAT READER par exemple, il faut le sélectionner dans les options d'IReport.
Cliquer sur « Options », « Paramètres », « Programmes externes » et spécifier le chemin jusqu'au logiciel dans la zone « visualiseur PDF » comme l'écran ci-dessous.
Cliquer ensuite sur « Créer » et sélectionner « Aperçu PDF ».



2. Compilation des maquettes

Les options de compilation des états sont disponibles dans l'onglet général de la fenêtre Options. Cocher les cases « Utiliser le répertoire de rapports à compiler » et « Enregistrement auto. Avant compilation » qui gère la compilation d'un état JASPER. Décocher la case « conserver fichier .java » qui est inutile pour les modifications de base.



3. Options facultatifs

Eviter la recherche de mise à jour : Cliquer sur « Options », « Configurer les plugins », « Check for IReport update », « Configurer » et décocher l'option « Rechercher les mises à jour lors du démarrage d'IReport ».

Eviter l'enregistrement des fichiers de sauvegarde : Cliquer sur « Options », « Paramètres », « Sauvegarde » et sélectionner l'option « Aucune sauvegarde ».

Eviter l'enregistrement des fichiers de programmation : Cliquer sur « Options », « Paramètres », « Compiler » et décocher l'option « Conserver le fichier java ».

2. Modification d'une maquette : 2 paramétrages

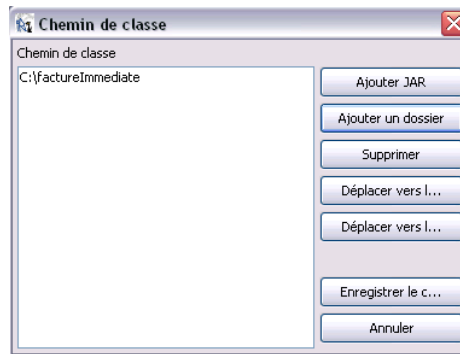
1. Sélection du chemin de classe « ClassPath »

Avant de modifier pour la première fois un nouvel état, il faut absolument renseigner le répertoire de base dans lequel sont stockés les états JASPER. C'est à partir de ce dossier qu'IReport fait le lien entre toutes les maquettes qui composent la publication.

Cliquer sur « Options », « Classpath ». Supprimer toutes les éventuelles références déjà présentes dans la liste. Cliquer sur « Ajouter un dossier ».

Sélectionner le dossier de base, c'est-à-dire le dossier créé après avoir dézippé le fichier zip contenant l'état. Ce dossier regroupe tous les états JASPER et les fichiers properties.

Cliquer sur « Enregistrer le Classpath ». Vérifier que seul le dossier figure dans la liste comme indiqué sur l'écran ci-dessous. Cette opération est à renouveler à chaque fois qu'un nouvel état doit être modifié.



2. Connexion à la source de données Sylob

IReport s'appuie sur différentes sources de données pour alimenter le rapport et afficher la maquette finale avec des données au format PDF.

Sylob a opté l'utilisation d'IReport avec la source de données XML.

L'écran ci-dessous représente les paramètres de connexion pour une source de données XML.

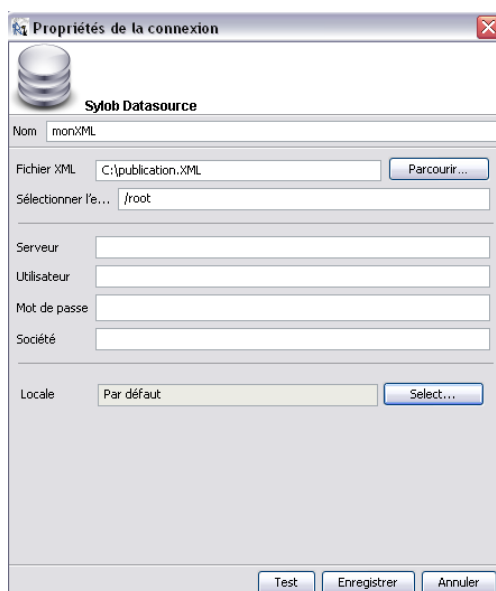
Pour établir une connexion avec un fichier XML, Cliquer sur « Data », « Connections », « Nouveau » et sélectionner « Sylob datasource ».

Le nom de la source de donnée n'a aucune incidence sur la connexion.

Le bouton « parcourir » permet de sélectionner le fichier XML contenant les données.

L'expression de base pour iReport, ici « /root » est spécifiée sur Sylob dans l'activité « Gérer les paramètres état ». L'expression de base varie selon les états. L'expression « /root » est utile pour la plupart des états proposés par Sylob. Cliquer sur « Enregistrer ».

Les informations du Serveur, Utilisateur, Mot de passe et Société sont utiles lorsque des requêtes sont présentes dans la maquette.



Fermer l'application et relancer IReport afin que la base de registre soit mise à jour. C'est au lancement d'IReport qu'il prend en compte ces deux paramètres.

VI Utilisation d'IReport

1. Structure du rapport

Un rapport se compose de plusieurs parties (band).

	background	
	title	
	pageHeader	
	columnHeader	
	detail	
	columnFooter	
	pageFooter	
	lastPageFooter	
	summary	

TITLE :

La bande TITLE est la première bande visible. Lors de l'édition, elle est créée une seule fois et peut être imprimée sur une page séparée. Se reporter au chapitre « Propriétés du rapport ».

PAGE HEADER :

La bande PAGE HEADER permet de définir un en-tête de page. La hauteur spécifiée lors de la phase de conception ne change généralement pas pendant le processus de création. L'en-tête de page apparaît sur toutes les pages imprimées dans la même position définie lors de la phase de conception. Les bandes TITLE et SUMMARY ne comprennent pas l'entête de page lorsqu'elles sont imprimées sur une page séparée.

COLUMN HEADER :

La bande COLUMN HEADER est imprimée au début de chaque DETAIL. Habituellement, les titres de colonnes des tableaux sont insérés dans cette bande.

DETAIL :

La bande DETAIL est renseignée à partir de chaque enregistrement qui est lu par la source de données. Cette section est répétée autant de fois qu'il y a d'enregistrements remontés par la source de données.

COLUMN FOOTER :

La zone COLUMN FOOTER apparaît à la fin du DETAIL. Habituellement le COLUMN FOOTER sert à afficher des totaux calculés à partir des données du tableau.

PAGE FOOTER :

La bande PAGE FOOTER apparaît au pied de chaque page sauf la dernière.

LAST PAGE FOOTER :

La bande LAST PAGE FOOTER permet de gérer le pied de page de la dernière page du rapport.

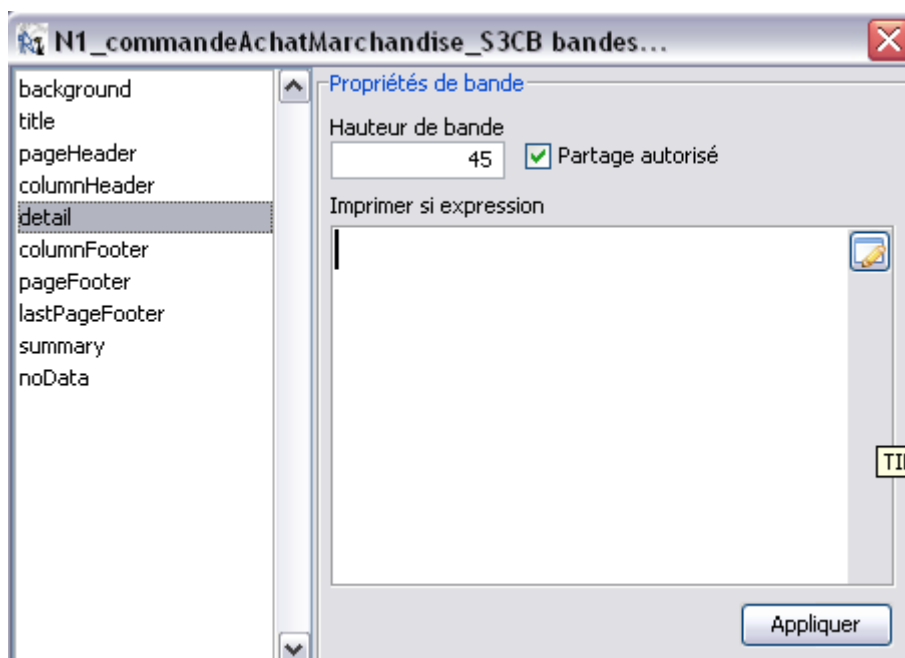
SUMMARY :

La zone SUMMARY permet d'insérer des champs concernant le calcul total, les moyennes, ce que l'utilisateur veut insérer à la fin du rapport. Cette zone apparaît sur la dernière page du rapport après les footer. Se reporter au chapitre « Propriétés du rapport ».

BACKGROUND :

Pour créer des filigranes ou des effets similaires comme un cadre autour de toute la page, il est possible de définir dans cette zone des éléments qui seront affichés sur chaque page en arrière plan.

Ci-dessous les propriétés propres à chaque zone.



Partage autorisé :

Les informations commencent à s'afficher sur la première page et continuent sur les pages suivantes si les informations ne peuvent pas être toutes affichées sur la première page. Si cette case n'est pas cochée, on évite de séparer les informations et elles sont affichées à partir de la seconde page.

Imprimer si expression :

Il est possible de conditionner l'affichage d'une zone.

Hauteur de bande :

Précise le nombre de pixels correspondant à la hauteur de la zone.

Largeur d'une bande:

Pour modifier la largeur d'une bande, il faut aller modifier la largeur du rapport. Cliquer sur Edit, Report properties et modifier la largeur. Se reporter au chapitre « Propriétés du rapport »

2. Propriétés du rapport

Comme tous les éléments de formulaire, un rapport a aussi des propriétés. Pour atteindre ces dernières, cliquer sur « Modifier », « Propriétés du rapport ». L'écran ci-dessous apparaît alors.



1. Taille de page

- Format du papier de d'impression (A4, A3, etc.)
- Largeur et la hauteur de bande
- Orientation de la page (Portrait ou paysage)

2. Marge de page

Une marge est l'espace qu'il y a entre le bord de la feuille et l'ensemble du document. La définition des marges doit être définie à la première modification d'une maquette afin d'optimiser au mieux la position des éléments sur le document.

3. Colonnes

Les colonnes proposent un affichage en liste sur plusieurs colonnes.

Exemple d'une liste d'article sur une colonne :

The screenshot shows the 'Propriétés du rapport' dialog box with the 'Colonnes' tab selected. The 'Nom du rapport' field contains 'listeArticle'. The 'Taille de page' section shows 'A4' as the predefined size, with dimensions of 20,990 cm width and 29,704 cm height, and 'Portrait' orientation. The 'Colonnes du rapport' section shows 1 column with a width of 19,438 cm and a spacing of 0,000 cm. The 'OK' and 'Annuler' buttons are visible on the right.

Propriétés du rapport	
Nom du rapport:	listeArticle
Taille de page	
Tailles prédéfinies:	A4
Largeur:	20,990 cm
Hauteur:	29,704 cm
Orientation:	Portrait
Marge de page Colonnes Classe de scriptlet Plus... i18n	
Colonnes du rapport	
Colonnes:	1
Largeur:	19,438 cm
Espacement:	0,000 cm

Structure du document

The screenshot shows the document structure in IReport. It features a horizontal ruler at the top with a scale from 0 to 20. Below the ruler, there is a table with three rows: 'CODE' with value '\$F(codsArticle)', 'DESIGNATION' with value '\$F(designationArticle)', and 'FAMILLE' with value '\$F(familleArticle)'. To the right of this table is a large rectangular area labeled 'detail'.

"CODE"	\$F(codsArticle)
"DESIGNATION"	\$F(designationArticle)
"FAMILLE"	\$F(familleArticle)

detail

Visualisation

CODE	C00009
DESIGNATION	ACIDE CITRIQUE PUR
FAMILLE	COSMETIQUE
CODE	10024
DESIGNATION	AU2GN (2618) DIAM 20
FAMILLE	ALUMINIUM
CODE	SNSBARD
DESIGNATION	BARDAGE D'HABILLAGE
FAMILLE	SOUS ENSEMBLE
CODE	SNSCORN
DESIGNATION	CORNIERE DE SUPPORT
FAMILLE	PIECE ACHETEE
CODE	C00005
DESIGNATION	CUTINA KD 16
FAMILLE	COSMETIQUE
CODE	C00010
DESIGNATION	DECANOX MTS 50
FAMILLE	COSMETIQUE
CODE	C00001
DESIGNATION	EAU DEMINERALISE
FAMILLE	COSMETIQUE
CODE	ELECON2
DESIGNATION	EMBASE SMA 3
FAMILLE	COMPOSANTS ELECTRONIQUES

Exemple d'une liste d'article sur deux colonnes :

Propriétés du rapport

Nom du rapport: listeArticle

OK

Annuler

Taille de page

Tailles prédéfinies: A4

Largeur: 20,990 cm

Hauteur: 29,704 cm

Orientation: Portrait

Marge de page

Colonnes

Classe de scriptlet

Plus...

i18n

Colonnes du rapport

Colonnes: 2

Largeur: 9,984 cm

Espacement: 0,000 cm

Structure du document

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
"CODE"	SF(codeArticle)																			
"DESIGNATION"	SF(designationArticle)																			
"FAMILLE"	SF(familleArticle)																			

Apparition de la limite de colonnes

Visualisation

CODE	C00009	CODE	C00008
DESIGNATION	ACIDE CITRIQUE PUR	DESIGNATION	PLANTSERVATIVE WSR
FAMILLE	COSMETIQUE	FAMILLE	COSMETIQUE
CODE	10024	CODE	SNSPLAT
DESIGNATION	AU2GN (2618) DIAM 20	DESIGNATION	PLAT OXYCOUPE
FAMILLE	ALUMINIUM	FAMILLE	MATIERE PREMIERE
CODE	SNSBARD	CODE	C00002
DESIGNATION	BARDAGE D'HABILLAGE	DESIGNATION	SODIUM BENZOATE
FAMILLE	SOUS ENSEMBLE	FAMILLE	COSMETIQUE
CODE	SNSCORN	CODE	SNSTUBE
DESIGNATION	CORNIERE DE SUPPORT	DESIGNATION	TUBE ALU
FAMILLE	PIECE ACHETEE	FAMILLE	MATIERE PREMIERE
CODE	C00005	CODE	C00007
DESIGNATION	CUTINA KD 16	DESIGNATION	VIATENZA PO 6
FAMILLE	COSMETIQUE	FAMILLE	COSMETIQUE

4. Plus

- L'option « Titre sur une nouvelle page » affiche la zone « Title » sur une première page. Le rapport comprenant les zones « PageHeader », « Detail » etc. sera affiché à partir d'une deuxième page.
- L'option « Résumé sur une nouvelle page » affiche la zone « SUMMARY » sur une nouvelle page en fin de rapport.

3. Element du rapport

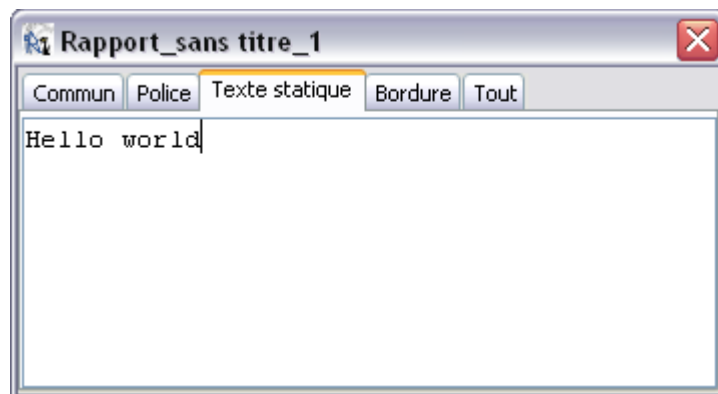
1. Flèche, Trait, Rectangle, Eclipse



Pour les 4 premiers éléments, Flèche, Trait, Rectangle, Eclipse, sélectionner l'élément à insérer et dessiner l'élément dans la zone de destination.

2. Texte statique

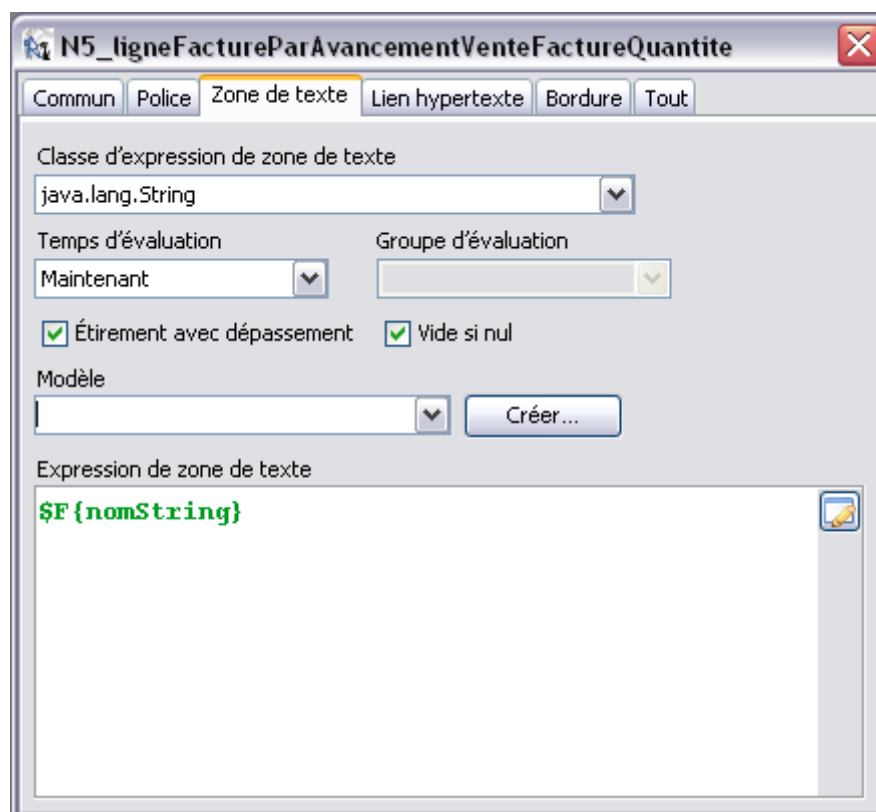
L'élément texte statique est utilisé pour afficher du texte non dynamique dans les rapports. L'onglet « texte statique » spécifie le texte à afficher. Cette section demande une chaîne de caractère sans guillemets.



3. Texte dynamique ou TextField

Le champ texte dynamique affiche la valeur d'une expression. Cet élément est principalement utilisé pour faire référence à un champ et donc en afficher la valeur.

L'onglet « Zone de texte » gère les caractéristiques de l'expression du TextField.



« Classe d'expression de zone de texte » :

Spécifie le type de retour de l'expression du champ. Les valeurs possibles sont nombreuses. Elles comprennent des types d'objet du langage de programmation Java. Au niveau de Sylob, nous utilisons la plupart du temps le type `java.lang.String`. Nous sommes amenés à utiliser d'autre type lorsque ce texte dynamique fait référence à un champ qui a un type différent de `String`. Dans ce cas là, le type du texte dynamique est identique à celui du champ.

« Etirement avec dépassement » :


Lorsque cette option est sélectionnée, elle permet au texte dynamique de s'adapter verticalement au contenu, si la zone d'affichage n'est pas suffisante pour contenir toutes les lignes du texte. Si la largeur du champ est trop petite alors le champ se redimensionnera automatiquement pour tout afficher. Les données s'afficheront donc sur plusieurs lignes.

« Vide si nul » :

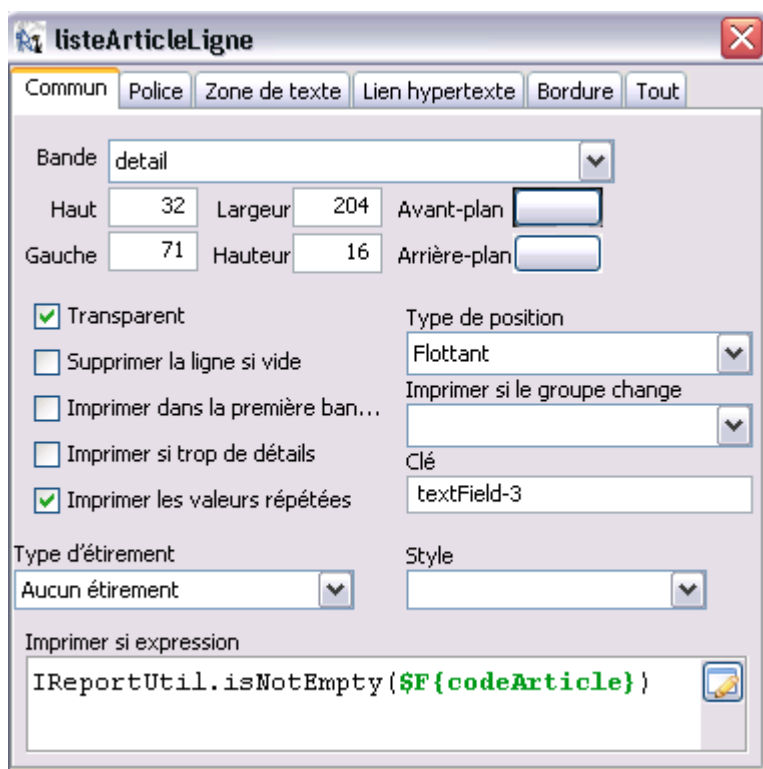
Afin d'éviter le texte "null" à l'affichage, cocher cette case si l'expression de champ peut retourner un objet null.

« Expression de zone de texte » :

C'est l'expression qui produit la valeur d'affichage, elle doit retourner un objet du même type que celui déclaré dans « Classe d'expression de zone de texte ». Dans l'exemple ci-dessus, le TextField affiche la valeur retournée par le champ « nomString ». Pour voir les différentes expressions possibles, se référer au chapitre JAVA.

Le texte dynamique peut aussi être utilisé comme un texte statique. Dans la zone expression, renseigner une chaîne de caractère entourée de guillemets. A la place de « `$F{nomString}` », il peut se trouver par exemple la chaîne « "Hello" ». Un éditeur d'expression  facilite la saisie des champs. Se reporter au chapitre « Editeur d'expression ».

L'onglet Commun gère les options du TextField



listeArticleLigne

Commun Police Zone de texte Lien hypertexte Bordure Tout

Bande: detail

Haut: 32 Largeur: 204 Avant-plan: []

Gauche: 71 Hauteur: 16 Arrière-plan: []

☒ Transparent

☐ Supprimer la ligne si vide

☐ Imprimer dans la première bande...

☐ Imprimer si trop de détails

☒ Imprimer les valeurs répétées

Type de position: Flottant

Imprimer si le groupe change: []

Clé: textField-3

Type d'étirement: Aucun étirement

Style: []

Imprimer si expression: `IReportUtil.isNotEmpty($F{codeArticle})`

« Bande » :

Précise dans quelle zone se situe l'élément. Les propriétés haut et gauche déterminent l'emplacement de l'élément dans la zone et les propriétés largeur et hauteur définissent les caractéristiques de l'élément.

« Transparent » :

Si cette option est sélectionnée, elle rend l'élément transparent. La transparence concerne le fond du TextField, pas le texte qu'il affiche.

« Supprimer la ligne si vide » :

Cette option supprime l'espace vertical occupé par un objet si la valeur de l'expression n'a aucune valeur : cas où la valeur est nulle.

« Imprimer dans la première bande » :

Cette option garantit que l'élément est affiché dans la page suivante ou colonne si le groupe déborde de la page. Ce type de garantie est utile lorsque l'impression répétée des valeurs d'attribut n'est pas actif.

« Imprimer si trop de détails » :

Cette option permet d'afficher l'élément dans la page suivante ou de la colonne quand la bande n'est pas suffisante pour tout afficher dans la page actuelle. Le champ commence à être affiché sur la première page puis sur la seconde page.

« Imprimer les valeurs répétées » :

Cette option détermine si l'élément doit s'afficher lorsque sa valeur est égale à celle de l'enregistrement précédent.

« Type de position » :

Cette option détermine la façon dont la coordonnée « haut » doit être considérée à l'affichage du rapport. Les trois valeurs possibles sont les suivantes :

- Placer relatif en haut : C'est le type de position prédéfinie, les valeurs des coordonnées ne changeront jamais.
- Flottant : L'élément est progressivement poussé vers le bas ou vers le haut par le comportement des éléments précédents. L'élément sera progressivement poussé vers le bas si les champs positionnés au dessus ont des hauteurs qui ont évolué. De même, il sera poussé vers le haut si les champs au dessus sont supprimés du rapport lors de l'affichage.
- Placer relatif en bas : La distance de l'élément à partir du bas de la bande reste constante. Cette option est habituellement utilisée pour les lignes qui séparent les enregistrements

« Type d'étirement » :

Cet attribut définit la façon de varier la hauteur de l'élément lors de l'affichage. Les trois valeurs possibles sont les suivantes :

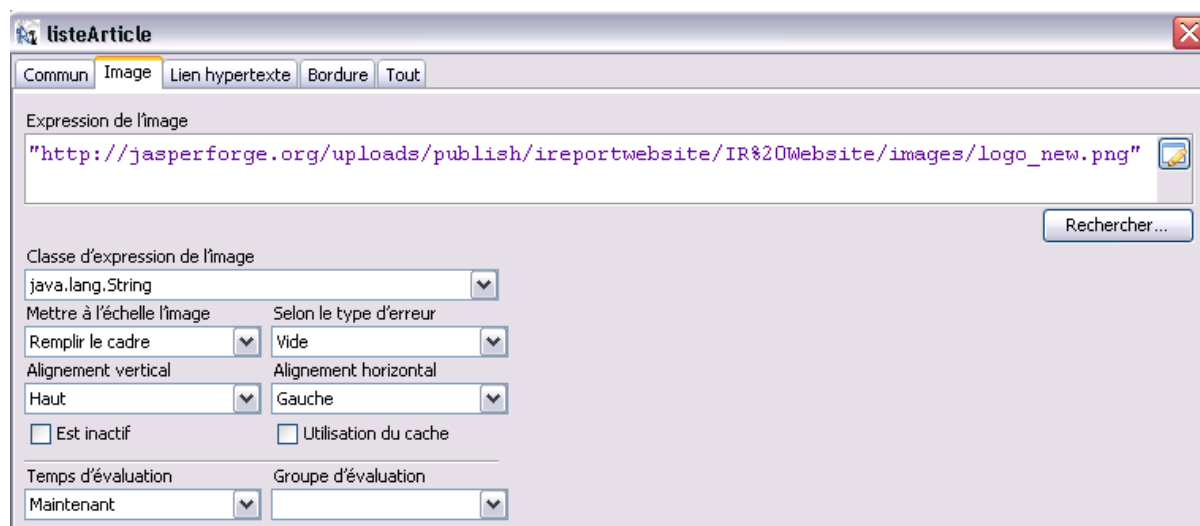
- Aucun étirement : la hauteur du champ est fixe
- Relatif à la hauteur de bande : La hauteur de l'élément est augmenté proportionnellement par rapport à la taille croissante de la bande, ce qui est utile pour les lignes verticales qui simulent des bordures de tableau.
- Relatif à l'objet le plus grand : L'élément modifie sa hauteur en fonction de la taille du champ le plus grand se trouvant sur la même ligne. Utile pour l'affichage des éléments en tableau.

« Imprimer si expression » :

Contient une expression qui doit retourner un objet Java de type Boolean. Si l'expression retourne vrai, alors l'élément est affiché. Une valeur nulle est considérée comme une valeur vraie. L'exemple ci-dessus montre que l'élément sera affiché si le champ `referenceArticle` est différent de null.

4. Image

Afficher une image sur une maquette est possible grâce à l'objet Image. 



listeArticle

Commun Image Lien hypertexte Bordure Tout

Expression de l'image

`"http://jasperforge.org/uploads/publish/ireportwebsite/IR%20Website/images/logo_new.png"`

Rechercher...

Classe d'expression de l'image

java.lang.String

Mettre à l'échelle l'image

Remplir le cadre

Alignement vertical

Haut

Est inactif

Temps d'évaluation

Maintenant

Selon le type d'erreur

Vide

Alignement horizontal

Gauche

Utilisation du cache

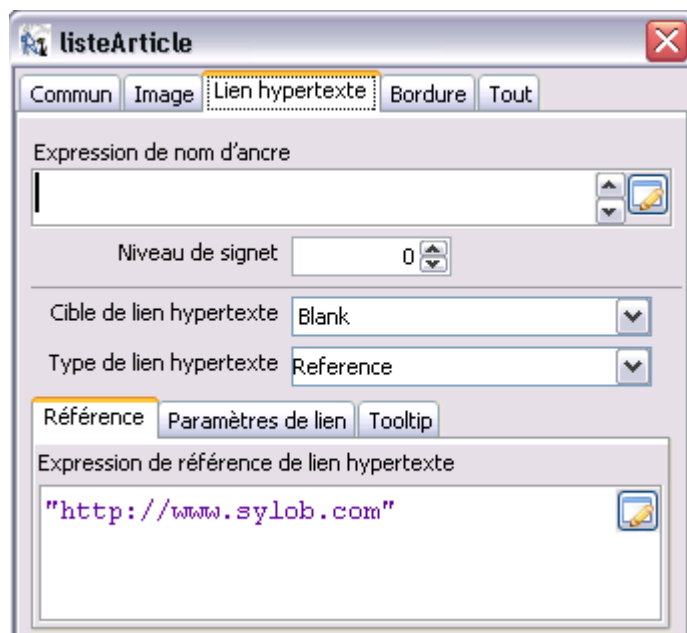
Groupe d'évaluation

L'expression de l'image doit contenir le lien vers une image. Cette image peut être stockée sur un serveur ou issue d'Internet.

La classe d'expression de l'image est de type « java.lang.String » pour des images pointées par un chemin.

Les options d'affichage « Mettre à l'échelle l'image », « Selon le type d'erreur », « Alignement vertical », « Alignement horizontal » et « Utilisation du cache » peuvent être spécifiées selon les exigences.

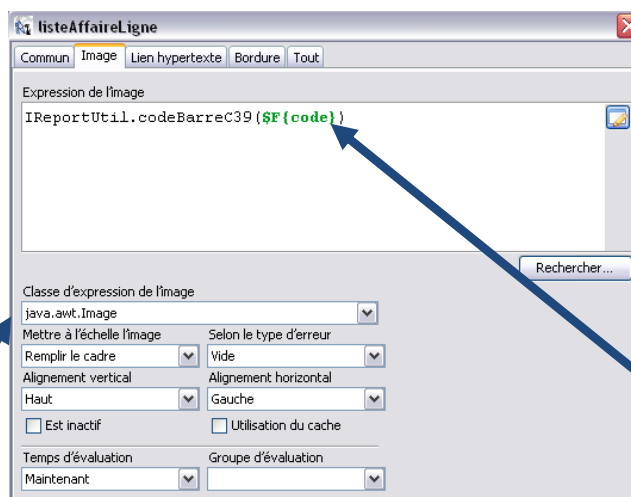
Un lien hypertexte peut être défini vers une adresse e-mail ou un site web comme le présente l'écran ci-dessous.



5. Code Barre

IReport gère les types de code barre C39 et EAN128.

L'affichage d'un code barre s'effectue grâce à l'objet image.



La classe d'expression de l'image doit être définie comme une « java.awt.Image ».

Les méthodes « IReportUtil.codeBarreC39 » et « IReportUtil.codeBarreEAN128 » permettent de transformer la valeur du champ en une image de type code barre.

```
IReportUtil.codeBarreC39($F{valeurCodeBarreAAfficher})
IReportUtil.codeBarreC39($F{valeurCodeBarreAAfficher}, 90)
IReportUtil.codeBarreEAN128($F{valeurCodeBarreAAfficher})
IReportUtil.codeBarreEAN128($F{valeurCodeBarreAAfficher}, 270)
```

Le second paramètre optionnel correspond au degré d'orientation du code barre. Les valeurs 0, 90, 180 et 270 sont possibles.

6. Subreport (sous-etat)

L'élément Subreport a pour fonction de référencer un autre rapport, appelé sous-rapport ou sous-état. Lors de la publication, Cet élément affiche le sous-rapport qui a été créé à partir d'un fichier JASPER et alimenté par une source de données. Ce lien entre rapport est spécifié dans les propriétés du sous-état.

La source de données provient soit du fichier XML ou soit d'une requête HQL.

Source de données issue du fichier XML : Fonction « subDatasource »

The screenshot shows the configuration window for 'N5_avoirVente_AEM'. The 'Rapport secondaire' tab is selected. The 'Expression de connexion/source de données' dropdown is set to 'Utiliser l'expression de source de données'. The text area contains the expression: `IReportUtil.subDataSource ($P{REPORT_DATA_SOURCE}, "societe")`. The 'Utilisation du cache' checkbox is checked.

Dans l'onglet « Rapport secondaire », renseigner la chaîne comme ci-dessus dans la partie « Expression de connexion de données ». La plupart des sous-états de Sylob utilisent la même expression de connexion de données. Il n'y a que la chaîne en violet qui varie selon le sous-état. Cette chaîne violette représente un nœud XML, voir chapitre sur la norme XML.

Source de données issue d'une requête HQL : Fonction « requeteDatasource »

The screenshot shows the configuration window for 'N5_tigneOffreDePrix'. The 'Rapport secondaire' tab is selected. The 'Expression de connexion/source de données' dropdown is set to 'Utiliser l'expression de source de données'. The text area contains the expression: `IReportUtil.requeteDataSource ($P{REPORT_DATA_SOURCE}, "select lot from LotEntite lot where article.lot.id='"+$F{idArticle}+"'")`. The 'Utilisation du cache' checkbox is checked.

Dans l'onglet « Rapport secondaire », renseigner la chaîne comme ci-dessus dans la partie « Expression de connexion de données ». La requête passée en paramètre est de type HQL. Utiliser l'activité « Gérer les requêtes » dans Sylob pour la construction de la requête. Dans le sous-rapport, les données sont accessibles à partir du nœud correspondant à l'objet principal spécifié dans la requête. Dans l'exemple, le nœud est « LotEntite ».

L'utilisation des requêtes est présente dans les maquettes des lignes de documents commerciaux proposées par Sylob pour récupérer les conditions clients et les conditions fournisseurs.

Si des requêtes doivent être utilisées dans les maquettes, les paramètres de connexion doivent être définis afin d'avoir les informations correspondantes lors de la visualisation de la maquette. Se reporter au chapitre « Connexion à la source de données Sylob ».

L'expression de rapport secondaire désigne le rapport ciblé. Le chemin commence à partir du répertoire spécifié dans le « chemin de classe ou Classpath ».

Il est possible de passer des paramètres à un sous-état. Se reporter au chapitre « paramètres » des éléments du formulaire.

N5_ligneFactureParAvancementVenteFactureQuantite

Commun Rapport secondaire Rapport secondaire (autre) Tout

Classe d'expression de rapport secondaire
java.lang.String

Expression de rapport secondaire
"dml/commun/PreportDuTotalEnHautDePage.jasper"

Paramètres de rapport secondaire Valeurs de retour de rapport secondaire

Paramètre	Expression
REPORT_RESOURCE_BUNDLE	\${REPORT_RESOURCE_BUNDLE}
DEVISE_DECIMALE	new Integer(2)
DEVISE_CODE	\${DEVISE_CODE}
SOMME_TOTAL_HT_NET	\${SommeTotalHTNet}

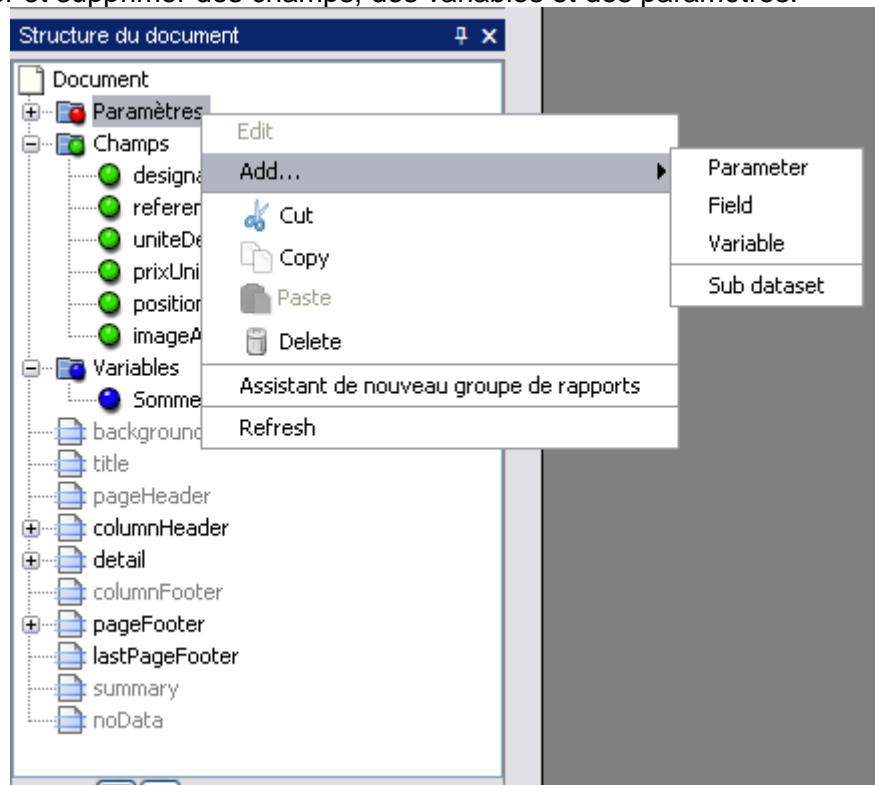
Ajouter Modifier Supprimer Copier à partir du fichier principal

4. Champ, paramètre et variable

Dans un rapport, il existe trois groupes d'objets qui peuvent stocker des valeurs: les champs, les paramètres et les variables. Ces objets sont utilisés dans les expressions.

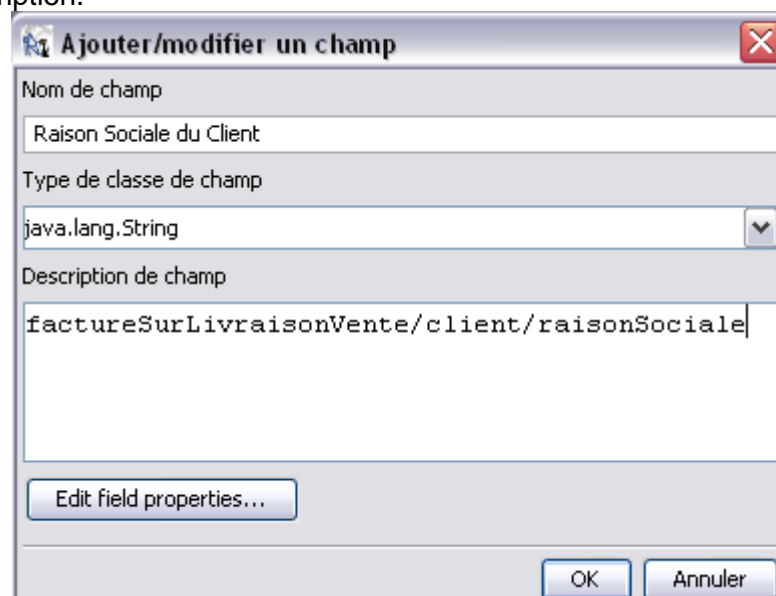
Les champs, les paramètres et les variables doivent être déclarés et sont référencés par un type (String, Boolean, Integer,...) dans le rapport afin d'être utilisés.

Dans le menu Affichage, se trouvent la liste des champs, variables, et paramètres utilisés par le rapport. La zone structure du document propose aussi la même liste d'éléments. Il est possible de déclarer, modifier et supprimer des champs, des variables et des paramètres.



1. Champ

Un rapport est créé à partir d'une source de données. Les champs sont identifiés par un nom, un type et une description.



Pour se référer à un champ, il faut utiliser le nom du champ. La syntaxe `$F{nomChamp}` fait référence au champ qui porte le nom « nomChamp ». Dans l'exemple ci-dessus, le nom du champ est « Raison Sociale du Client » et dans le rapport, les éléments utilisent la chaîne `$F{Raison Sociale du Client}` pour récupérer la valeur retournée par ce champ.

Pour accéder aux informations, la description du champ définit le chemin où IReport ira récupérer la valeur dans la source de donnée (fichier XML). Pour connaître le chemin exact, il faut s'appuyer sur le fichier XML pour accéder aux données.

Prenons un exemple avec le fichier XML défini dans la partie norme XML :

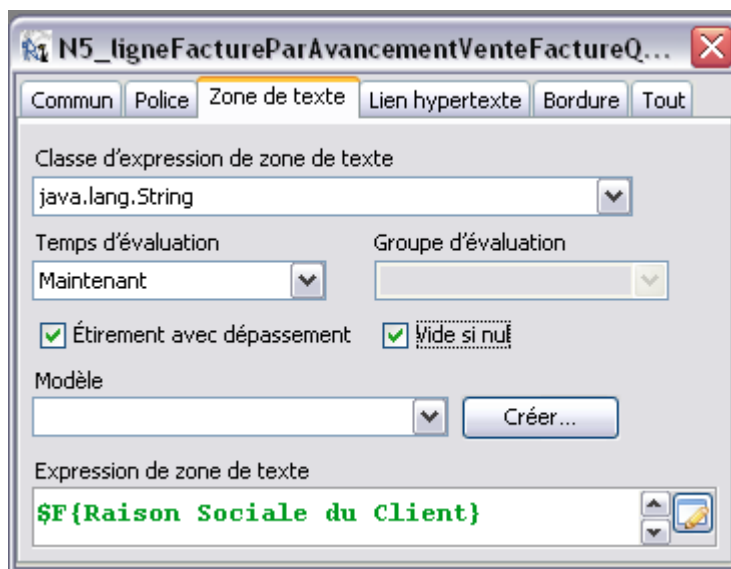
Ci-dessus, l'écran qui permet d'afficher la valeur « ETS LESCORT » qui est la raison sociale du client de la facture publiée.

Le chemin entier est `/root/factureSurLivraisonVente/client/raisonSociale`. L'expression « `/root` » est définie dans la connexion à la source de données l'expression.

Nous nous trouvons donc au nœud « root »

Pour accéder à la raison sociale du client à partir du nœud « root », le système doit passer par le nœud « factureSurLivraisonVente », puis le nœud « client » et enfin le nœud texte « raisonSociale ». Chaque nœud est représenté par la valeur de la balise suivi de la chaîne « `/` ». Donc la « Description du champ » doit contenir l'expression « `factureSurLivraisonVente/client/raisonSociale` ».

Pour afficher un champ dans un TextField, définir l'expression et le type du champ dans l'onglet « Zone de texte » du TextField.



Il est possible d'automatiser la création de l'expression et du type en faisant glisser un champ à partir de la liste des champs vers la bande désirée du rapport.

2. Paramètre

Les paramètres sont des valeurs généralement transmis au rapport qui crée l'affichage, d'un rapport vers un sous-rapport. Ils sont principalement utilisés pour fournir des données supplémentaires dans le contexte d'impression.

Comme pour les champs, le type des paramètres est déclaré au moment de la création.

L'exemple ci-dessous présente la création d'un paramètre `RAISON_SOCIALE_CLIENT` de type String.

Ajouter/modifier un paramètre

Nom de paramètre
RAISON_SOCIALE_CLIENT

Type de classe de paramètre
java.lang.String

☐ Utiliser comme invite

Expression de valeur par défaut
\$F{Raison Sociale du Client}

Description de paramètre

Edit parameter properties...

OK Annuler

L'expression de valeur par défaut fait référence à un champ. L'exemple montre que le paramètre « RAISON_SOCIALE_CLIENT » fait référence au champ `$F{Raison Sociale du Client}`. Pour se référer à un paramètre, il faut utiliser le nom du paramètre. La syntaxe `$P{nomParametre}` fait référence au paramètre qui porte le nom `nomParametre`.

iReport propose une liste de paramètre par défaut. Sylob utilise deux de ces paramètres :

REPORT_RESOURCE_BUNDLE est le paramètre utilisé par iReport pour affichés les libellés suivant la langue de l'utilisateur..

REPORT_LOCALE est utilisé pour définir les paramètres linguistiques utilisés pour remplir le rapport. Si aucune locale n'est précisée, le système prend la locale par défaut de l'utilisateur.

Passage de paramètre à un sous-état :

N5_ligneFactureParAvancementVenteFactureQuantite

Commun Rapport secondaire Rapport secondaire (autre) Tout

Classe d'expression de rapport secondaire
java.lang.String

Expression de rapport secondaire
"dm1/commun/PreportDuTotalEnHautDePage.jasper"

Paramètres de rapport secondaire Valeurs de retour de rapport secondaire

Paramètre	Expression
REPORT_RESOURCE_BUNDLE	\$P{REPORT_RESOURCE_BUNDLE}
DEVISE_DECIMALE	new Integer(2)
DEVISE_CODE	\$F{DEVISE_CODE}
SOMME_TOTAL_HT_NET	\$V{SommeTotalHTNet}

Ajouter Modifier Supprimer Copier à partir du fichier principal

Dans l'exemple ci-dessus, il y a 4 paramètres.

REPORT_RESOURCE_BUNDLE : Fait référence au paramètre « REPORT_RESOURCE_BUNDLE »

DEVISE_DECIMALE : Fait référence à une valeur numérique.

DEVISE_CODE : Fait référence à un champ DEVISE_CODE.

SOMME_TOTAL_HT_NET : Fait référence à une variable SommeTotalHTNet.

Pour récupérer la valeur d'un paramètre dans un sous-état, il faut créer un paramètre dans le sous-état portant le même nom que celui passé en paramètre depuis le rapport parent.

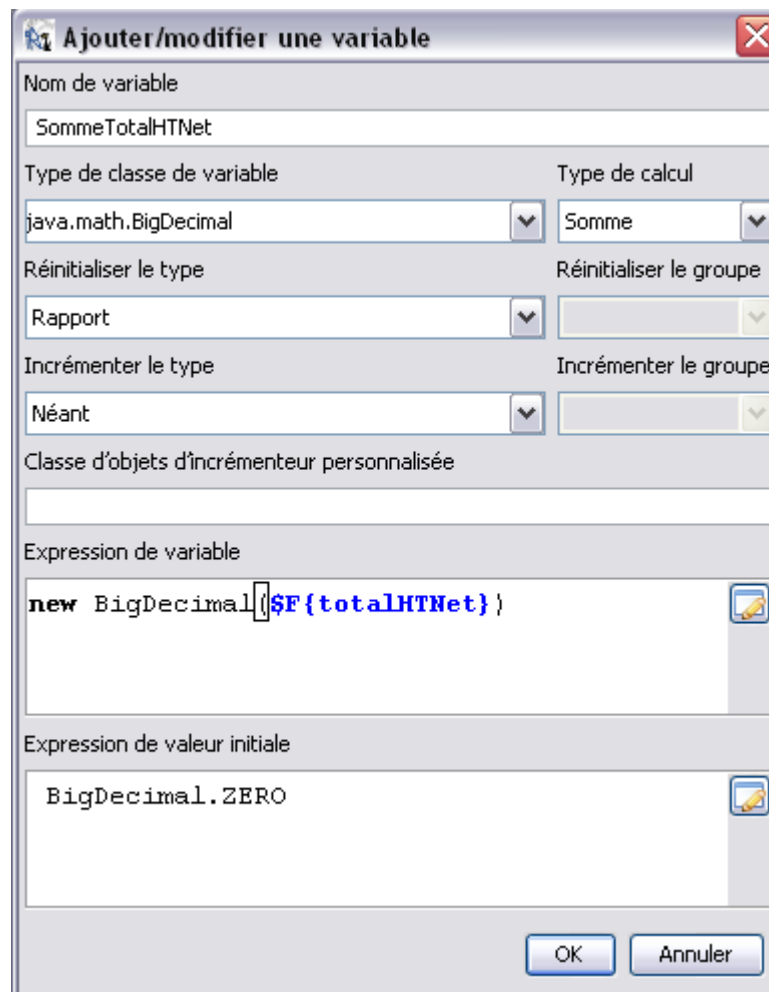
Dans le sous-rapport PreportDuTotalEnHautDePage, les paramètres REPORT_RESOURCE_BUNDLE, DEVISE_DECIMALE, DEVISE_CODE et SOMME_TOTAL_HT_NET doivent exister.

3. Variable

Les variables sont les objets utilisés pour stocker les résultats des calculs tels que les sous-totaux, totaux, etc. Pour se référer à une variable, il faut utiliser le nom de la variable. La syntaxe \$V{nomVar} fait référence à la variable qui porte le nom nomVar.

Comme pour les paramètres et les champs, les variables sont définies par un type. Ce dernier doit être en adéquation avec le type de calcul. S'il existe un type de calcul, alors le champ doit être obligatoirement de type numérique (Integer, Double...).

L'exemple ci-dessous présente la définition de la variable nommée SommeTotalHTNet qui fait la somme de tous les totalHTNet de tous les enregistrements. La valeur initiale commence à zéro.



Ajouter/modifier une variable

Nom de variable
SommeTotalHTNet

Type de classe de variable
java.math.BigDecimal

Type de calcul
Somme

Réinitialiser le type
Rapport

Réinitialiser le groupe
(empty)

Incrémenter le type
Néant

Incrémenter le groupe
(empty)

Classe d'objets d'incrémenteur personnalisée
(empty)

Expression de variable
new BigDecimal(**\$F{totalHTNet}**)

Expression de valeur initiale
BigDecimal.ZERO

OK Annuler

« Le type de calcul » :

Est utilisé pour stocker le résultat dans la variable. Le système effectue le calcul spécifié en changeant la valeur de la variable en fonction de chaque nouvel enregistrement qui est lu à partir de la source de données.

Nothing : Aucun calcul n'est effectué. Il est utilisé lorsque le type de calcul est intrinsèque à l'expression qui est spécifié et évalués pour chaque nouvel enregistrement.

Count : Ce dernier comptabilise combien de fois le résultat de l'expression est différente de null. Ce n'est pas la même chose que le type de calcul Sum, ce qui rend les sommes réelles basées sur la valeur numérique de l'expression.

Distinct Count : Ce dernier comptabilise le nombre de résultats d'expression différents, l'ordre d'évaluation de l'expression ne fait rien.

Sum : Cette somme s'ajoute à chaque itération la valeur d'expression de la valeur actuelle de la variable.

Average : calcule la moyenne arithmétique de toutes les expressions reçues en entrée.

Lowest : renvoie la valeur la plus faible expression reçue en entrée.

Highest : renvoie la valeur la plus élevée expression reçue en entrée.

StandardDeviation : renvoie l'écart type de toutes les expressions reçues en entrée.

Variance : renvoie la variance de toutes les expressions reçues en entrée

System : ne fait pas de calcul, et l'expression n'est pas évalué. Dans ce cas, le moteur de rapport ne retient que la dernière valeur fixée pour cette variable en mémoire. Ce paramètre peut être utilisé pour stocker le résultat des calculs effectués.

« Réinitialiser le type » :

Il est possible de réinitialiser la valeur de la variable. Ce critère spécifie quand une valeur de variable doit être remise à zéro, à la valeur initiale ou tout simplement à null. Le concept de réinitialisation de la variable est un élément fondamental quand on veut faire des calculs de groupe comme les sous-totaux ou les moyennes.

Aucun : L'expression de la valeur initiale est toujours ignorée.

Rapport : La variable est initialisée qu'une seule fois au début de la création de rapport en utilisant l'expression de valeur initiale.

Page : La variable est initialisée à nouveau dans chaque nouvelle page.

Colonne : La variable est initialisée à nouveau dans chaque nouvelle colonne (ou dans chaque page si le rapport est composé d'une seule colonne).

Groupe : La variable est initialisée à nouveau dans chaque nouveau groupe (le groupe spécifié dans le paramètre du groupe Reset).

« Incrémenter le type » :

La définition de l'évolution de la variable est spécifiée avec le type incrémentation. L'incrément du type spécifie quand une valeur de variable doit être évaluée. Les types d'incrément sont les mêmes que les types de calcul.

None : L'expression valeur initiale est toujours ignorée.

Rapport : La variable est incrémentée une seule fois au début de la création de rapport en utilisant l'expression de valeur initiale.

Page : La variable est incrémentée à nouveau dans chaque nouvelle page

Column : La variable est incrémentée à nouveau dans chaque nouvelle colonne (ou dans chaque page si le rapport est composé d'une seule colonne).

Group : La variable est incrémentée à nouveau dans chaque nouveau groupe (le groupe spécifié dans le paramètre du groupe Reset).

« Variable expression » :

A partir de la valeur définie dans ce champ, le type de calcul est appliqué à la variable à chaque itération.

Par exemple, pour une somme la valeur du champ sera additionnée à la variable.

« Valeur initiale expression »:

La variable est initialisée avec cette valeur.

5. Clé message

Sylob offre la possibilité d'effectuer des publications internationalisées. Par exemple on peut effectuer des publications de documents commerciaux dans la langue du tiers afin de l'envoyer à celui-ci. L'affichage de données statiques comme les entêtes de colonne par exemple doit pouvoir se faire en français, en anglais etc... Il existe un fichier par langue qui stocke toutes les clés message utilisées dans tous les états de Sylob.

Ce fichier est composé de plusieurs lignes, chacune respectant la convention clemessage = valeur affichée dans l'état.

Exemple :

Dans le fichier ireport_fr.properties (Fichier par défaut), se trouve la ligne

ligneCommandeVente.prixUnitaireHT = Prix unitaire

Dans le fichier ireport_en.properties (Fichier pour l'anglais) se trouve la ligne

ligneCommandeVente.prixUnitaireHT = Unit price

Dans les rapports, pour afficher la valeur de la clé message ligneCommandeVente.prixUnitaireHT, l'expression `$R{ligneCommandeVente.prixUnitaireHT}` doit être utilisée dans l'expression d'un texte dynamique. Lors de la publication en français, le système ira chercher la valeur « Prix unitaire » et lors de la publication en anglais, le système ira chercher la valeur « Unit price ».

Il est possible de référencer des clés avec des valeurs variables passées en argument. La fonction msg est alors utilisée pour construire dynamiquement des messages en utilisant des arguments. msg est composée de la clé ainsi que des arguments placés chronologiquement pour l'affichage de la valeur. La position de l'argument est exprimée en utilisant des nombres entre accolades.

Exemple :

Dans le fichier properties :

commandeLivraison = Commande {0} de la livraison {1}

Dans le rapport :

msg(\$R{ commandeLivraison }, \$F{numeroCommande}, \$F{numeroLivraison})

Le système va récupérer la valeur de la clé commandeLivraison et va remplacer le {0} par la valeur du champ numeroCommande et {1} par la valeur du champ numeroLivraison.

Si des clés doivent être créées lors d'une modification d'une maquette, elles doivent être ajoutées dans les fichiers commençant par « specif-etat ». Ces fichiers sont propres à chaque client. Attention, à chaque chargement d'une maquette dans Sylob, le système va écraser les fichiers « specif-etat.properties » qui sont stockés sur le serveur. Il faut donc être attentif et vérifier que les fichiers « specif-etat.properties » mis dans le fichier ZIP contiennent toutes les clés créées auparavant.

6. Option de groupe et de tri

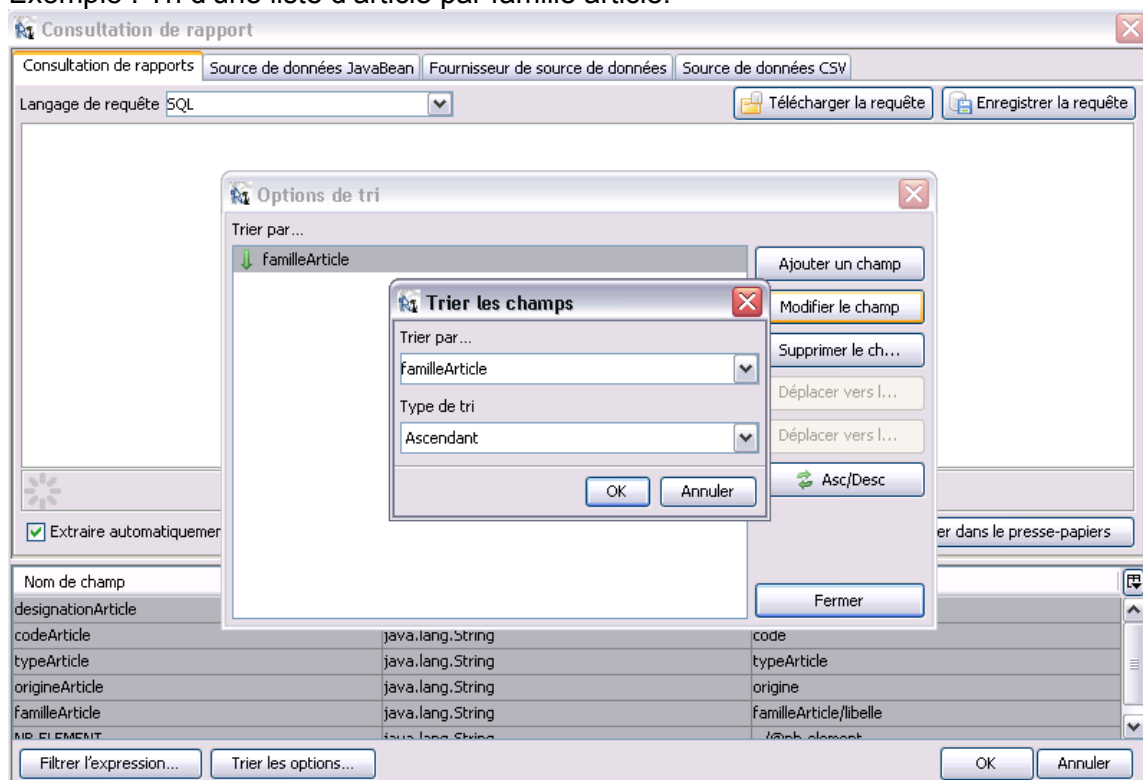
IReport est un outil de présentation des informations. Les données affichées en liste peuvent donc être triées ou regroupées selon un ou plusieurs critères.

1. Tri

La définition du tri doit obligatoirement être spécifiée dans le rapport qui contient les données à trier. Si les éléments à trier sont des articles, le tri doit être défini dans le rapport qui contient les zones de texte qui affichent les données dynamiques de l'article.

Pour définir un tri, cliquer sur « Modifier », « Consultation de rapports » et « Trier les options ».

Exemple : Tri d'une liste d'article par famille article.



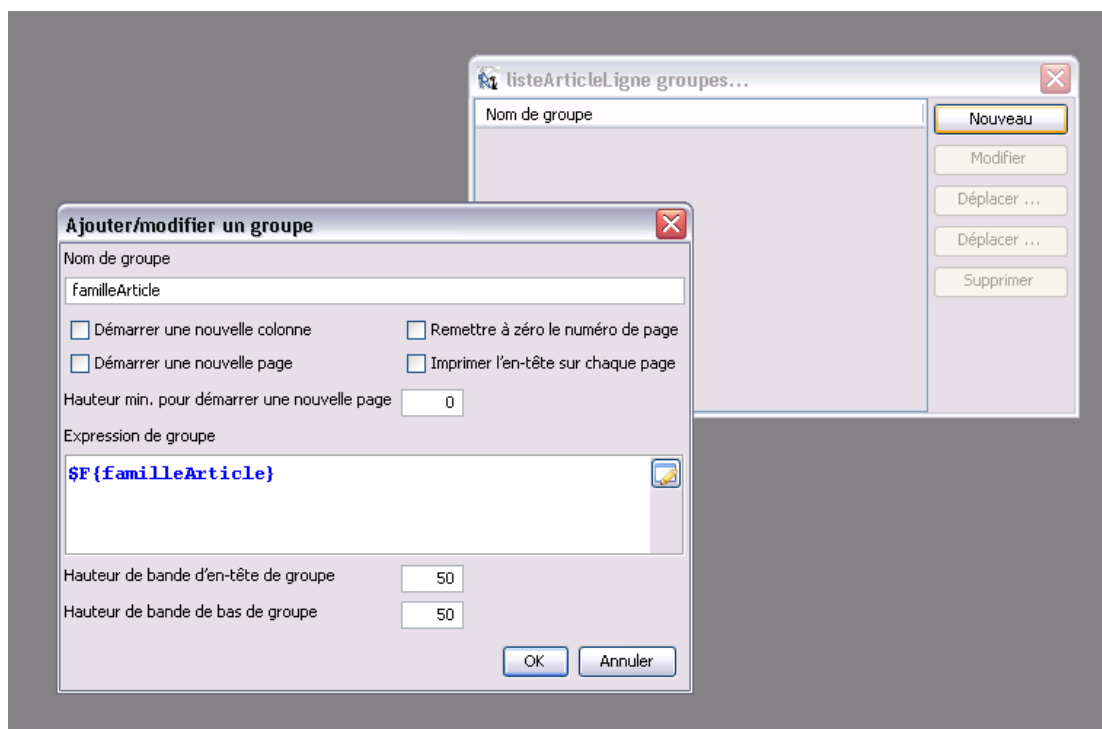
Lorsqu'un tri est créé dans un rapport, la source de données n'est plus un Sylob datasource. De ce fait, il n'est pas possible d'insérer dans ce rapport un lien vers un sous-rapport.

2. Groupe

Un groupe permet de créer autant de zone que de valeur attribuée au champ de groupe. Pour créer un groupe, il faut dans un premier temps définir un tri sur le champ sur lequel porte le groupe. Ensuite, cliquer sur « Afficher », « Groupes de rapports », « Nouveau ». Saisir ensuite un nom de groupe et le champ sur lequel porte le groupe.

Exemple : Groupe les articles par famille article

Après avoir fait un tri sur le champ « familleArticle », le groupe est créé sur le champ « familleArticle » comme le montre l'écran ci-dessous.



Chaque groupe possède une entête « familleArticleHeader » et un pied « familleArticleFooter » qui apparaissent à chaque fois qu'il y a un changement dans la valeur associée. Il est possible d'effectuer plusieurs groupes.

Dans la visualisation finale du groupe sur la famille article, j'ai donc mis le champ « familleArticle » en entête de groupe afin que la valeur de la famille de l'article n'apparaisse qu'une seule fois.

Structure du document

str(\$F{familleArticle})			
familleArticleHeader			
\$F{designationArticle}	\$F{codeArticle}	\$F{...}	\$F{...}
familleArticleFooter			

Visualisation

ARTICLES ACHETES

PEINTURE	PEINT	Matière	Acheté
----------	-------	---------	--------

COMPOSANTS ELECTRONIQUES

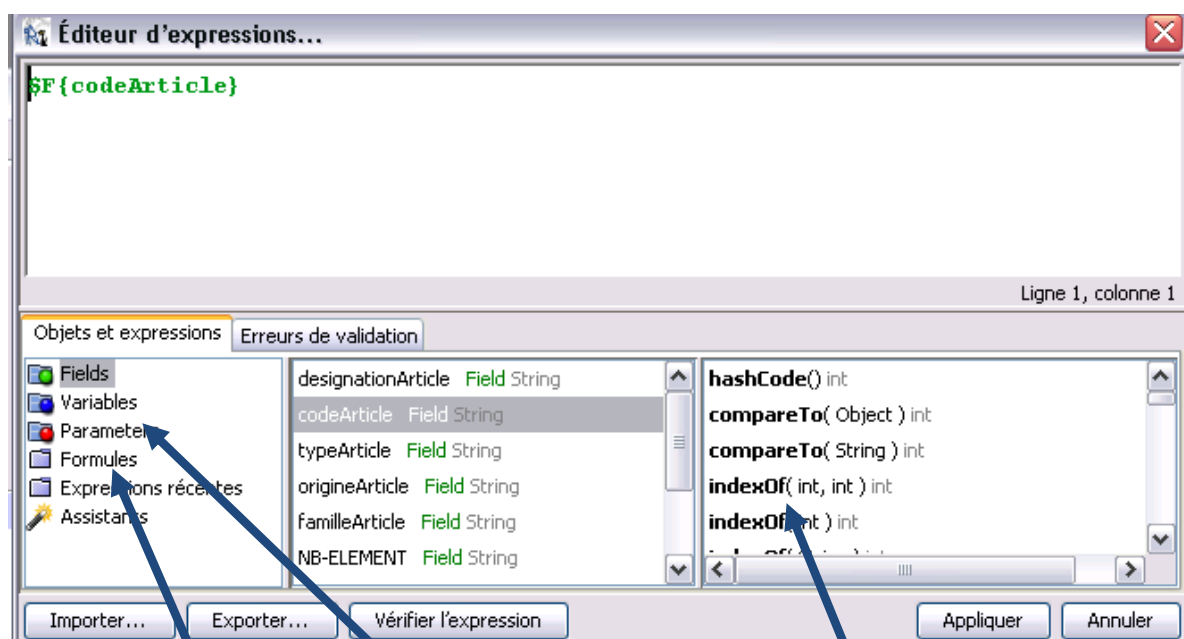
EMBASE SMA 3	ELECON2	Matière	Acheté
FIL DE SOUDURE	ELEFILSOUD	Matière	Acheté

COSMETIQUE

ACIDE CITRIQUE PUR	C00009	Matière	Acheté
CUTINA KD 16	C00005	Matière	Acheté
DECANOX MTS 50	C00010	Matière	Acheté
EAU DEMINERALISE	C00001	Matière	Acheté
GELIFIANT	C00003	Matière	Acheté
KELTROL	C00004	Matière	Acheté
MYTRIOL 318	C00006	Matière	Acheté
PARFUM N°4	C00011	Matière	Acheté
PLANTSERVATIVE WSR	C00008	Matière	Acheté
SODIUM BENZOATE	C00002	Matière	Acheté
VIATENZA PO 6	C00007	Matière	Acheté

Des variables calculant le total d'un montant par groupe est possible. Se reporter au chapitre « variable ».

7. Editeur d'expression



Liste des fonctions disponibles pour l'objet codeArticle

Liste des éléments pouvant être référencé dans la zone de texte

Liste des formules disponibles

Fieds, Variable, Parameters

La liste des champs, variables et paramètres définis dans la structure du document peuvent être ajoutés dans la zone de visualiation. De plus, une liste de fonctions est applicable à l'objet sélectionné.

Formules :

Toutes les expressions iReport peuvent être éditées en utilisant l'éditeur d'expression. Pour créer une expression, sélectionner la fonction « Formules ». Choisir alors une expression parmi celles disponibles. Dans la fenêtre qui apparaît, iReport propose l'ajout, la suppression et la modification des expressions. Cet assistant est utile pour comprendre certaines expressions utilisées dans les états Sylob.

VII Création d'une activité de publication

Sylob offre la possibilité de créer une activité dont l'**unique action sera d'effectuer une publication** à partir d'une liste d'éléments.

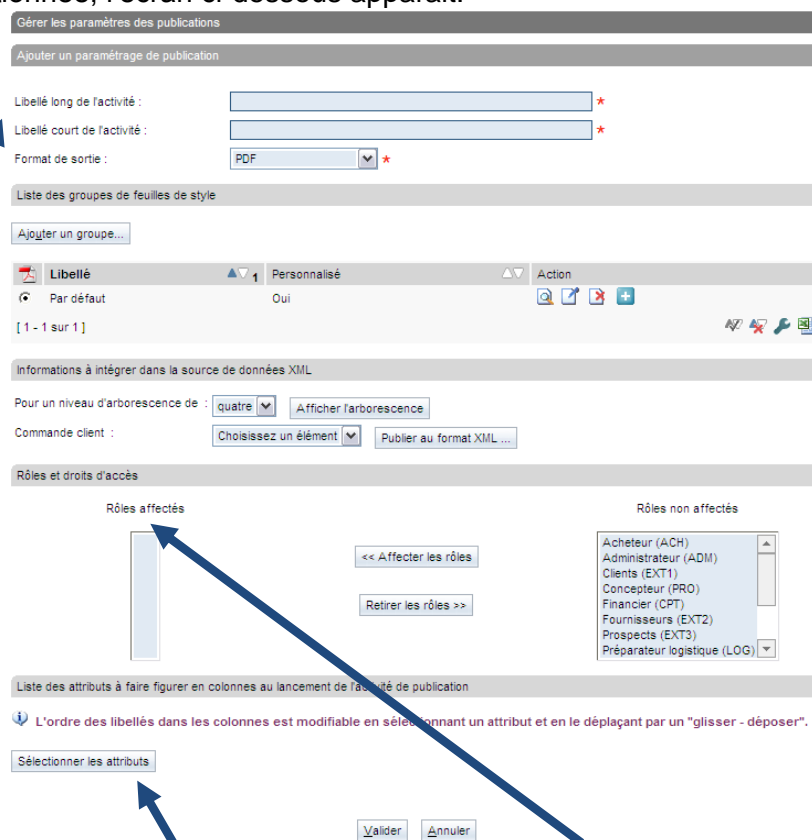
Cette nouvelle activité affichera une liste d'éléments, l'utilisateur sélectionnera alors les éléments qu'il désire publier et Sylob effectuera la publication.

La génération d'une activité de publication s'effectue par la création d'un nouveau paramètre état au niveau de l'activité « Gérer les paramètres de publication ».

Ci-dessous le scénario à suivre.

Dans l'activité « Gérer les paramètres de publication », cliquer sur le bouton « Ajouter ». Sélectionner l'entité principale de votre publication. L'entité principale est l'élément sur lequel porte la publication (Exemple : commande, ordre de fabrication, affaire, etc.)

Une fois l'entité sélectionnée, l'écran ci-dessous apparaît.



Nom de l'activité générée

Attributs de l'entité qui figureront dans la liste affichée par l'activité générée

Rôles d'accès à l'activité générée

Renseigner les paramètres ci-dessus et ceux détaillés au paragraphe « II – 3 ».

A la validation du paramètre de publication, Sylob crée un nouveau paramètre état et génère une nouvelle activité. Comme pour toutes les activités présentes dans Sylob, elle est accessible par les raccourcis ou par démarrer une activité et peut aussi être ajoutée à un menu personnalisé.

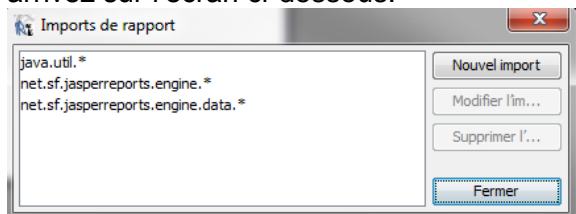
VIII Expression JAVA

Ci-dessous une liste des formules et expressions JAVA utilisées dans les états Sylob.

IReportUtil :

Sur les maquettes déjà modifiées, la librairie n'est peut-être pas encore déclarer.

Pour déclarer cette librairie, Cliquer sur « Modifier », « Directives d'importation des rapports ». Vous arrivez sur l'écran ci-dessous.

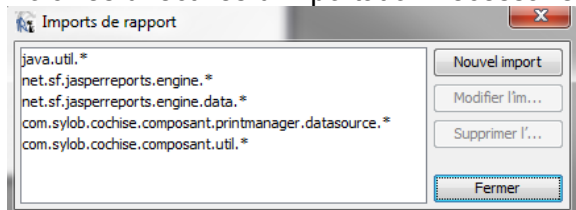


Cliquer sur Nouvel import et ajouter les lignes

com.sylob.cochise.composant.printmanager.datasource.*

com.sylob.cochise.composant.util.*

Voici les directives d'importation nécessaires pour utiliser la librairie IReportUtil.



Référence à des objets :

$\$F\{\text{nomChamp}\}$ -> référence au champ nommé nomChamp

$\$R\{\text{clemessage}\}$ -> référence à une clé dans le fichier properties

$\$P\{\text{nomParametre}\}$ -> référence au paramètre nommé nomParametre

$\$V\{\text{nomVariable}\}$ -> référence à la variable nommée nomVariable

Fonctions Sylob :

sylob:logoSociete() : affiche le logo de la société

sylob:commentaire() : affiche le commentaire de l'élément

sylob:document() : affiche l'image d'un élément. Cette image doit être créée en document attaché et doit porter comme nom le code de l'élément où le numéro de l'élément.

Champ null, non-null, renseigné ou non-renseigné :

Champ « nomChamp » est null

$\$F\{\text{nomChamp}\}==\text{null}$

Champ « nomChamp » différent de null

$\$F\{\text{nomChamp}\}!=\text{null}$

Champ « nomChamp renseigné (différent de null et différent de chaine vide)

$\text{IReportUtil.isEmpty}(\$F\{\text{nomChamp}\})$

Champ « nomChamp non-renseigné (Soit null ou chaine vide)

$\text{IReportUtil.isEmpty}(\$F\{\text{nomChamp}\})$

Concaténation de chaîne de caractère et des champs :

`$F{nomChamp1} + " et " + $F{nomChamp2}`

Le système va concaténer le champ1, la chaîne de caractère « et » et le champ2. La concaténation est possible pour tous les types de champ.

Fonctions sur les chaînes de caractères String :

Champ « nomString » est égal à « MaChaine »

`IReportUtil.equals($F{nomChamp}, "MaChaine")` -> retourne un boolean vrai-faux

Champ « nomString » est différent de « MaChaine »

`IReportUtil.notEquals($F{nomChamp}, "MaChaine")` -> boolean vrai-faux

Champ « nomChamp » se termine par « MaChaine »

`IReportUtil.endsWith($F{nomChamp}, "MaChaine")` -> boolean vrai-faux

Champ « nomChamp » commence par « MaChaine »

`IReportUtil.startsWith($F{nomChamp}, "MaChaine")` -> boolean vrai-faux

Retourne la longueur de la chaîne « nomChamp »

`IReportUtil.length($F{nomChamp})` -> int nb de caractère

Retourne la chaîne « nomChamp » coupée du caractère 2 au 6

`IReportUtil.substring($F{nomChamp}, 1, 6)` -> string coupe la chaîne du caractère 2 au 6

Champ « nomChamp » contient la chaîne « MaChaine »

`IReportUtil.contains($F{nomChamp}, "MaChaine")` -> boolean vrai-faux

Champ « nomChamp » ne contient pas la chaîne « MaChaine »

`IReportUtil.notContains($F{nomChamp}, "MaChaine")` -> boolean vrai-faux

Fonction sur les numériques de type Long et Integer :

Champ « nomInteger » est égal à 1

`IReportUtil.equals($F{ nomInteger }, 1)` -> retourne un boolean vrai-faux

Champ « nomInteger » est différent de 1

`IReportUtil.notEquals($F{ nomInteger }, 1)` -> retourne un boolean vrai-faux

Champ « nomInteger » est supérieur à 1

`IReportUtil.higher($F{ nomInteger }, 1)` -> retourne un boolean vrai-faux

Champ « nomInteger » est inférieur à 1

`IReportUtil.lower($F{ nomInteger }, 1)` -> retourne un boolean vrai-faux

Ajout de 5 au champ « nomInteger »

`IReportUtil.add($F{ nomInteger }, 5)`

Soustraction de 5 au champ « nomInteger »

`IReportUtil.subtract($F{ nomInteger }, 5)`

Multiplication du champ « nomInteger » par 5
`IReportUtil.multiply(${ nomInteger }, 5)`

Division du champ « nomInteger » par 5
`IReportUtil.divide(${ nomInteger }, 5)`

Fonctions utiles pour l'affichage :

Quantité :
`IReportUtil.formatNumber(${Quantite}, 2, ${REPORT_LOCALE})`

Montant :
`IReportUtil.formatNumber(${Montant}, ${nbDecimale}, ${REPORT_LOCALE})`

Prix :
`IReportUtil.formatNumber(${ Prix }, ${nbDecimale}, ${REPORT_LOCALE})`

La fonction `formatNumber` prend trois paramètres en entrée. Le champ, le nombre de décimales et la locale.

Le nombre de décimale peut être issu d'un champ déclaré « `${nbDecimale}` » ou d'une valeur statique « 2 ».

Le paramètre `REPORT_LOCALE` est défini au lancement de la publication et permet le formatage pour l'internationalisation.

Date :
Affiche et formate le champ « `maDate` »
`IReportUtil.formatDate(${maDate}, ${R{pattern.date8}})`

La fonction `formatDate` prend la date en premier paramètre et une clé en second paramètre. Cette clé est la structure d'affichage de la date.

Pour afficher la date du jour, il faut remplacer `${maDate}` par `new Date()`.

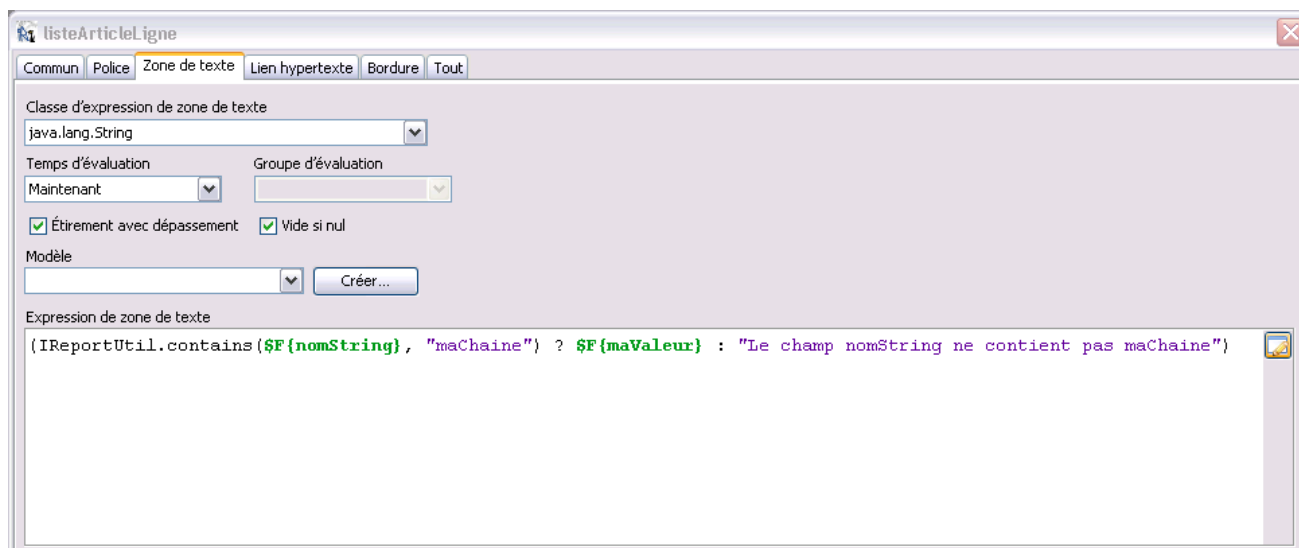
Structure Conditionnelle :

(condition ? expression1 : expression2)

Se lit de la manière suivante :

Si condition alors expression 1 sinon expression 2 FinSi

L'exemple ci-dessous détermine : si le champ « `nomString` » contient la chaîne « `maChaine` » alors le `TextField` affiche le champ « `maValeur` » sinon le `TextField` affichera la chaîne de caractère « Le champ `nomString` ne contient pas `maChaine` ».

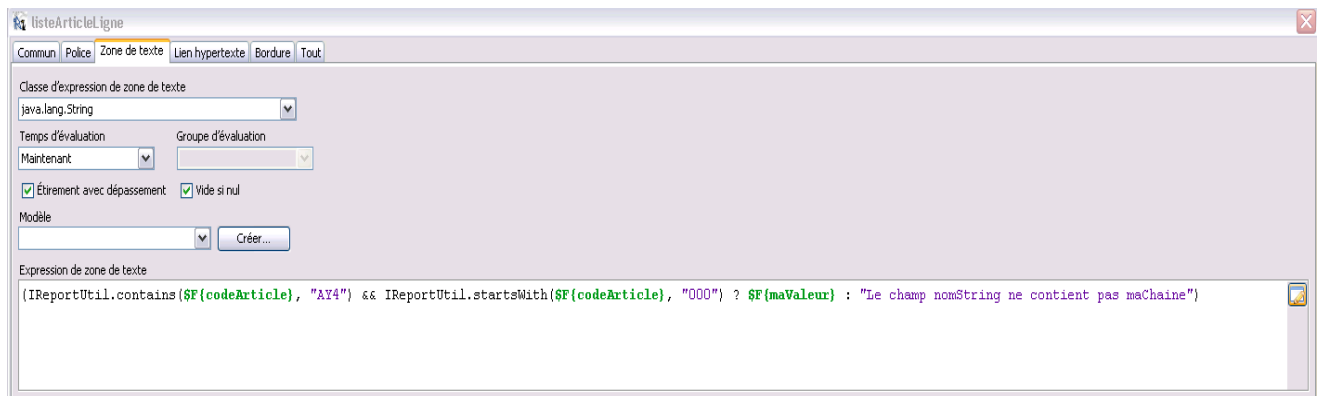


Une condition peut porter sur plusieurs critères. Le connecteur logique entre les deux conditions est soit « ET » ou alors « OU ».

Le connecteur ET se retranscrit pas &&

Le connecteur OU se retranscrit par ||

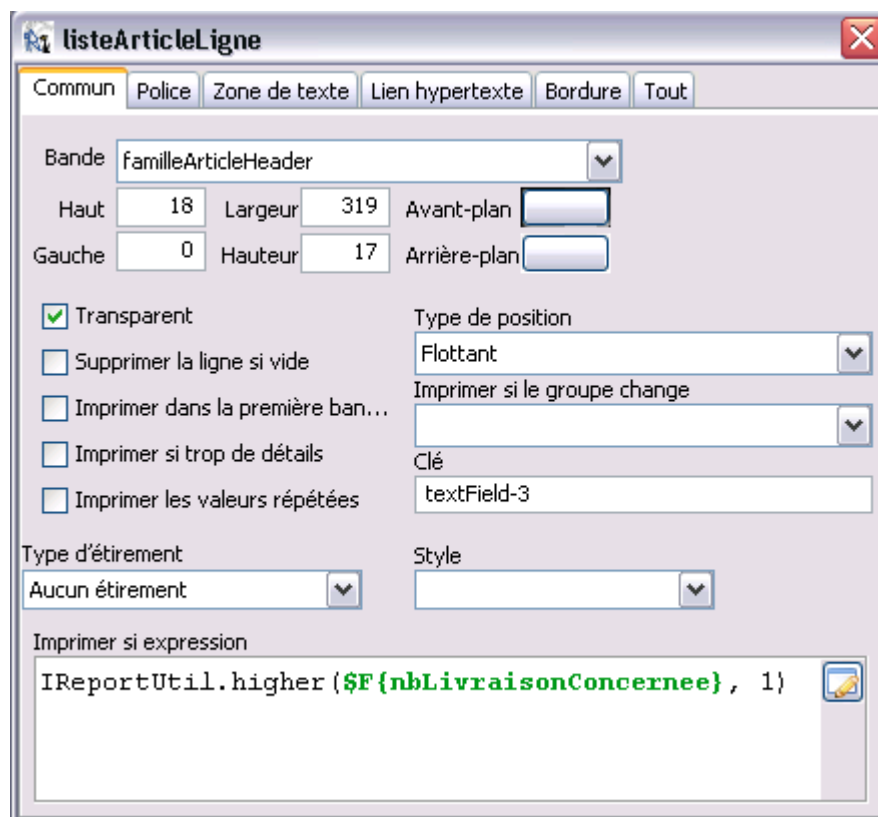
Les parenthèses sont utilisées pour séparer les conditions.



L'expression ci-dessus teste si le code de l'article contient « AY4 » ET commence par « 000 ».

Impression conditionnée :

L'exemple ci-dessous affichera la valeur du TextField si le champ nbLivraisonConcernee est supérieur à 1.



IX Bibliothèque de méthodes

Afin de formater l'affichage des numériques, des dates et des durées, Sylob propose une bibliothèque composée de 3 fichiers de type JAVA. Ces fichiers se trouvent dans l'arborescence du répertoire « com ».

Ci-dessous, le détail des méthodes proposées dans ces fichiers

- DureeUtil.class
getDureeArrondiMinute(long) = Arrondi la durée à la minute.
getDureeFromMillisecond(String, boolean) = Converti une durée en milliseconde. Le booléen définit si oui ou non secondes s'affiche.
getMillisecondFromDuree(String) = Retourne le nombre de milliseconde de la durée.
isValidDuree(String) = Retourne vrai si la durée est vraie.
- DateUtil.class
addDayToDate(Date, int) = Ajoute ou retire X jours à la date, sans tenir compte du calendrier
addHourToDate(Date, int) = Ajoute ou retire X heures à la date, sans tenir compte du calendrier
addMonthToDate(Date, int) = Ajoute ou retire X mois à la date, sans tenir compte du calendrier
addWeekToDate(Date, int) = Ajoute ou retire X semaines à la date, sans tenir compte du calendrier
addYearToDate(Date, int) = Ajoute ou retire X années à la date, sans tenir compte du calendrier
compareHeureFromDateFormat(Date, Date) = Compare 2 heures à partir de 2 dates. Retourne -1 pour antérieure, 0 pour identiques et 1 pour postérieure.
compareTo(Date, Date) = Compare 2 dates. Retourne -1 pour antérieure, 0 pour identiques et 1 pour postérieure
getDate(long) = Transforme un long en Date
getDateLongFromString(String, Locale) = Transforme la date en Date selon la locale
getDateWithoutHour(Date) = Retire les heures à la Date
getDay(Date) = Retourne le jour de la Date
getDifferenceInDays(Date, Date) = Retourne le nombre de jours positifs entre les Dates
getDureeFromDate(Date, Date) = Retourne un entier de différence entre les Dates.
getFirstDayOfMonth() = Retourne le premier jour du mois.
getFirstDayOfWeek(Date) = Retourne le premier jour de la semaine.
getHeureStringFormat(Date) = Retourne le format Heure d'un Date
getHours(Date) = Retourne les heures de la date.
getLastDayOfMonth(Date) = Retourne le dernier jour du mois
getLastDayOfWeek(Date) = Retourne le dernier jour de la semaine
getMessageKeyInitialeJourDate(Date) = Retourne la messageKey du jour
getMessageKeyLibelleMoisDate(Date) = Retourne la messageKey du mois
getMinutes(Date) = Retourne le nombre de minutes de la Date.
getMonth(Date) = Retourne le mois de la Date
getNbMoisBetween(Date, Date) = Retourne le nombre de mois entre les Dates.
getNumeroSemaine(Date) = Retourne le numéro de la semaine de la Date.
getPositiveDifferenceInDays(Date, Date) = Retourne le nombre positif de jour de différence entre les Dates.
getSeconds(Date) = Retourne les secondes de la Date
getString6DigitFromDate(Date, Locale) = Formate la date sur 6 caractères selon la locale
getString8DigitFromDate(Date, Locale) = Formate la date sur 8 caractères selon la locale
getTime(Date) = Retourne un Long correspondant à la Date
getYear(Date) = Retourne l'année de la Date.
isDateMinuit(Date) = Retourne vrai si la Date est positionnée à minuit.
parseDate(String, Locale) = Retourne la Date selon la locale
today() = Retourne la Date du jour.
- NumberUtil.class
formatNumber(BigDecimal, Integer, Locale) = Formate le nombre BigDecimal selon le nombre de décimale Integer et dans la locale.

`formatNumber(Double, Integer, Locale)` = *Formate le nombre Double selon le nombre de décimale Integer et dans la locale.*
`formatNumber(Double, Locale)` = *Formate le nombre Double selon le nombre de décimale Integer.*
`arrondir(BigDecimal, int)` = *Arrondit le nombre BigDecimal selon le nombre de chiffre après la virgule.*
`arrondir(double, int)` = *Arrondit le nombre double selon le nombre de chiffre après la virgule.*
`divide(BigDecimal, BigDecimal)` = *Divise le nombre BigDecimal par BigDecimal*
`formatPourcentage(BigDecimal, Locale)` = *Formate le pourcentage BigDecimal dans la Locale*
`formatNumberForField(BigDecimal, Integer)` = *Formate le BigDecimal en chaine*
`getNbdecimale(String)` = *Retourne le nombre de décimale du nombre.*
`isValidNumber(String, int, boolean)` = *Retourne vrai si le nombre est un nombre.*

Exemple : Retourne du numéro de semaine de la date
`DateUtil.getInstance().getNumeroSemaine(new Date(new Long(${dateCreation})))`

X Références

The Definitive Guide to iReport de Giulio Toffoli (Anglais)

The Definitive Guide to JasperReports de Teodor Danciu et L Chirita (Anglais)