

# Reporting avec Birt

---

David THIBAU – 2021

[david.thibau@gmail.com](mailto:david.thibau@gmail.com)



# Agenda

---

## Introduction

- BI et Reporting
- Présentation BIRT
- Installation et distributions
- Présentation de BIRT Designer
- Préparer le développement d'un rapport

## Données d'un rapport

- Source de données
- Jeu de données
- Liaison de données

## Composition d'un rapport

- Structure et types d'éléments
- Les conteneurs
- Éléments simples
- Formatage des éléments
- Page maître et sauts de page

## Manipulation des données

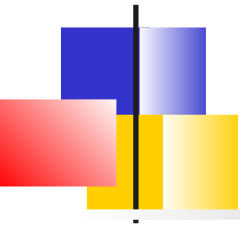
- Les expressions
- Tri et filtres
- Groupes et agrégation
- Paramètres

## Éléments avancées

- « Sous-rapports »
- Liens hypertexte
- Graphiques
- Tableaux croisés

## Pour aller + loin

- Internationalisation
- Développement collaboratif



# Introduction

---

Business Intelligence  
Présentation de BIRT  
Installation et distribution  
BIRT Designer  
Préparer le développement



# Business Intelligence

---

- L'informatique décisionnelle ou **Business Intelligence (BI)** regroupe les technologies et les applications qui permettent aux sociétés de mieux comprendre leur contexte commercial ou plus largement leur activité
- Ces technologies ont pour but de proposer des historiques, des vues courantes ou prévisionnelles de l'activité de l'entreprise
- Les fonctions communes du BI sont :
  - Le reporting
  - L'OLAP (Online Analytic Processing)
  - L'analyse de données (corrélation)
  - Le data mining (L'extraction de données)
- L'objectif étant l'aide à la décision.

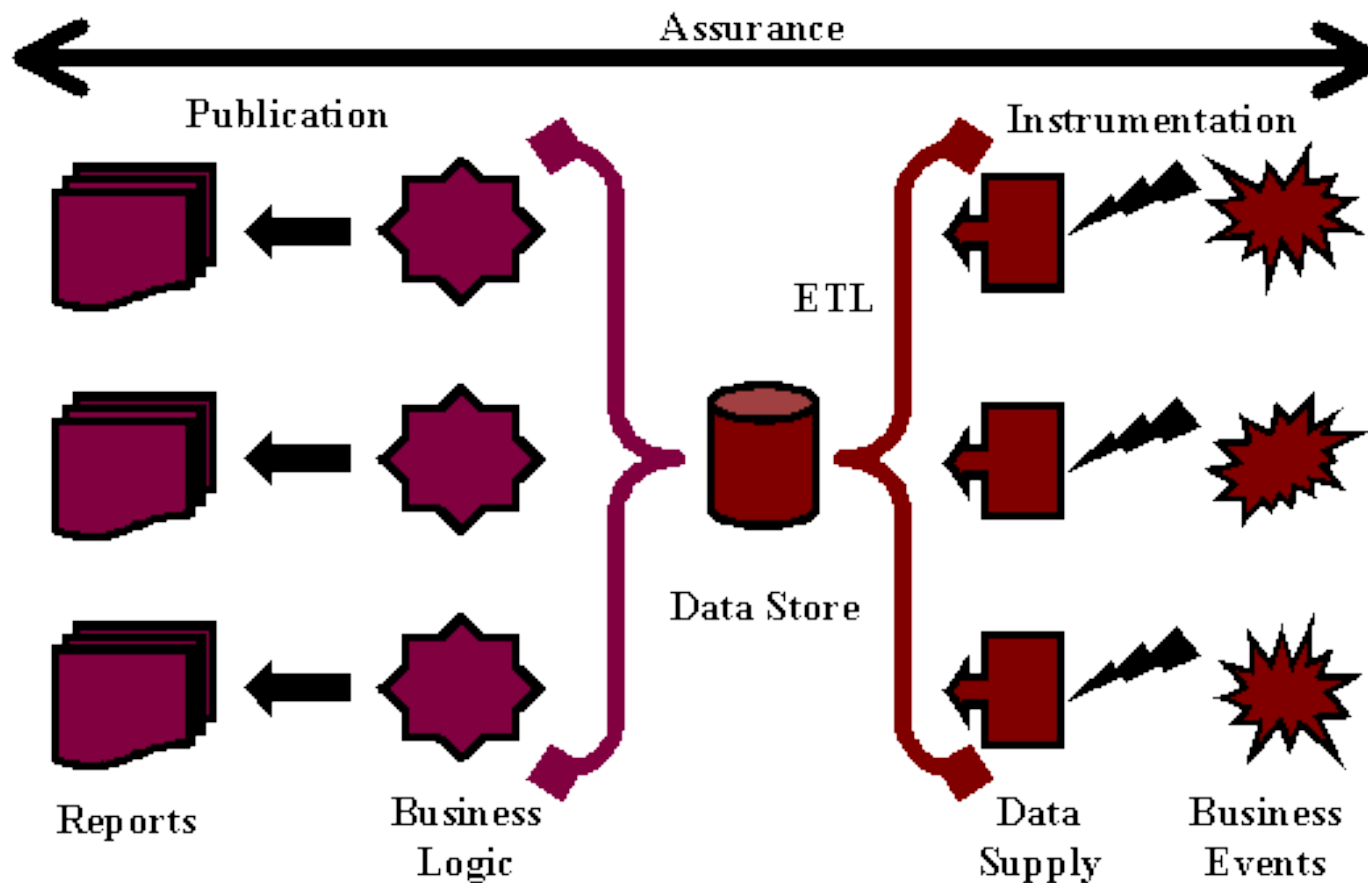


# Entrepôt de données

---

- Les outils et applications du BI utilisent en règle générale des **entrepôts de données** (*datawarehouse*) pour stocker des données transverses provenant de plusieurs sources hétérogènes
- Ces entrepôts sont construits grâce à des outils d'**ETL** (*Extract Transform Load* ou *datapumping*).
- A un degré de moindre, les activités de reporting utilisent des bases de données relationnelles classiques en exploitation.  
Quelquefois, le schéma peut être adapté pour faciliter le reporting.

# Modèle





# Reporting

---

Le **reporting** consiste donc à fournir régulièrement des informations à des décideurs d'une organisation.

- Formes : graphiques, textes, tableau
- Mise à disposition :
  - Typiquement une application web métier, intranet, portail d'entreprise
  - Alternativement, email ou format papier

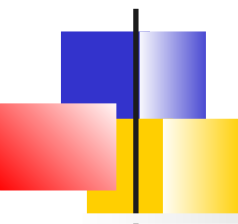


# Acteurs du marché

---

- Les acteurs traditionnels du marché datant des années 1970 ont souvent été intégrés dans les géants de l'informatique :
  - Business Objects → SAP
  - Cognos → IBM
  - Hyperion → Oracle
  - SAS
  - Microstrategy
  - Information Builders
- Ces acteurs sont concurrencés par des solutions plus légères et moins onéreuses, basées généralement sur des modèles OpenSource : *BIRT, JasperReport, ...*





# Présentation de BIRT



# Introduction

---

BIRT est un outil de reporting pour les applications **Java**.

Il est constitué de 2 principaux composants :

- Un **designer** basé sur l'IDE Eclipse utilisé par les concepteurs de rapport
- Un **moteur d'exécution** qui peut être déployé sur un serveur applicatif ou embarqué dans une application Java et qui permet de fournir les rapports aux utilisateurs finaux



# Rapports typiques

---

Différents types de rapports peuvent être mis au point par BIRT

- **Listes** : Les rapports les plus simples consistent en des listes de données pouvant être groupées et pouvant afficher des données agrégées
- **Graphiques** : BIRT fournit de nombreux types de graphiques pouvant être rendus en différents formats dont le format SVG supportant des interactions utilisateur
- **Tableaux croisés** : Permettant l'affichage de données agrégées suivant différentes dimensions.
- **Lettres et Documents** : Des rapports de type lettre, factures, etc. peuvent être facilement créés par BIRT.
- **Rapports composites** : Les rapports peuvent combiner les différents types dans un rapport dynamique. BIRT offre des possibilités de lier ces sous-rapports, de les disposer facilement, de gérer les sauts de page, etc.



# Composants BIRT

---

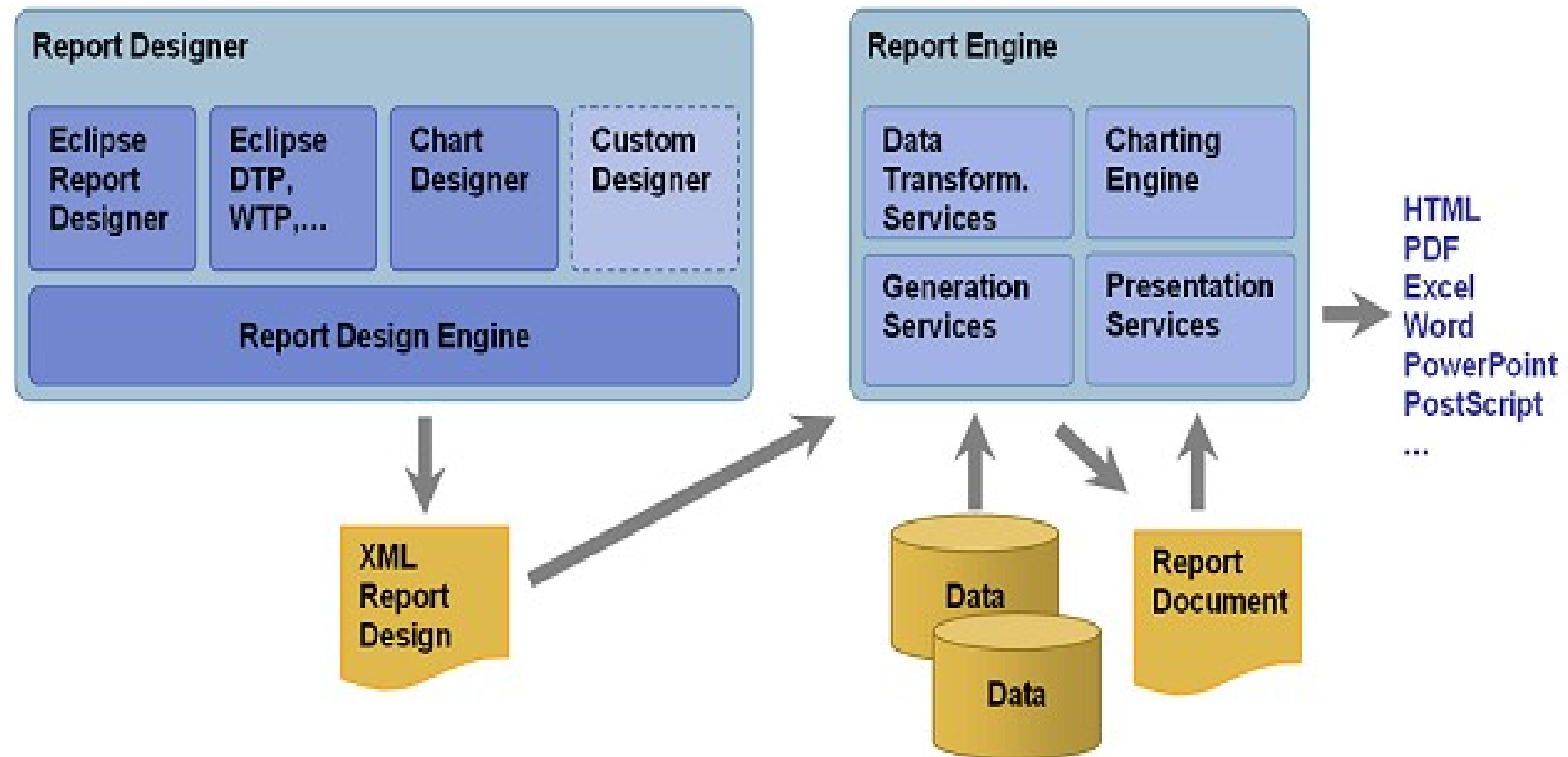
La projet OpenSource BIRT est décomposé en plusieurs parties qui chacune expose leur API Java.

- **Le moteur de Design** (*Design Engine API*) : API permettant de créer et modifier les rapports.
- **Le moteur de rapport** (*Runtime Engine API*) : Utilise les fichiers de design pour générer et rendre les rapports.
- **Le moteur des graphiques** (*Chart Engine API*) : Utilisé pour mettre au point et générer les graphiques.

2 outils utilisant ces APIs sont distribués par BIRT :

- **BIRT Report Designer** : Outil permettant de créer les rapports dans un format XML
- **BIRT Viewer** : Application web, utilisée pour prévisualiser les rapports dans le Report Designer.  
Elle est également distribuée indépendamment et peut être déployée sur un serveur JavaEE supportant JSP.

# Composants





# Apports de BIRT

---

La génération d'un rapport peut être vue comme un traitement en 4 étapes :

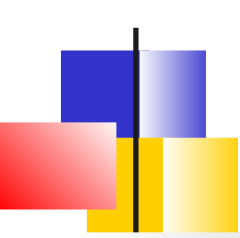
- **Extraction de données** : Pour cela, BIRT fournit un support pour JDBC, XML, les services Web et les fichiers à plat. Les données peuvent également provenir de code Java. Un rapport peut inclure des données de plusieurs sources et BIRT fournit un mécanisme pour lier des sources de données distinctes via des jointures.
- **Transformation de données** : BIRT permet de trier, filtrer, grouper et agréger les données extraites.
- **Logique métier** : Les données sources ne sont pas toujours structurées pour les besoins d'un rapport. BIRT permet d'ajouter ou de réutiliser de la logique métier pour transformer les données.
- **Présentation et formatage** : Une fois les données prêtes. BIRT propose de nombreux éléments graphiques et outils de formatage et de style



# Transformation de données

---

- **Colonnes calculées** : Permettent de définir d'autres données que celles retournées par la requête. La logique peut être une expression simple, un script ou un appel à du code Java.
- **Paramètres d'entrée et de sortie** : agissant sur les requêtes SQL ou les procédures stockées.
- **Méta-données** pour les colonnes : Des alias peuvent être fournis pour faciliter la manipulation de données dans le rapport et sa maintenabilité
- **Tri et Filtre** : Des filtres en javascript ou java permettent de filtrer les données provenant de la requête
- **Groupe** : Des critères de regroupement sont définis et permettent d'organiser les données en groupe. Des calculs d'agrégation peuvent être effectués



# Intégration de code métier

---

BIRT intègre un **moteur Javascript** permettant d'appeler de la logique métier à partir des rapports BIRT.

BIRT expose une **API Javascript** permettant d'accéder au ROM (Report Object Model) incluant les aspects design et exécution du rapport

En plus de Javascript, des **gestionnaires d'événements** peuvent être codés en **Java**. BIRT fournit un ensemble d'interfaces pour la création d'élément de rapport et la récupération de données. Le code Java peut être débuggé avec le debugger d'Eclipse





# Formatage conditionnel

---

BIRT fournit plusieurs fonctionnalités de formatage conditionnel :

- **Visibilité** : Les éléments d'un rapport peuvent être cachés selon différents types de conditions
- **Association de valeur** : Les codes de la base de données peuvent être associés à des libellés plus explicites
- **Mises en valeur** : De simples expressions permettent de sélectionner le style à appliquer sur un élément



# Internationalisation

---

BIRT fournit un support pour l'internationalisation et la localisation.

- **Localisation de texte** : Un rapport peut être généré en différentes langues
- **Packages de langues** : Des packages de langue permettent d'exécuter BIRT en différentes langues.
- **Localisation des données** : Les données peuvent être formatées en fonction de la locale (date, devise, etc.)
- **Formatage dynamique** : BIRT ajuste automatiquement la taille des éléments du rapport permettant de s'adapter à des langues très différentes
- **Support bidirectionnel** : BIRT supporte le formatage bidirectionnel permettant de supporter des langues s'écrivant de la droite vers la gauche.



# Projet de reporting

---

BIRT s'intègre avec les capacités de **gestion de projet d'Eclipse** pour organiser les rapports. En particulier, les rapports peuvent être stockés dans des outils de gestion des sources : CVS/SVN/Git

BIRT permet de **mutualiser les styles** utilisés dans les rapports ou de réutiliser des fichiers css d'une application web

BIRT supporte les **bibliothèques** qui permettent de mutualiser tous les types d'éléments utilisés dans les rapports. Les différents rapports d'un même projet peuvent alors faire appel aux mêmes bibliothèques



# Extensibilité

---

BIRT fournit plusieurs points d'extensions :

- Le framework « **Data Tools Open Data Access** » (ODA) permet ajouter d'autres sources de données.
- Les **éléments** utilisables dans les rapports peuvent également être étendus
- Des plugins pour les **graphiques** peuvent être ajoutés
- Des **convertisseurs** peuvent être développés pour supporter d'autres formats de sortie



# ROM et XML

---

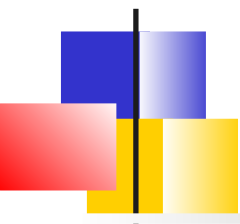
BIRT utilise un modèle de document simple.

Un design n'est rien d'autre qu'un fichier XML avec l'extension ***.rptdesign***

Le fichier XML contient toutes les informations nécessaires pour la génération du rapport

L'ensemble des balises XML supportées par BIRT est appelé le ***Report Object Model (ROM)***

La documentation est consultable à  
<http://www.eclipse.org/birt/phoenix/ref/>



# Installation et distribution



# Application designer

---

Il existe 2 applications permettant d'élaborer des rapports BIRT :

- ***BIRT Report Designer*** qui nécessite l'IDE Eclipse et certains de ses composants. Cette distribution fournit toutes les fonctionnalités de designer plus la possibilité de modifier la couche Java ou javascript de BIRT
- ***BIRT RCP Report Designer*** est un module standalone basé sur la plateforme RCP d'Eclipse qui nécessite uniquement un runtime Java



# Installation

---

Le téléchargement s'effectue à partir du site BIRT : (<http://www.eclipse.org/birt/phoenix/>)

Plusieurs distributions sont disponibles :

- ***All-in-one*** : Eclipse + BIRT Designer
- ***Birt Designer*** : Seulement les plugins Eclipse (Eclipse doit être déjà installé)
- ***RCP report Designer*** : Le module autonome





# Installation All-in-one

---

Une fois l'archive correspondant au système cible téléchargée, il suffit de la décompresser dans un répertoire de travail.

L'archive crée un sous-répertoire *eclipse* dans lequel se trouve l'exécutable

Au démarrage d'eclipse, ouvrir la perspective *Report Design*

*Window → Open Perspective → Report Design*



# Installation BIRT RCP

---

BIRT RCP ne s'exécute que sur Windows

Après avoir téléchargé l'archive, la décompresser dans un répertoire de travail

L'extraction crée un sous-répertoire *birt-rcp-report-designer* contenant l'exécutable *BIRT.exe*




# Ajout d'une langue

---

Il est possible d'installer un package de langue afin d'avoir l'interface en français par exemple.

L'installation s'effectue :

- Soit via un téléchargement direct,  
l'archive doit être décompressée dans  
*eclipse/folder*
- Soit via l'Update site d'Eclipse  
<http://download.eclipse.org/technology/babel/update-site/juno>



# Mise à jour

---

Les mises à jour du designer peuvent s'effectuer via l'Update Manager d'Eclipse

*Help → Software Updates → Find and Install*

Le module autonome nécessite lui une réinstallation



# Déploiement

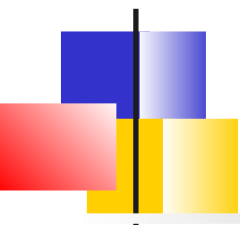
---

Une fois réalisés, les rapports peuvent être fournis aux utilisateurs finaux via une application web.

BIRT fournit une application web prête à l'emploi et facilement déployable sur un serveur web Java tel que Tomcat : **Birt Web Viewer**

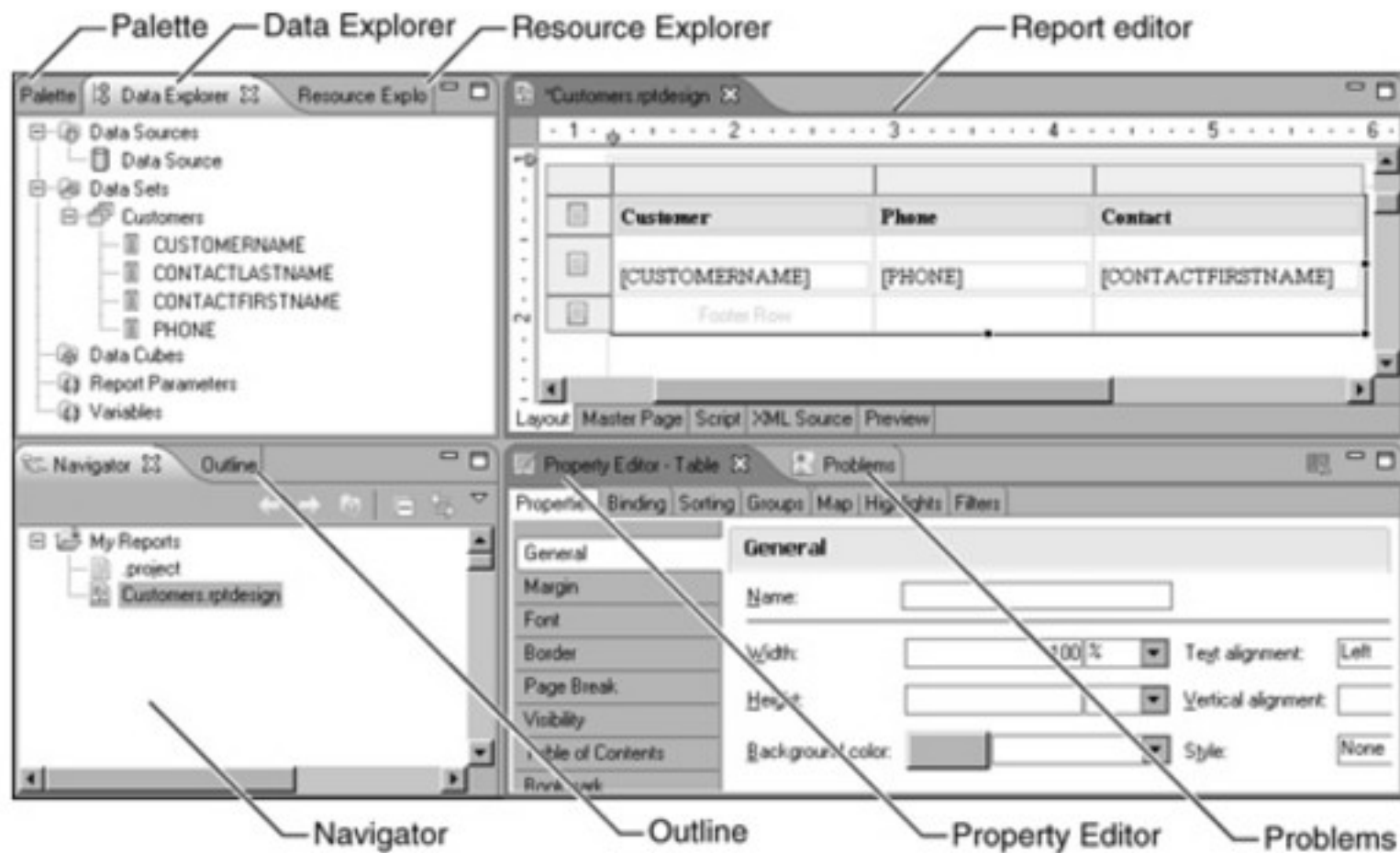
L'application peut être adaptée en :

- Configurant son descripteur de déploiement (*web.xml*)
- En adaptant, le style ou les pages JSP de l'archive



# Présentation de Birt Designer

# Vues BirtDesigner





# Éditeur de rapport

---

La fenêtre permet de mettre au point et prévisualiser le rapport.

Il est possible d'ouvrir plusieurs rapports.

L'éditeur contient 5 onglets :

- **Le layout** : Fenêtre principale de développement
- **La page maître** : Les éléments apparaissant sur toutes les pages du rapport
- **L'éditeur de script** : Permet l'insertion de code Javascript
- **La source XML** : Permet de voir le source
- **La prévisualisation<sup>1</sup>** : Permet de démarrer le BIRT viewer intégré

1. Dans certaine distribution, la prévisualisation est disponible par le menu *Run* → *View Report*





# Les autres vues

---

**La palette** : Éléments disponibles, drag & drop

**L'explorateur de données** : les sources, les jeux de données, les paramètres

**L'explorateur de ressources** : Les ressources partagées par tous les rapports (images, feuilles de style, ensembles d'éléments)

**L'éditeur de propriété** : Permet de visualiser et modifier les propriétés de l'élément sélectionné

**Le navigateur** : Visualise tous les projets et leurs rapports

**La vue Outline** : Visualisation arborescente du rapport proche du XML. Permet la sélection d'éléments

**La vue Problème** : Affiche les erreurs. Les erreurs sont cliquables et permettent d'accéder directement à l'élément du design en erreur



# Prévisualisation

---

Il est possible de prévisualiser le rapport dans une fenêtre séparée en utilisant le bouton prévisualiser ou le menu : *Run → View Report*

Le format de prévisualisation peut être :

- HTML dans l'application *Web Viewer*
- HTML dans une fenêtre du navigateur par défaut
- PDF
- Postscript
- Format Libre Office
- Format Microsoft Office



# Birt Report Viewer

---

Ce visualiseur offre plusieurs fonctionnalités :

- une **table des matières** permettant de naviguer dans le rapport généré (section ou pages)
- **Ré-exécuter** un rapport pour rafraîchir les données
- **Imprimer** un rapport
- **Exporter** un rapport dans un des format supporté

# Birt Report Viewer

Table of contents

Page navigation

BIRT Report Viewer

BIRT Report Viewer

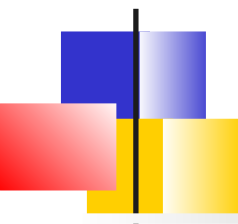
Showing page 1 of 1

Go to page:

- + Classic Cars
- + Motorcycles
- + Planes
- [-] Ships
  - S18\_3029
  - S24\_2011
  - S700\_1138
  - S700\_1938
  - S700\_2047
  - S700\_2610
  - S700\_3505
  - S700\_3962
  - S72\_3212

Classic Models Corporation

Product Code	Product Name	Product Vendor	Qty in Stock	Price Each	MSRP	Orders
Classic Cars						
S10_1949	1952 Alpine Renault 1300	Classic Metal Creations	7305	98.58	214.3	678
S10_4757	1972 Alfa Romeo GTA	Motor City Art Classics	3252	85.68	136	705
S10_4962	1962 LanciaA Delta 16V	Second Gear Diecast	6791	103.42	147.74	693
S12_1099	1968 Ford Mustang	Autoart Studio Design	68	95.34	194.57	586
S12_1108	2001 Ferrari Enzo	Second Gear Diecast	3619	95.59	207.8	548



# Préparer le développement d'un rapport



# Étapes de mise au point

---

1. Spécifier le rapport (par ex. Le dessiner sur un bout de papier)
2. Démarrer un projet Eclipse et créer un design à partir d'un gabarit ou d'une feuille blanche
3. Spécifier les données à utiliser (La source de données et le jeu de données)
4. Choisir et disposer les éléments du rapport
5. Formater les éléments (Appliquer des styles)
6. Mettre au point la page maître
7. Tester et prévisualiser (Tester le jeu de données puis le résultat et éventuellement le code ajouté)

Le processus est itératif



# Planification du rapport

---

Répondre aux questions :

- ✓ Quel est l'objectif du rapport ?  
*Montrer les ventes mensuelles par région, puis par commerciaux et mettre en valeur les ventes inférieures à un certain niveau*
- ✓ Qui va lire le rapport ?
- ✓ Quelles données doivent apparaître ?
- ✓ D'où les données proviennent ?
- ✓ Des données doivent-elles être calculées ?  
Comment (Expression, javascript, ...) ?
- ✓ L'utilisateur peut-il sélectionner les données à visualiser ?  
(Paramètres)



# Format d'affichage final

---

Le design doit être fait dans les conditions finales de visualisation (taille de la page, format de visualisation, polices, couleurs, ....)

- ✓ Faut il privilégier le print ou la visualisation online ?
- ✓ Quel est le format privilégié ?
- ✓ Les navigateurs supportés ?
- ✓ Le rapport supporte l'internationalisation ?

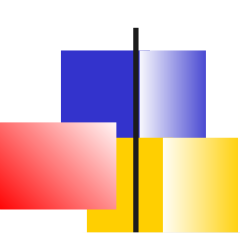




# Disposition et format

---

- ✓ Quelle disposition pour les données du rapport ?  
Section unique ou multiple
- ✓ Doit on grouper les données ?
- ✓ Quel ordre de tri ?
- ✓ Doit on faire un résumé ?
- ✓ Doit on mettre en valeur certaines données sous certaines conditions ?
- ✓ De quoi sont constitués les entêtes et bas de page ?
- ✓ Doit on utiliser la charte « corporate » ? un style css disponible sur un site web ?
- ✓ Y a t il des gabarits à utiliser ?



# Réutilisation de composants

---

- ✓ Combien de rapports doivent être développés ?
  - ✓ Est-ce que les rapports ont des éléments communs ?  
Source de données, styles, entêtes, ...
- => Utilisation des bibliothèques et des gabarits



# Gestion des ressources externes

---

Un rapport utilise des ressources externes (images, bibliothèque, fichiers *.java* ou *.properties*)

BIRT propose un **dossier de ressources** pour organiser ces fichiers externes et faciliter leur déploiement



# Déploiement

---

- ✓ Comment le rapport est distribué aux utilisateurs ?

*Application, email*

- ✓ Comment le rapport sera généré ?

*Par les utilisateurs en temps-réel, par des traitements batch*



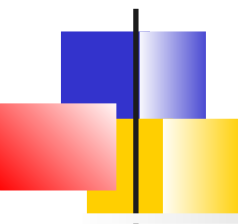
# Données d'un rapport

---

Les sources de données

Le jeu de données

Les liaisons de données (*bindings*)



## Sources de données



# Introduction

---

- ♦ Une **source de données** est un objet qui encapsule les informations de connexions au système stockant les données.
- ♦ Chaque type de source nécessite des informations particulières
- ♦ Un rapport peut contenir plusieurs sources de données de différents types



# Data sources supportées

---

- **Connexion JDBC** : Connexion à une base de données relationnelles en Java
- **Fichier texte à plat CSV ou Excel :**
- **Fichier XML, Services web SOAP**
- **MongoDB, Hive**
- **Script** : Une classe Java et des scripts remplissent le jeu de données

Les développeurs peuvent également utiliser l'API ODA pour écrire des pilotes personnalisés





# Rappels sur JDBC

---

- Une connexion JDBC permet à un programme Java de se connecter à une base de données relationnelles (toutes les bases sont supportées)
- Pour cela, il est nécessaire d'utiliser les pilotes JDBC (bibliothèque Java) fournis par l'éditeur de la base. Les pilotes sont positionnés dans le classpath
- Configurer une connexion JDBC consiste à fournir les informations suivantes :
  - La **classe principale** du Driver
  - L'**URL** d'accès à la base (syntaxe dépendante de l'éditeur)
  - Un **compte** base de données valide (login/mot de passe)
- *BIRT* fournit les drivers JDBC pour Derby



# Erreurs communes

---

- 3 types d'erreurs peuvent survenir lorsque l'on configure une connexion JDBC :
  - **ClassNotFoundException** : Le driver n'a pas pu être chargé. Il n'est pas présent dans le classpath
  - **L'URL** n'est pas correcte.  
Vérifier le format de l'URL chez l'éditeur
  - Les **paramètres de connexion** ne sont pas corrects.  
Vérifier le nom de la base, l'utilisateur et le mot de passe.



# Explorateur de données

---

Une source de données JDBC se crée via l'explorateur de données

Les informations à saisir sont :

- La classe du driver (liste déroulante des drivers installés dans Eclipse)
- L'URL jdbc dépendant de l'éditeur
- Les créidentiels d'accès à la base
- Eventuellement, le nom JNDI du pool de connexion si l'on est dans un environnement serveur

Un bouton de tester la connexion



# Installation d'un driver

---

Si le driver n'est pas dans la liste, le bouton « Manage Drivers » permet d'en ajouter.

Il faut alors indiquer le chemin vers le fichier *.jar* téléchargé sur le site de l'éditeur

Eclipse copie alors le driver dans le répertoire :  
`eclipse/plugins/org.eclipse.birt.report.data.oda.jdbc_<version>/drivers`

Il est possible d'associer un nom et un gabarit d'URL au driver



# Fichier texte

---

Un rapport BIRT peut se connecter à un fichier texte (ou Excel) respectant la structure suivante :

- La première ligne peut contenir les noms des colonnes séparés par des virgules.  
Si cette ligne n'existe pas, BIRT assigne les noms Column\_1, Column\_2, etc..
- La seconde ligne peut spécifier le type des données de chaque colonne dans le même ordre que la première ligne en utilisant comme caractère séparateur : , ; \t |
- Les lignes restantes contiennent les données séparées par : , ; \t |
- Chaque ligne doit contenir le même nombre de champs
- Il ne doit pas avoir de ligne vide
- Le caractère " permet d'ajouter les caractères séparateurs comme données



# Types de données

---

Abbreviation	Data type	Examples
BIGDECIMAL	java.sql.Types.NUMERIC	
DATE	java.sql.Types.DATE	YYYY-MM-DD or MM/DD/YYYY Examples: 2003-01-31 01/31/2003
DOUBLE	java.sql.Types.DOUBLE	
INT	java.sql.Types.INTEGER	
STRING	java.sql.Types.VARCHAR	
TIME	java.sql.Types.TIME	hh:mm:ss Examples: 12:59:59 AM 12:59:59 pm
TIMESTAMP	java.sql.Types.TIMESTAMP	YYYY-MM-DD hh:mm:ss.nnnnnn



# Exemple

---

Nom, Prénom, ID, Date

STRING, STRING, INT, TIMESTAMP

"Dupont", "Eric", 5, 01/31/2003 09:59:59 AM

"Dupond", "Paul, Henri, Gaston", ,



# Spécification la source

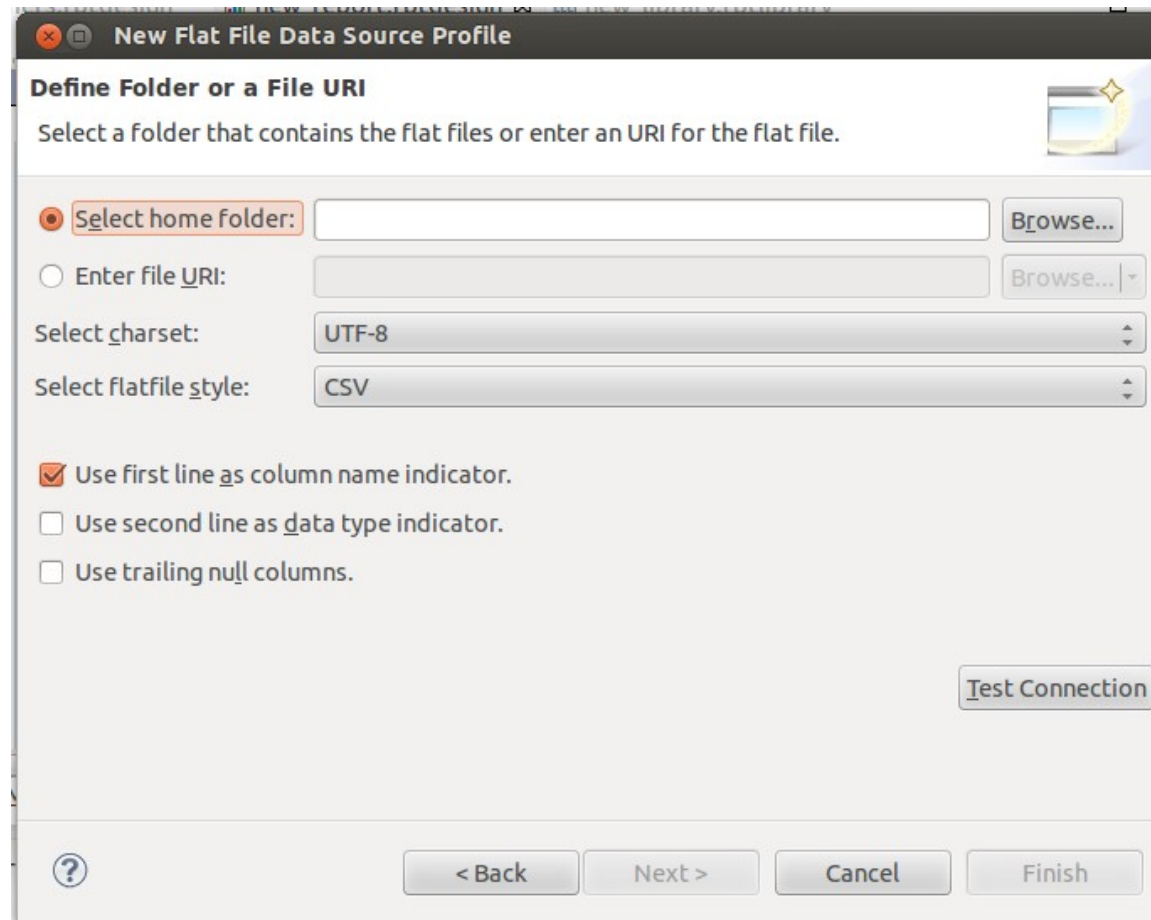
---

Dans le cas d'un fichier à plat, les informations à renseigner sont :

- L'emplacement du fichier
- Le jeu de caractères
- Le caractère séparateur
- La présence ou nom des 2 premières lignes spéciales



# Fichier à plat



The image shows a 'New Flat File Data Source Profile' dialog box. It has a title bar with standard window controls. The main area is titled 'Define Folder or a File URI' and contains instructions: 'Select a folder that contains the flat files or enter an URI for the flat file.' There are two radio buttons: 'Select home folder:' (selected) and 'Enter file URI:'. Each has a text field and a 'Browse...' button. Below these are two dropdown menus: 'Select charset:' set to 'UTF-8' and 'Select flatfile style:' set to 'CSV'. There are three checkboxes: 'Use first line as column name indicator.' (checked), 'Use second line as data type indicator.' (unchecked), and 'Use trailing null columns.' (unchecked). A 'Test Connection' button is at the bottom right. The footer contains a help icon, '< Back', 'Next >', 'Cancel', and 'Finish' buttons.

**New Flat File Data Source Profile**

**Define Folder or a File URI**  
Select a folder that contains the flat files or enter an URI for the flat file.

☒ Select home folder:  **Browse...**

☐ Enter file URI:  **Browse...**

Select charset: **UTF-8**

Select flatfile style: **CSV**

☒ Use first line as column name indicator.  
☐ Use second line as data type indicator.  
☐ Use trailing null columns.

**Test Connection**

**< Back** **Next >** **Cancel** **Finish**



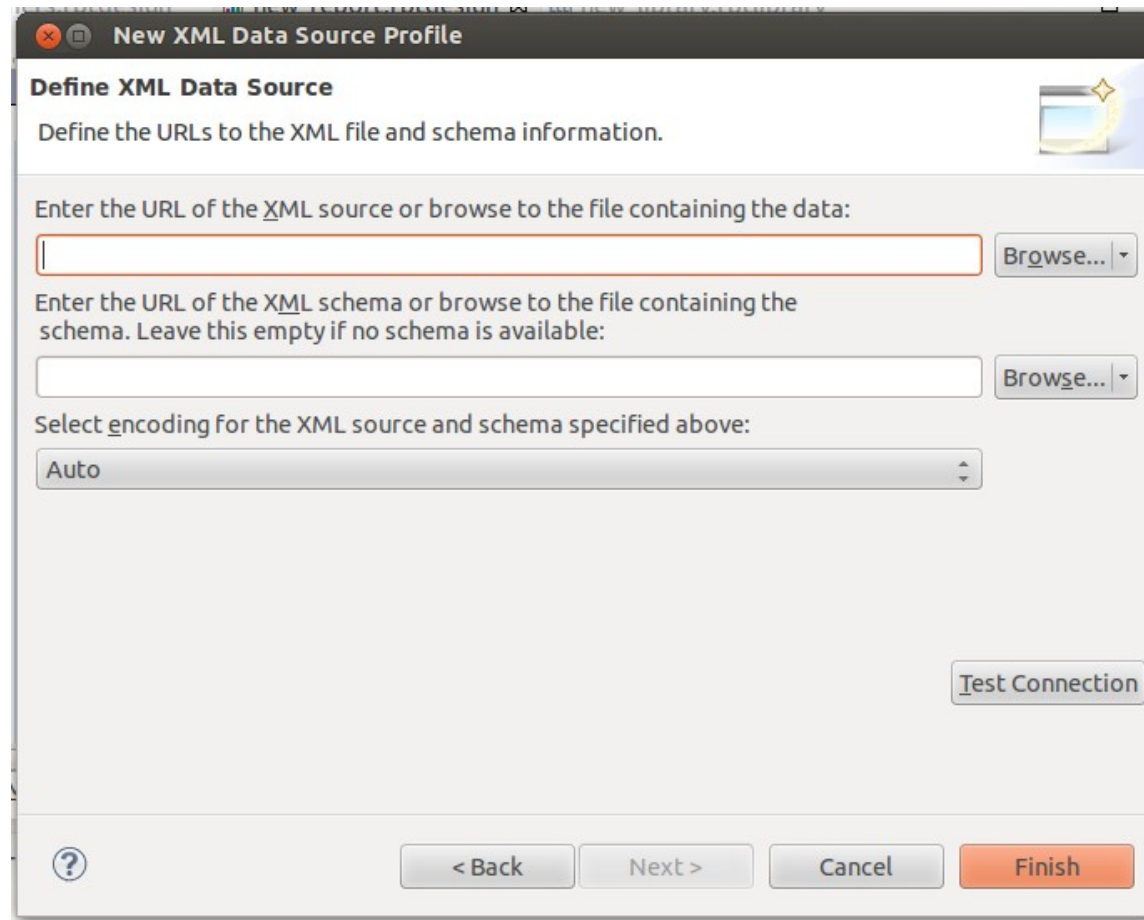
# Fichier XML

---

La connexion à un fichier XML consiste à fournir :

- Son emplacement
- Éventuellement, l'emplacement du schéma
- Le jeu de caractères

# XML



**New XML Data Source Profile**

**Define XML Data Source**  
Define the URLs to the XML file and schema information.

Enter the URL of the XML source or browse to the file containing the data:

Browse...

Enter the URL of the XML schema or browse to the file containing the schema. Leave this empty if no schema is available:

Browse...

Select encoding for the XML source and schema specified above:

Auto

Test Connection

? < Back Next > Cancel Finish



# Service web

---

Les services Web utilisent :

- WSDL pour décrire les opérations fournies par le service web
- SOAP pour transférer les données. (protocole au dessus d'HTTP)
- XML pour structurer les données

Lors de la création d'une source BIRT,  
l'information à préciser est l'URL WSDL



# Autres sources de données

---

BIRT supporte également d'autres types de sources de données :

- MongoDB
- Classes Java POJO
- Sources de données scriptées  
(intégration à une source de données via Javascript)
- Data warehouse Apache Hive



# Réutilisation des sources de données

---

Afin d'éviter de renseigner les données de connexion à chaque rapport. On peut :

- Définir un profil de connexion Eclipse
- Ou stocker la source de données dans la bibliothèque



# Explorateur de source de données

---

Pour faire apparaître l'explorateur :

*Windows → Show view → Other*

Puis

*Data Management → Data Source Explorer*

Les profils sont stockés par défaut dans

`./metadata/.plugins/org.eclipse.datatools.connectivity/ServerProfiles.dat`

Une fois un nouveau profil défini, il peut être utilisé dans l'assistant de création de source de données en faisant référence au fichier d'Eclipse stockant les profils



# Propriétés de connexion à l'exécution

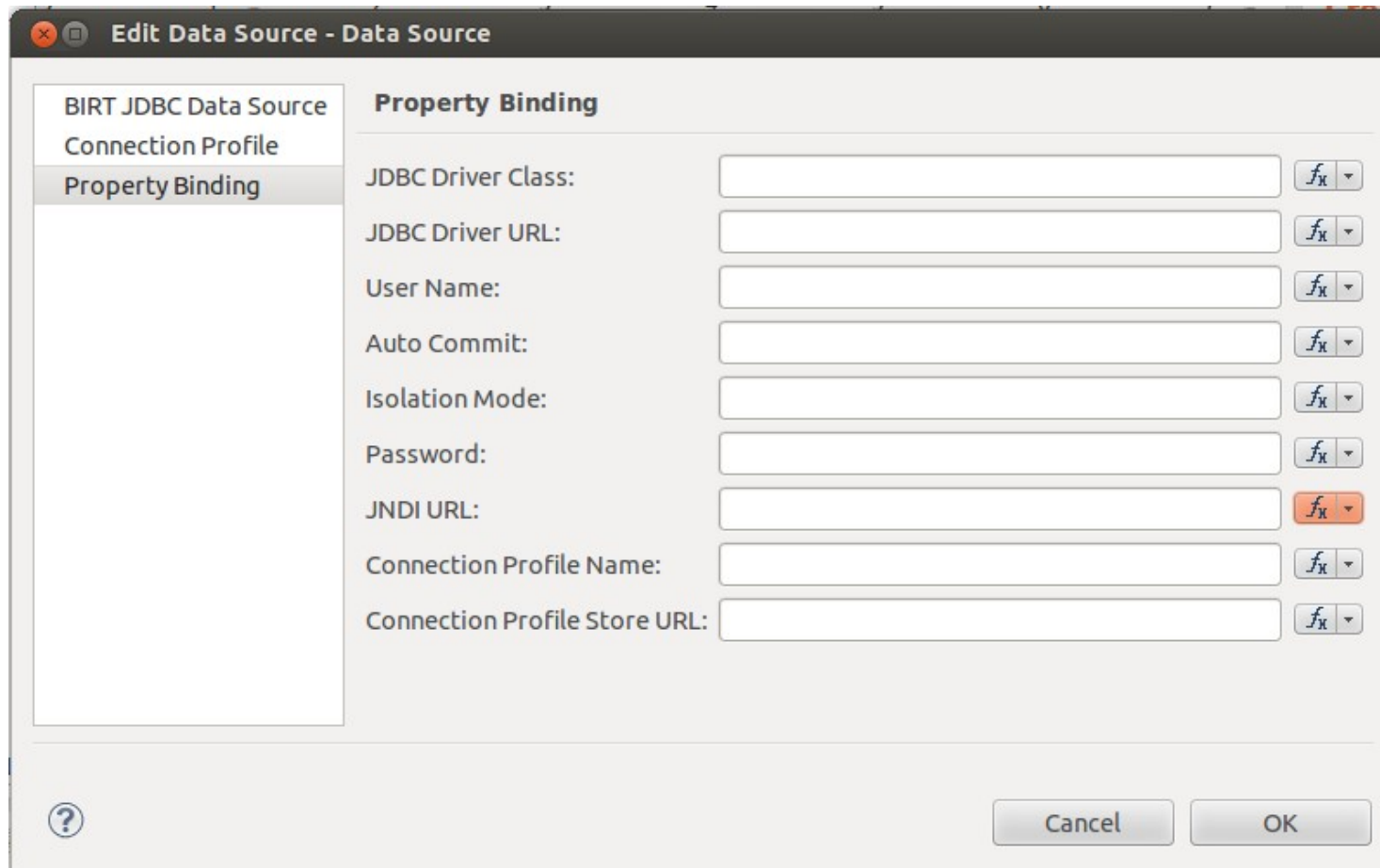
---

Au lieu d'avoir les informations en dur dans le rapport, il est possible de résoudre les informations de connexions à l'exécution du rapport

- Cette technique facilite le déploiement en production
- Les propriétés de connexion sont alors associées à une expression Javascript



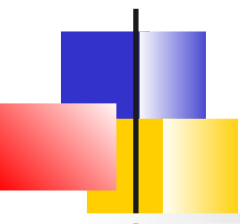
# Association de propriétés



The screenshot shows a dialog box titled "Edit Data Source - Data Source". On the left, there is a sidebar with three items: "BIRT JDBC Data Source", "Connection Profile", and "Property Binding", with the last one selected. The main area is titled "Property Binding" and contains several input fields, each with a function icon (fX) to its right:

- JDBC Driver Class:
- JDBC Driver URL:
- User Name:
- Auto Commit:
- Isolation Mode:
- Password:
- JNDI URL:
- Connection Profile Name:
- Connection Profile Store URL:

At the bottom left is a help icon (question mark in a circle). At the bottom right are "Cancel" and "OK" buttons.



# Jeux de données



# Introduction

---

- Un **jeu de données** définit toutes les données disponibles dans un rapport.
- BIRT fournit des assistants pour créer les jeux de données à partir d'une source
- Les données extraites de la source peuvent être traitées (renommage, calcul, filtre, ...)
- Un rapport peut contenir plusieurs jeux de données
- Les éléments itératifs d'un rapport peuvent partager le même jeu de données



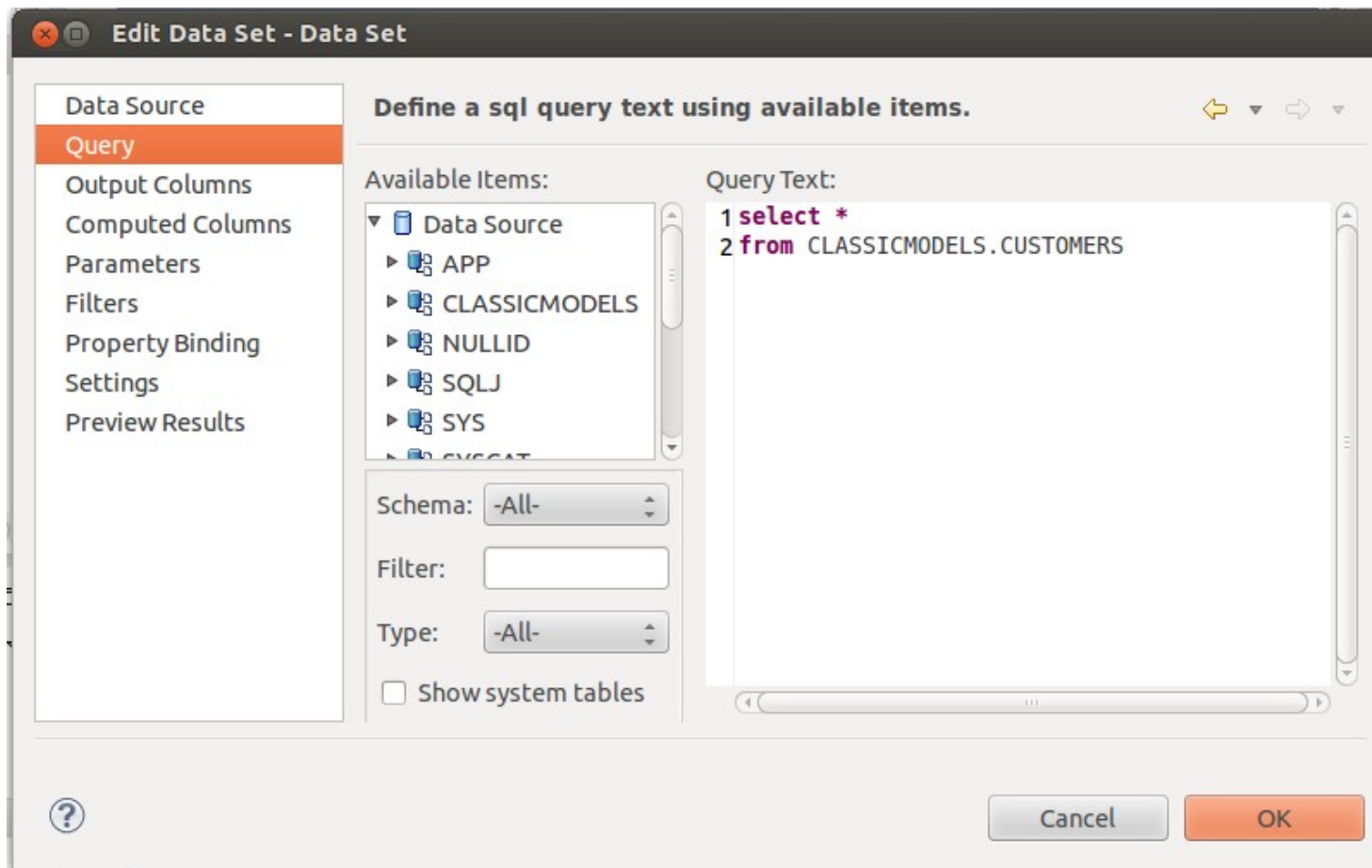
# Éditeur de jeux de données

---

L'éditeur comporte plusieurs onglets :

- **Query** : Affiche et édite la requête
- **Data Source** : Affiche le nom de la source de données.
- **Preview Results** : Permet de prévisualiser les résultats retournés
- **Computed Columns** : Colonnes calculées par BIRT via des expressions JavaScript ou appel Java
- **Parameters** : Paramètres utilisés par la requête pour remplacer les marqueurs "?"
- **Filters** : Filtres additionnels Javascript que BIRT applique.
- **Output Columns** : Affiche les colonnes de la requête et permet de définir des alias utilisés dans le rapport
- **Settings** : Configuration de la requête
- **Property Binding** : Associe une propriété du jeu de données à une expression Javascript

# Éditeur





# Vue tabulaire

Quelque soit la source (JDBC, Fichier XML), un jeu de données est un tableau constitué de ligne et de colonnes sur lequel le moteur BIRT est capable d'itérer

<b>1</b>	36001	36140	Aigurande	ut
<b>2</b>	36002	36150	Aize	ut
<b>3</b>	36003	36120	Ambrault	ut
<b>4</b>	36004	36210	Anjouin	utvatan
<b>5</b>	36005	36120	Ardentes	utvatan
<b>6</b>	36006	36200	Argenton-sur-Creuse	utlachatre
<b>7</b>	36007	36500	Argy	utleblanc
<b>8</b>	36008	36700	Arpheuilles	utleblanc
<b>9</b>	36009	36330	Arthon	utvatan
<b>10</b>	36010	36290	Azay-le-Ferron	utleblanc
<b>11</b>	36011	36210	Bagneux	utvatan
<b>12</b>	36012	36270	Baraize	utlachatre
<b>13</b>	36013	36110	Baudres	utvatan



# Jeu de données JDBC

---

Un jeu de données JDBC nécessite une **requête SQL**

- Il n'y a pas de limitation sur la requête (jointure, utilisation de vues ou procédures stockées)

BIRT permet de visualiser les éléments du schéma JDBC de les filtrer par type ou par leur nom et d'utiliser le glisser/déposer pour construire la requête



# Utilisation d'une procédure stockée

---

Une procédure stockée peut :

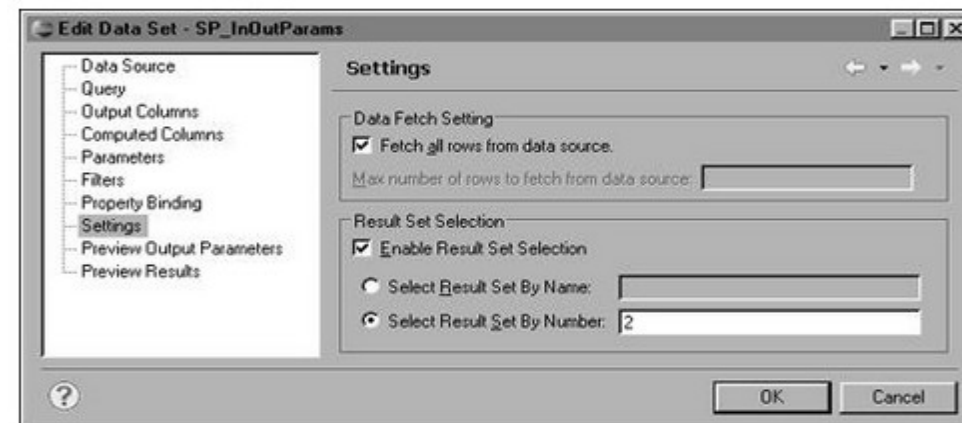
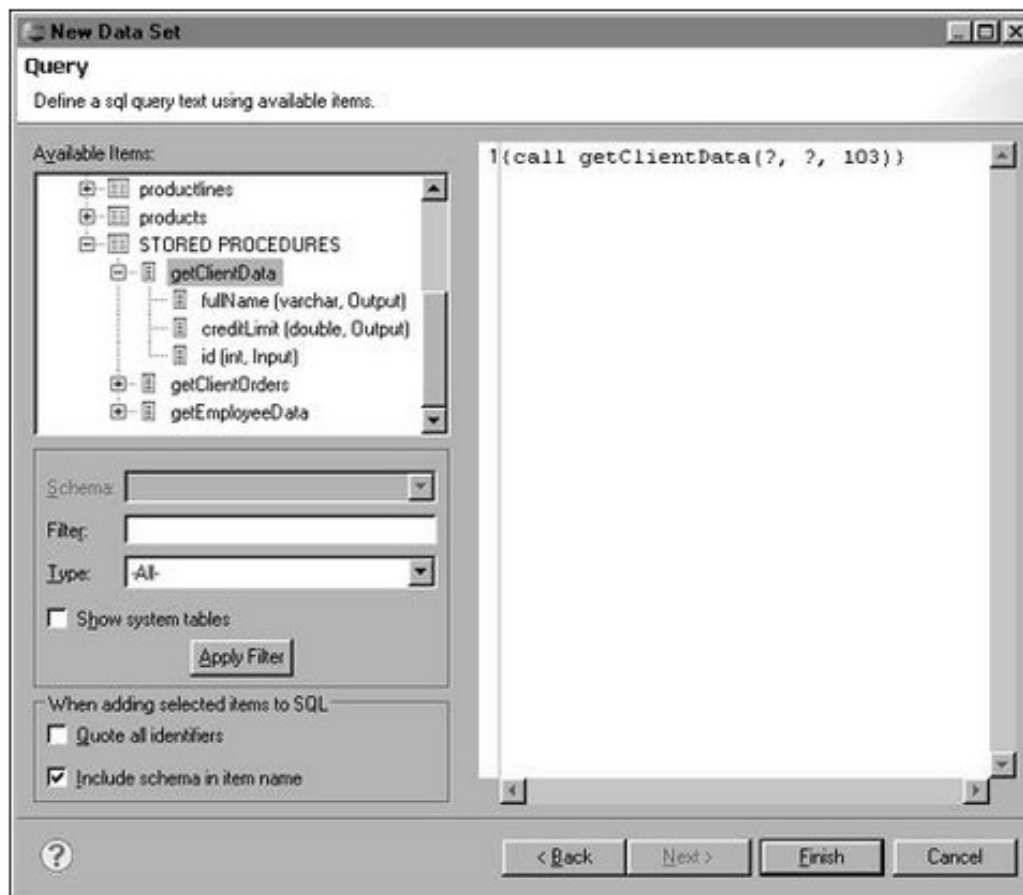
- Retourner un ou plusieurs tableaux de données
- Accepter des paramètres d'entrée
- Utiliser des paramètres de sortie pour retourner des valeurs

BIRT permet de visualiser les procédures stockées d'un schéma, de les glisser-déposer pour générer l'appel SQL

- Sélectionner le result set à utiliser
- Passer les paramètres d'entrée
- Prévisualiser les paramètres de sortie



# Exemple





# Jeu de données texte

---

Lors de l'utilisation d'une source de données texte, BIRT permet de :

- sélectionner les colonnes à utiliser
- renommer les champs



# Jeu de données XML

---

L'assistant utilise des expressions ***XPath*** pour sélectionner les données du document XML

- L'assistant permet de définir un élément XML comme l'élément de haut-niveau définissant les lignes.
- Les colonnes du tableau consistant en des attributs ou d'autres éléments peuvent alors être via *XPath* relativement à l'élément de haut-niveau



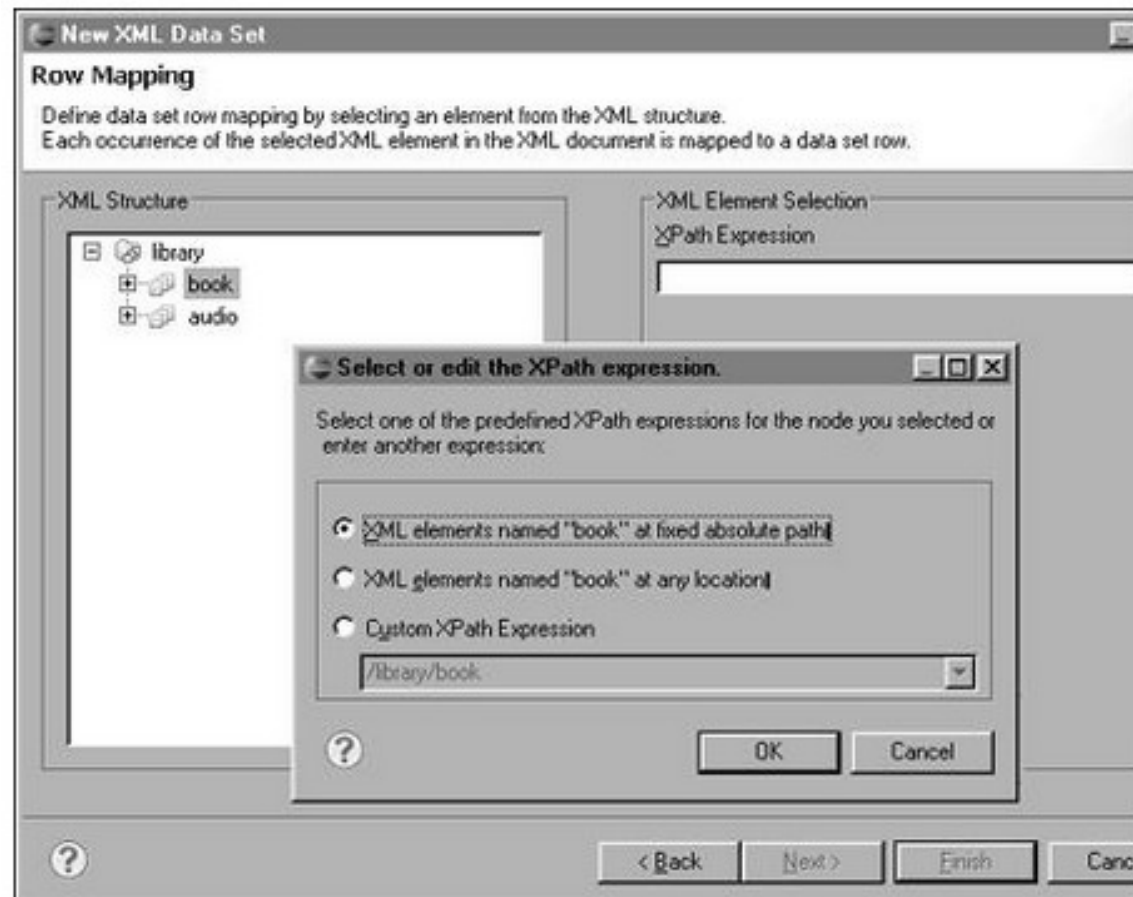
# Assistant *XPath*

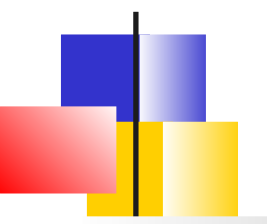
---

BIRT propose un support pour les requêtes *XPath*

- Il est possible de naviguer dans la structure du document XML source, sélectionner un élément et obtenir l'expression *XPath* correspondante
- Cet assistant est utile pour la sélection des éléments de haut-niveau et les associations de colonnes

# Assistant XPath





# Assistant pour les colonnes

**Column Mapping**

Select one of the predefined XPath for the current column, or please enter another XPath expression.

Column Name:

Data Type:

**XPath Selection**

- ☒ XML elements at fixed absolute path
- ☐ XML elements named "category" at any location
- ☐ Custom XPath Expression

? OK Cancel



# Assistant XML Data Set

**New XML Data Set**

**Column Mapping**

Create data set columns by selecting and adding items from the XML structure.  
Each added XML element or attribute is mapped to a data set column.

**XML Structure**

- library
  - book
    - @category
    - title
    - author
      - @name
      - @country
    - year
    - audio

**Column Mapping**

Name	XPath	Type
category	/@category	String
title	/title	String
name	/author/@name	String
year	/year	String

Buttons: Add, Edit, Remove, Up, Down, Show Sample Data

Navigation: < Back, Next >, Finish, Cancel



# Jeu de données d'un service web

---

Les étapes d'obtention des données dans le cas d'un service Web sont :

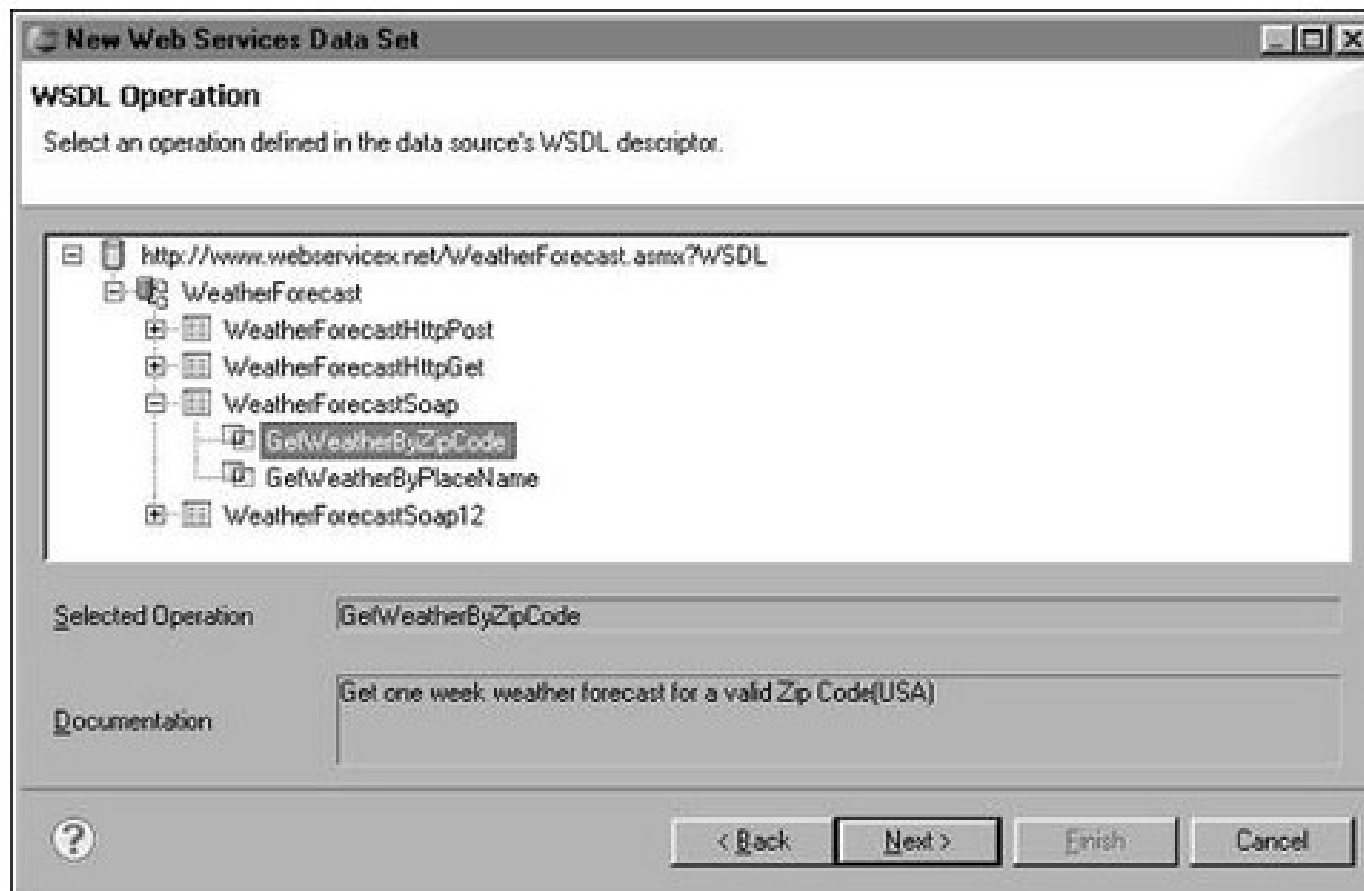
- L'envoi d'une requête SOAP identifiant le service, la fonction à appeler, les paramètres
- Le parsing XML de la réponse SOAP
- L'extraction des données voulues de la réponse

L'assistant permet donc d'effectuer la requête SOAP et de naviguer dans la réponse XML afin d'extraire les données via *XPath*

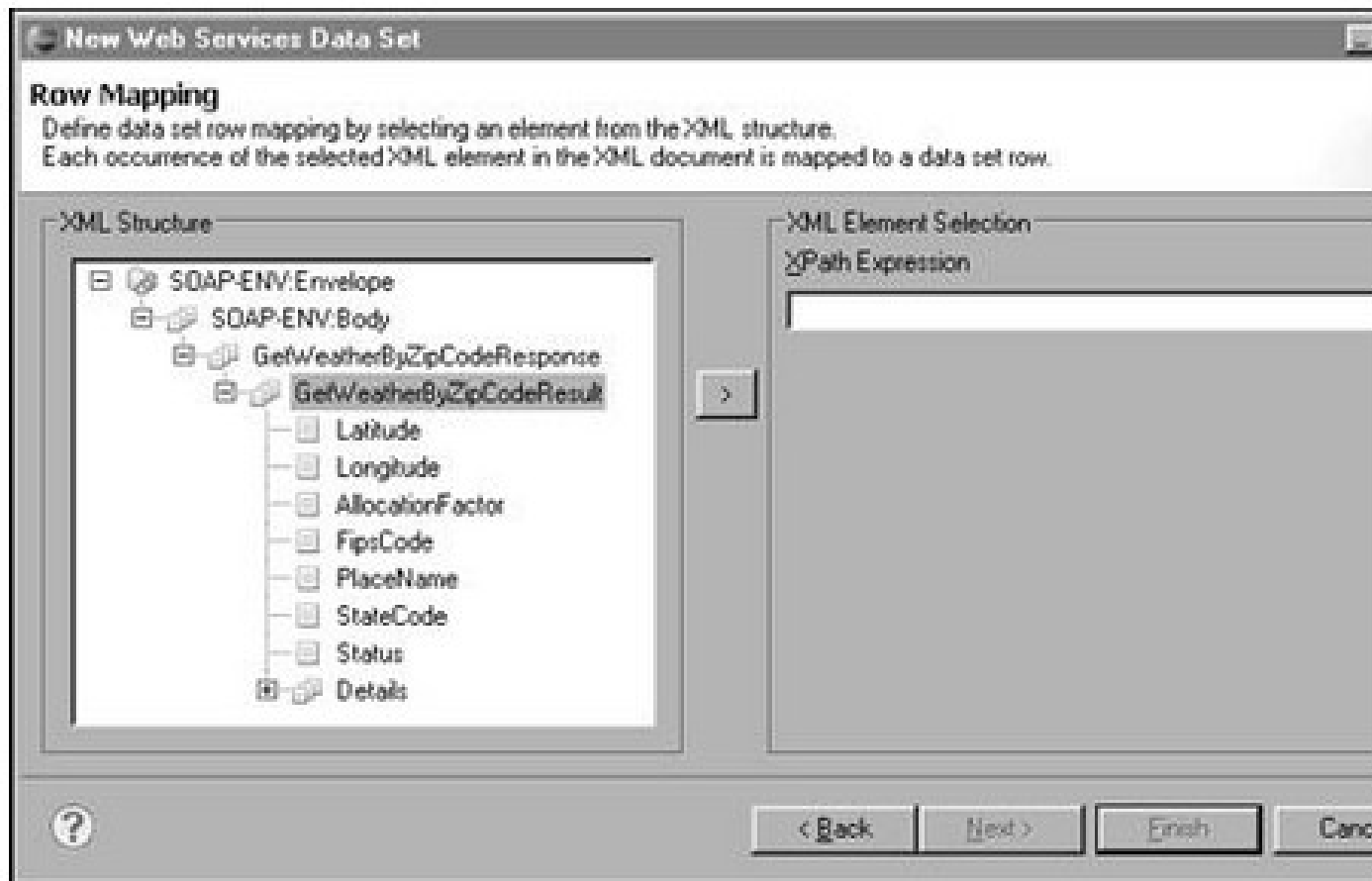




# Sélection de l'opération



# Assistant XPath





# Jeu de données liés

---

Il est possible de combiner les données de 2 jeux distincts avec les **jeux de données liés**

- Les jeux liés ne sont pas nécessairement de même type.  
Par exemple, combiner JDBC et XML



# Jointure

---

Lier 2 jeux de données est similaire à lier 2 tables d'une base de données en précisant la colonne devant correspondre.

Il y a cependant 2 limitations par rapport à SQL :

- Une seule jointure par jeux de données est permise
- Les types de jointure supportés sont : *inner*, *left outer*, *right outer*, *full outer*

Il est cependant possible de lier un jeu de données liés à un autre jeu de données



# Rappel sur les jointures

---

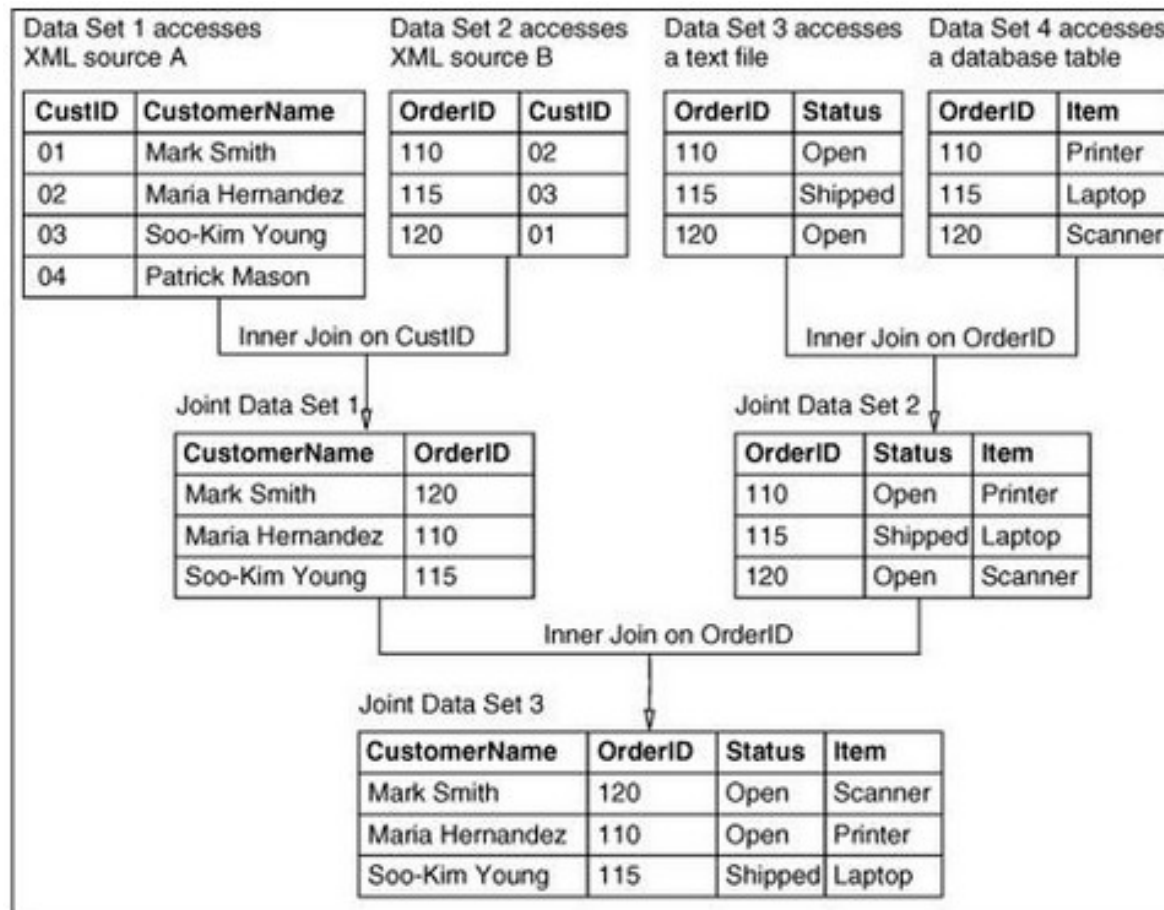
***Inner*** : retourne les lignes des deux jeux de données lorsque la clé correspond

***Left outer join*** : retourne toutes les lignes du premier jeu de données même si il n'y a pas de correspondance dans le second jeu

***Right outer join*** : retourne toutes les lignes du second jeu de données même si il n'y a pas de correspondance dans le premier jeu

***Full Outer*** : retourne toutes les lignes du second jeu de données même si il n'y a pas de correspondance

# Example



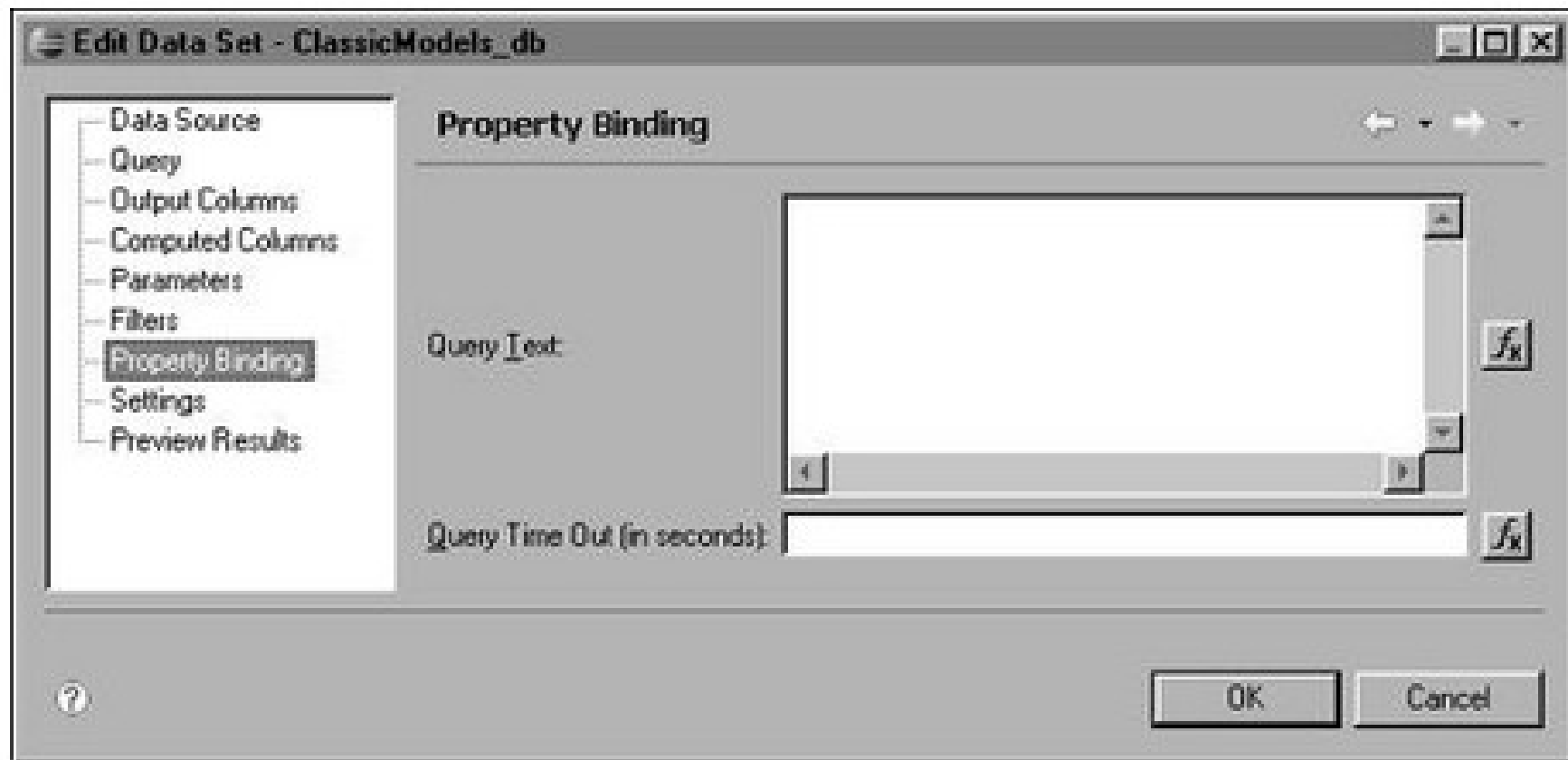


# Spécification du jeux de données à l'exécution

---

- Quelquefois, il est nécessaire de choisir le jeu de données en fonction de critères disponibles à l'exécution
- L'éditeur de jeu de données permet de définir des association de propriétés (property binding)
- En particulier, la propriété *Query Text* permet de spécifier une expression Javascript retournant une requête SQL valide

# Association de propriété







# Propriété *Query Text*

---

BIRT stocke la requête (SQL, Xpath, SOAP ou autre) dans une propriété *Query Text* (visible quand on consulte le XML)

L'expression indiquée lors de la définition d'un jeu de données dynamiques doit respecter la syntaxe correspondant au type de source de données



# Exemple d'expression

---

```
"select customername, customernumber  
from customers where  
country='Australia'"
```

```
"select Date, Open, High from  
StockHistory.csv"
```

```
"select PRODUCTNAME, QUANTITYSTOCK ,  
MSRP from " + params["pTextFile"]
```



# Colonnes et alias

---

- ♦ Les jeux de données permettent de ramener les colonnes de la table
- ♦ Il est possible, et recommandé, de donner des alias et des noms d'affichage aux colonnes si ceux de la base de données ne sont pas explicites.
- ♦ Ce sont ces noms qui sont manipulés dans le rapport



# Colonne calculée

---

Il est possible d'ajouter des **colonnes calculées** au jeu de données.

Les colonnes calculées sont des colonnes dont la valeur est une expression Javascript

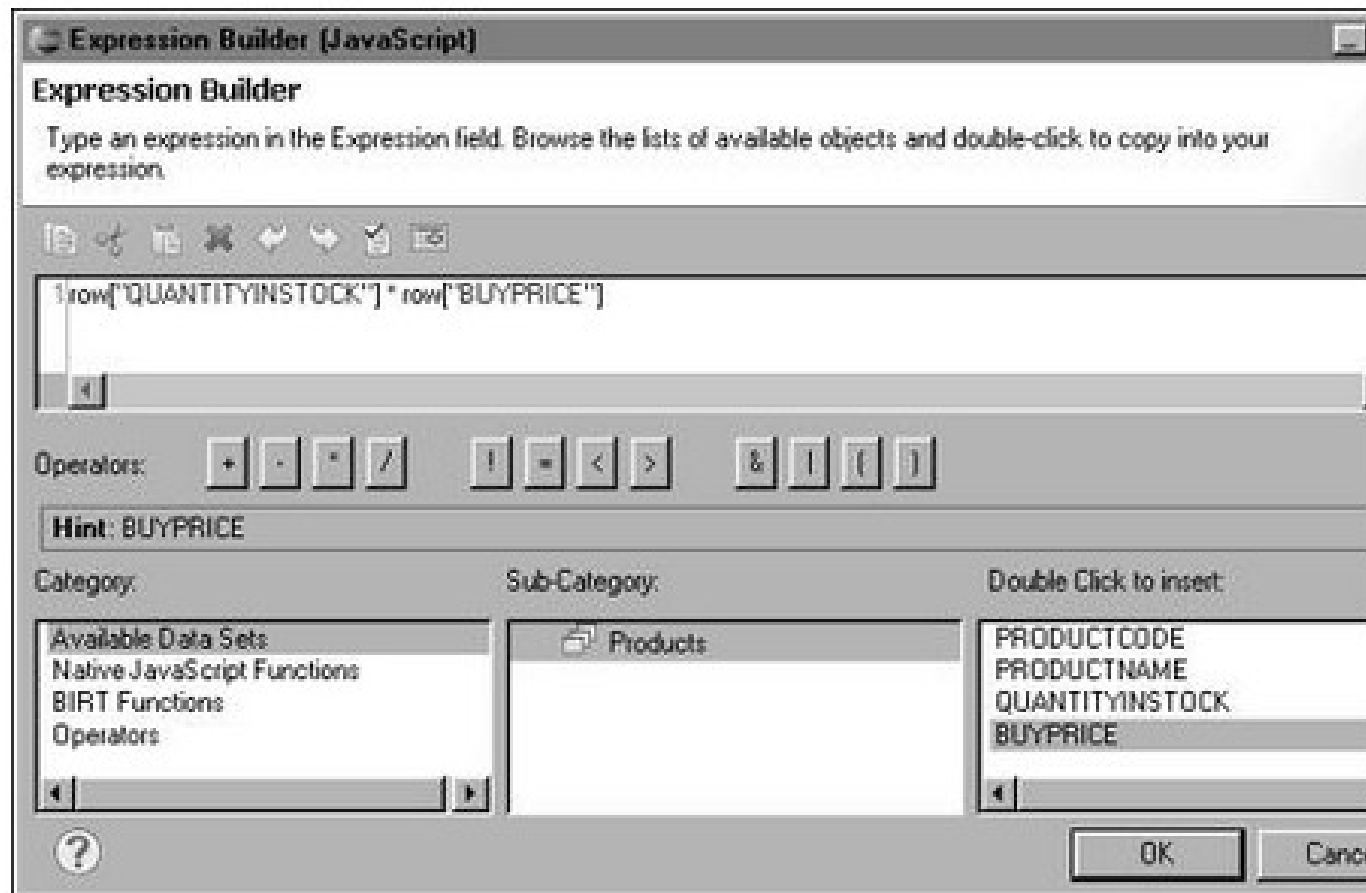
- Par exemple, il est possible de définir une colonne comme étant le produit de 2 autres :

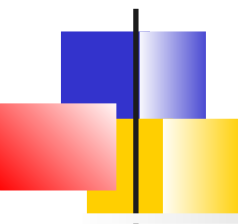
```
dataSetRow["pricequote"]*dataSetRow["quantity"]
```

Les colonnes calculées peuvent également des résultats d'agrégation

BIRT propose un assistant pour la mise au point des expressions : l'expression builder

# Expression Builder





# Liaisons de données (data binding)



# Introduction

---

Le jeu de données fournit les données utilisables dans le rapport. Cependant, avant de pouvoir les afficher dans un élément, il est nécessaire de créer une **liaison (binding)**

Une liaison définit :

- une **expression** indiquant la donnée dynamique à afficher
- Un **nom** qui est utilisé dans le rapport

Les liaisons de données sont proches de la notion de variable et peuvent permettre de réutiliser facilement dans le rapport un calcul, une expression.

Une liaison de données est toujours liée à un élément du rapport qui définit alors son contexte d'utilisation : l'élément lui-même + les éléments contenu dans cet élément



# Liaison automatique

---

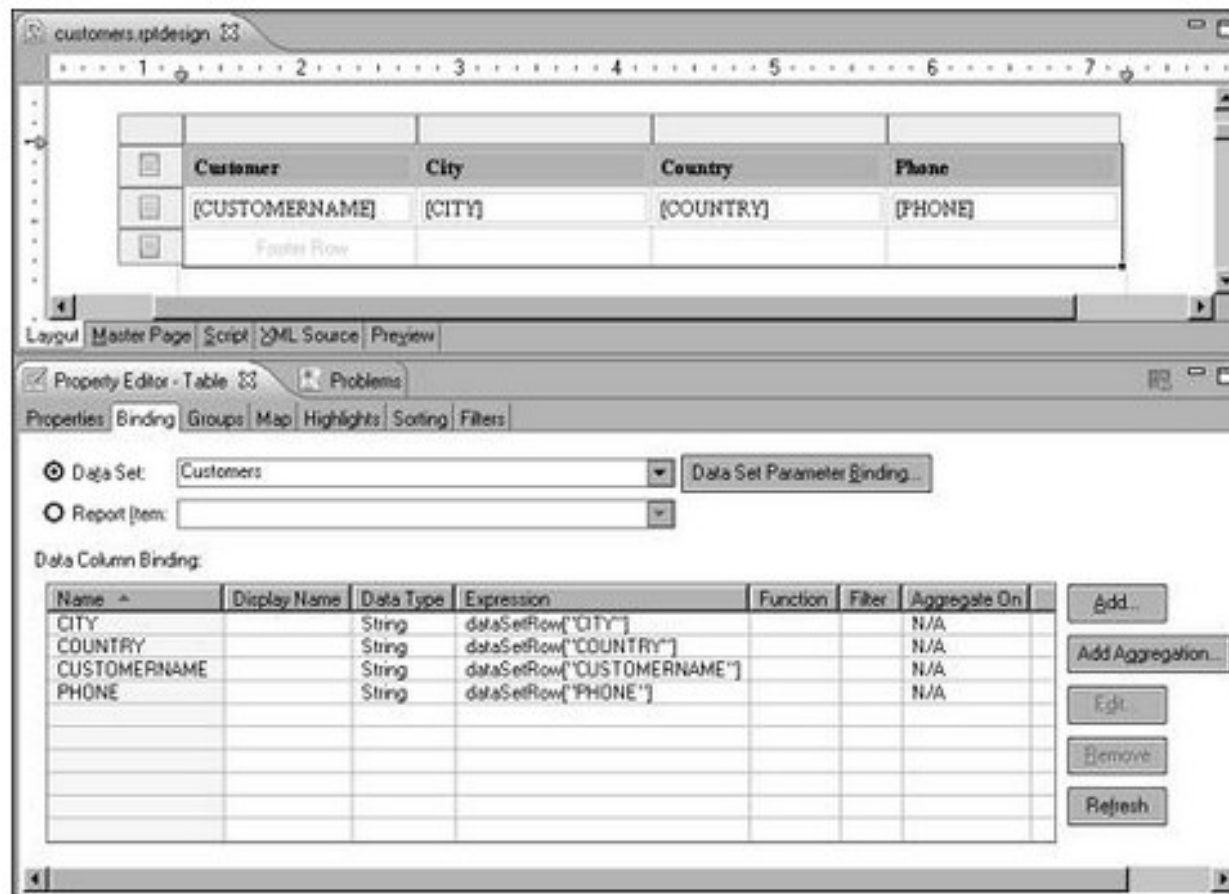
En glissant/déposant le premier champ d'un jeu de données dans une table, BIRT Designer effectue plusieurs opérations :

- Il associe le jeu de données à la table
- Crée une liaison de données reprenant le nom de la colonne
- Crée un élément utilisant la liaison pour afficher la donnée

Ces informations sont alors visibles dans l'onglet « **Binding** » de l'éditeur de propriété



# Exemple Table





# Liaison du jeu de données

---

La liaison de données créée automatiquement par BIRT

- A pour nom le nom du champ
- A pour expression :  
***dataSetRow["NomDuChamp"]***
- Dans la vue design, la liaison est indiquée par :  
***[NomDeLaLiaison]***



# Autres liaisons

---

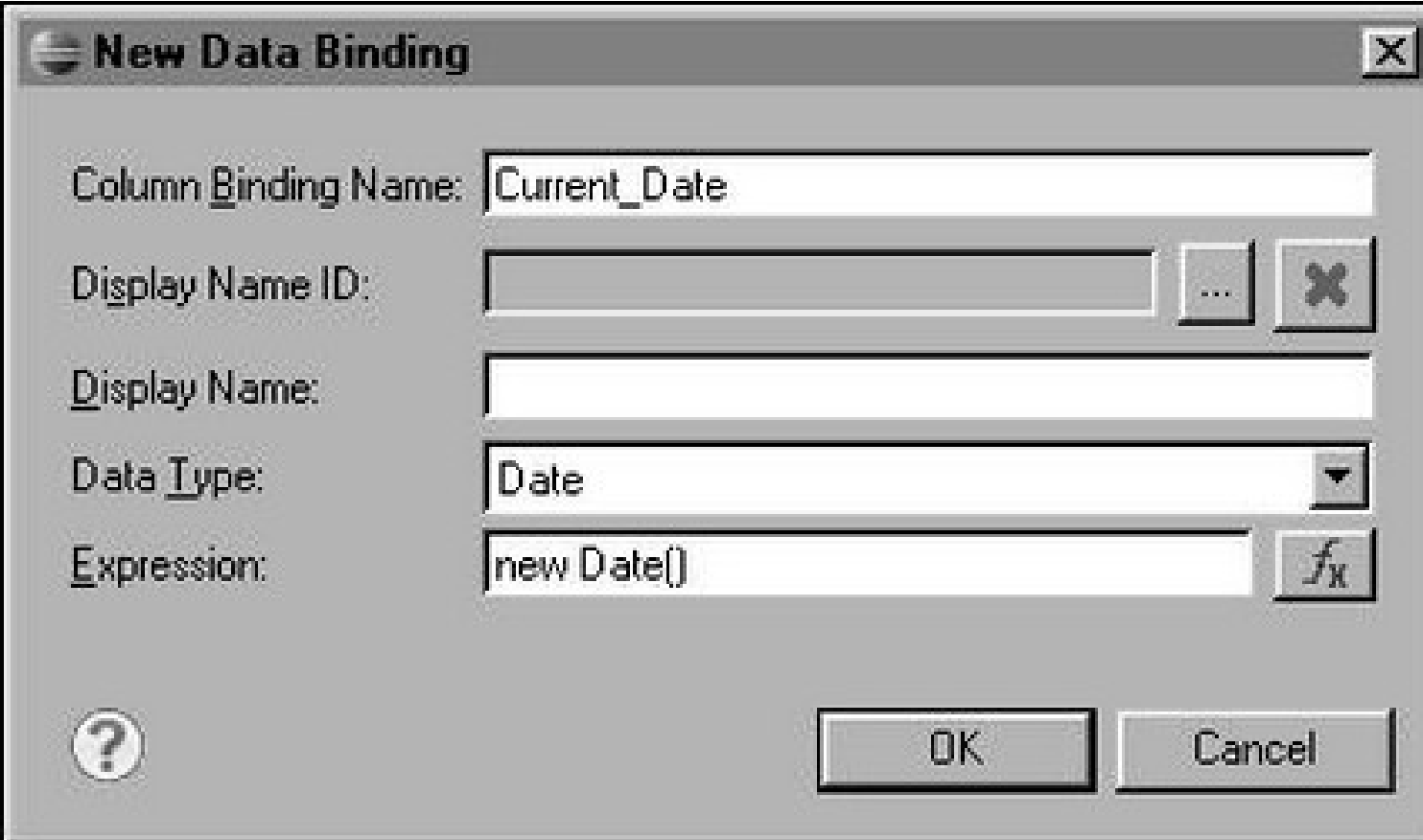
Les liaisons ne concernent pas nécessairement le jeu de données.

Une liaison peut être n'importe quelle expression supportée par BIRT (Extension de Javascript)

Les liaisons définies peuvent également servir à définir d'autres liaisons.

- Elles sont alors accessible par la syntaxe :  
***row["NomDeLaLiaison"]***

# Création de liaison



The image shows a 'New Data Binding' dialog box with the following fields and controls:

- Column Binding Name:** A text box containing 'Current\_Date'.
- Display Name ID:** A text box that is empty, followed by a button with three dots and a button with an 'X'.
- Display Name:** An empty text box.
- Data Type:** A dropdown menu showing 'Date'.
- Expression:** A text box containing 'new Date()', followed by a button with the formula symbol 'fx'.
- Buttons:** A question mark icon, an 'OK' button, and a 'Cancel' button.



# Contexte d'utilisation d'une liaison

---

Une liaison est toujours associée à un élément du rapport.

La liaison est alors utilisable dans l'élément **et** dans tous les sous-éléments contenus.

=> Par exemple, toutes les liaisons d'une table sont utilisables dans tous les éléments utilisés à l'intérieur de la table.

=> Pour des soucis de clarté et de maintenabilité, il est recommandé de ne définir des liaisons qu'au niveau des containers de haut-niveau



# Édition de liaison

---

Une fois créé, il est dangereux de changer le nom de la liaison

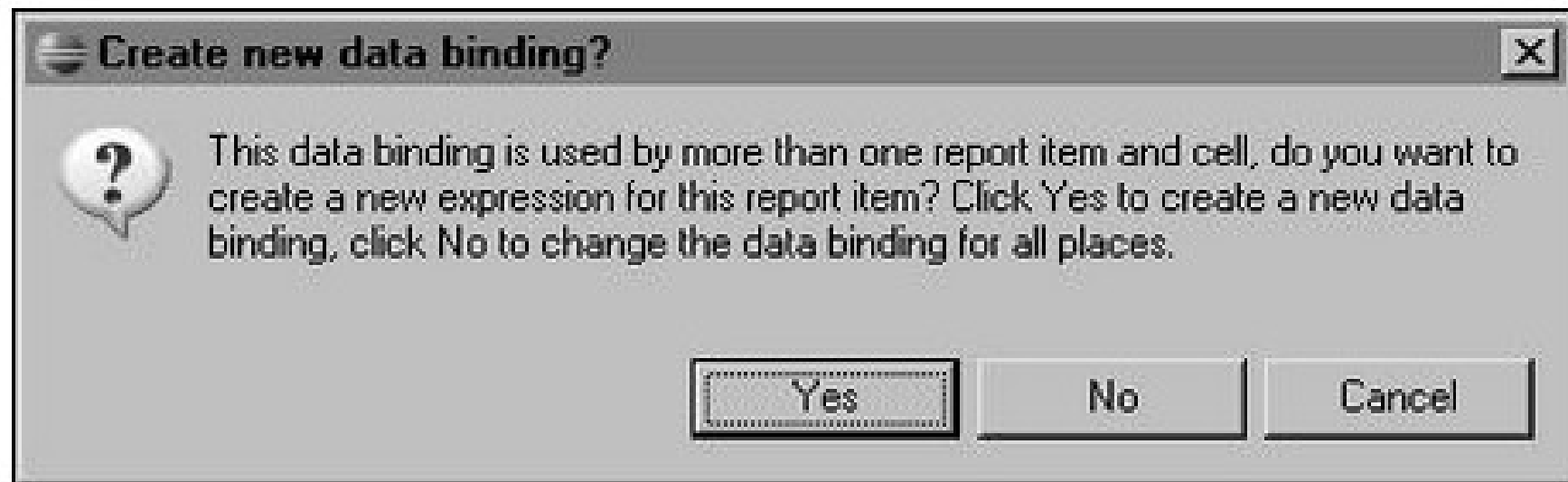
- Car cette liaison peut être utilisée dans différents endroits du rapport et il n'est pas facile d'avoir une vue globale de tous ces endroits.

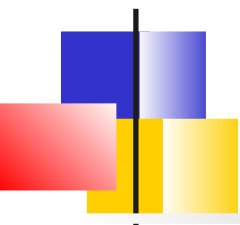
Cependant au moment de la modification ou de la suppression d'une liaison, BIRT avertit si d'autres éléments sont affectés



# Mise à jour de liaison

---





# Composition de rapport

---

Structure d'un rapport et types  
d'éléments

Les conteneurs

Les éléments simples

Formatage des éléments

Page maître et sauts de page





# Structure d'un rapport

---

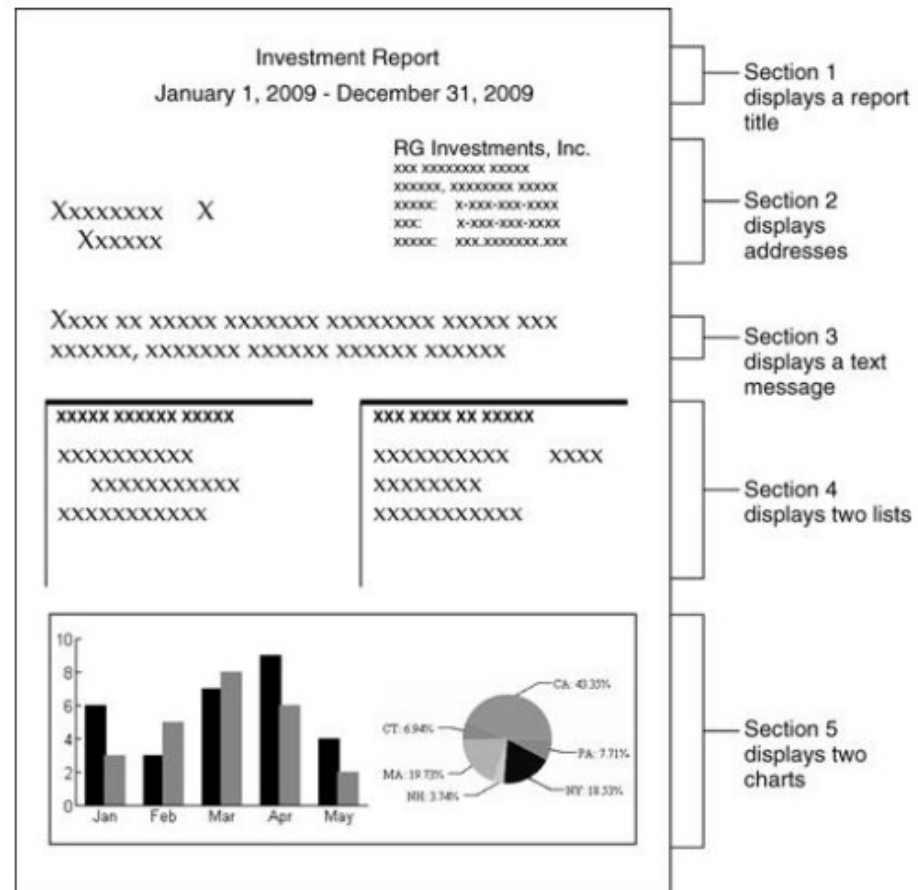
Il est utile de voir un rapport comme une séquence de blocs de contenu horizontaux : les **sections**

Une section correspond généralement à un container de haut-niveau

La division d'un rapport en section permet :

- Utiliser un jeu de données différents par section
- Formater et dimensionner les sections de façon indépendante
- Spécifier des sauts de page entre les sections
- Conditionner l'affichage d'une section

# Example sections





# Éléments d'un rapport

---

La mise au point d'un rapport consiste à faire glisser/déposer des éléments de la palette sur la page.

On peut distinguer 2 types d'éléments :

- Les **éléments affichant** des données
- Les **éléments permettant d'organiser** plusieurs éléments dans une section



# Méthodologie

---

La méthodologie de mise au point d'un rapport consiste à :

- Identifier les sections
- Pour chaque section, :
  - Soit ajouter un élément unique
  - Soit ajouter un conteneur organisant des éléments
- Pour les sections contenant des conteneurs, ajouter des autres conteneurs ou des éléments d'affichage



# Éléments disponibles

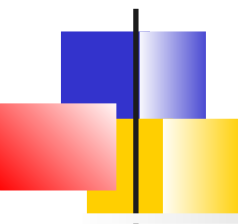
---

## Containers

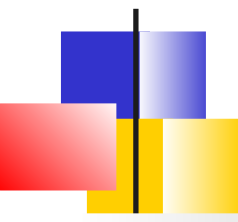
- **Grid** : Organise plusieurs éléments dans une table. Le nombre de lignes est fixe
- **Table** : Organise les données d'un jeu de données dans une table. Le nombre de lignes correspond aux nombres de ligne dans le jeu de données
- **List** : Organise les données d'un jeu de données dans différents layouts

## Éléments

- **Label** : Affichage d'un texte statique
- **Text** : Affichage d'un texte dynamique contenant éventuellement du formatage HTML
- **Dynamic Text** : Données CLOB
- **Data** : Affichage d'une valeur calculée ou d'un champ du jeu de données
- **Image** : Affichage d'une image
- **Chart** : Affichage du jeu de données sous forme de graphique
- **Crosstab** : Affiche des données agrégées dans un tableau



# Les conteneurs



# Grille

Une grille est similaire à une table HTML. Elle automatise l'alignement de blocs de contenu

Au placement d'une grille, l'éditeur affiche une structure lignes/colonnes dont chaque cellule a par défaut les mêmes dimensions

- Les cellules peuvent être ensuite redimensionnés et formatées individuellement
- A tout moment, on peut ajouter/supprimer une ligne ou une colonne
- Les lignes vides ont des effets différents en fonction du format.
  - Par défaut, les lignes vides n'apparaissent pas dans les formats HTML et PDF. Pour créer une ligne vide dans ces formats, il suffit de préciser la hauteur de la ligne
  - Dans les format DOC et XLS, les lignes vides apparaissent



# Table

Une table comme la grille automatise l'alignement des éléments. La table itère sur toutes les lignes d'un jeu de données et ne peut être associée qu'à un seul jeu de données

Une table contient 3 types de lignes dans lesquels sont placés les éléments du rapport :

- **L'entête** (header) : La première ligne d'affichage répétée par défaut sur chaque page occupée par la table
- **Le détail** : Les données dynamiques de la table répétée autant de fois qu'il y a de lignes dans le jeu de données
- **Le bas** de table (footer) : Éléments s'affichant une seule fois en bas du tableau

Une table peut contenir plusieurs lignes du même type





# Liste

---

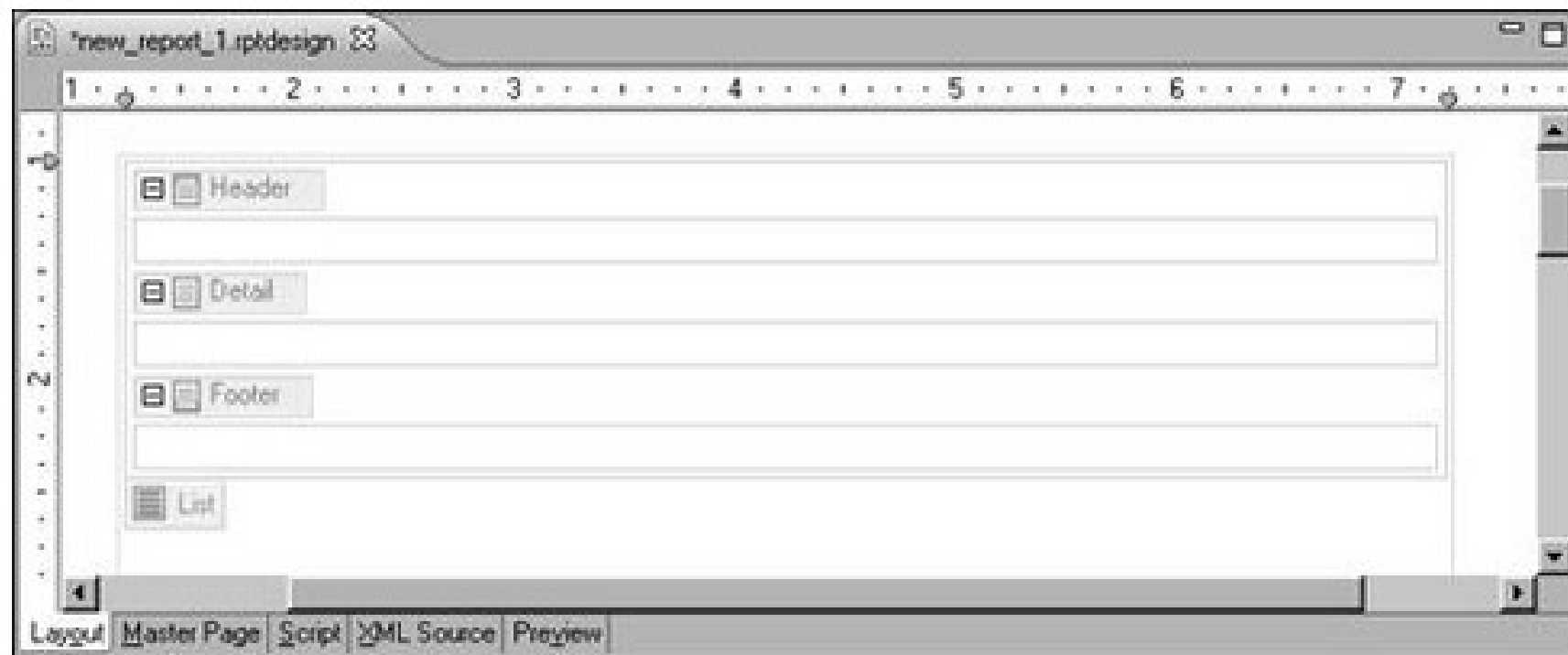
Les listes sont utiles pour organiser des données dans toutes les structures différentes d'une table.

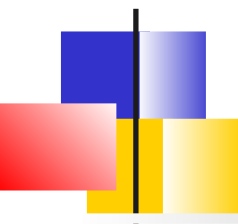
Comme une table, une liste itère sur les lignes du jeu de données

La structure d'une liste différencie :

- **L'entête** : Apparaît une seule fois au début de la section
- **Le détail** : Les données dynamiques apparaissant autant de fois qu'il y a de lignes dans le jeu de données
- **Le bas** de liste : Apparaît une seule fois à la fin de la section

# Structure d'une liste





# Les éléments simples



# Images

---

- BIRT supporte les formats BMP, GIF, ICO, JFIF, JPE, JPEG, JPG, PNG, TIF, TIFF et SVG
- 4 possibilités pour l'insertion d'une image :
  - Spécifier l'emplacement du fichier (<http://> ou <file:///>)
  - Utiliser le dossier ressource
  - Embarquer l'image
  - Indiquer un champ du jeu de données



# Propriétés de l'image

---

L'élément image affiche l'image à sa taille réelle.

Elle peut être redimensionnée en précisant les dimensions dans la fenêtre de propriétés

Un lien hypertexte et un texte alternatif peuvent également être précisés



# Textes

---

Les textes d'un rapport peuvent être :

- Un texte statique dont la valeur ne change pas et est connue au moment du design
- Une chaîne de caractères, nombre ou date provenant du jeu de données
- Une chaîne de caractères, nombre ou date provenant d'expression Javascript

Les textes peuvent être de n'importe quel longueur et BIRT adapte la taille de l'élément au contenu



# Types d'éléments

---

Il y a 4 types d'éléments texte :

- **Data** : Affiche des valeurs dynamiques du jeu de données, de champs calculés ou d'expression Javascript. Par défaut sur une seule ligne
- **Dynamic Text** : Champ CLOB du jeu de données pouvant contenir du formatage HTML
- **Label** : Texte statique court
- **Text** : Texte multilignes, Texte HTML, Texte statique + champ dynamique, Contenu interactif spécifié par du code contenu dans une balise `<script>`



# Cas d'usage

---

- ✓ Formatage différent dans un texte statique : *Text* (le formatage d'un label ou d'un data s'applique globalement)
- ✓ Texte statique + données dynamique : *Text* ou *Data*
- ✓ Données dynamiques HTML : *Text* ou *Dynamic Text*
- ✓ Données dynamiques HTML + texte statique : *Text*
- ✓ Données dynamiques fournies par Javascript : *Text* et *Data*





# Text

---

Pour un élément Text, un éditeur spécialisé est disponible. Il permet de :

- Sélectionner le type de texte (Auto, Plain, HTML)
  - Le type Auto et plain ne permettent pas de formatage ni de données dynamiques
  - L'éditeur HTML permet de rajouter des balises HTML et des références à du css
- Il permet également la balise `<VALUE-OF>` permettant de délimiter une expression javascript (code ou accès aux liaisons de données) et d'indiquer le formatage à appliquer



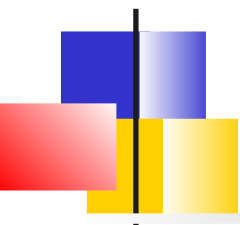
# Exemple

---

Cher <VALUE-OF>**row["genre"] == "M" ?**  
**"Monsieur" : "Madame"**</VALUE-OF>

**<span style="font-size: larger">**Merci pour  
votre commande <VALUE-  
OF>row["OrderId"]</VALUE-OF></span>

<VALUE-OF **format="dd/MM/yy"**>new  
Date( )</VALUE-OF>



# Formatage des éléments



# Formatage Birt

---

BIRT fournit de nombreuses propriétés (basées sur les propriétés CSS ) pour formater les éléments d'un rapport (alignement, couleur, police, ...)

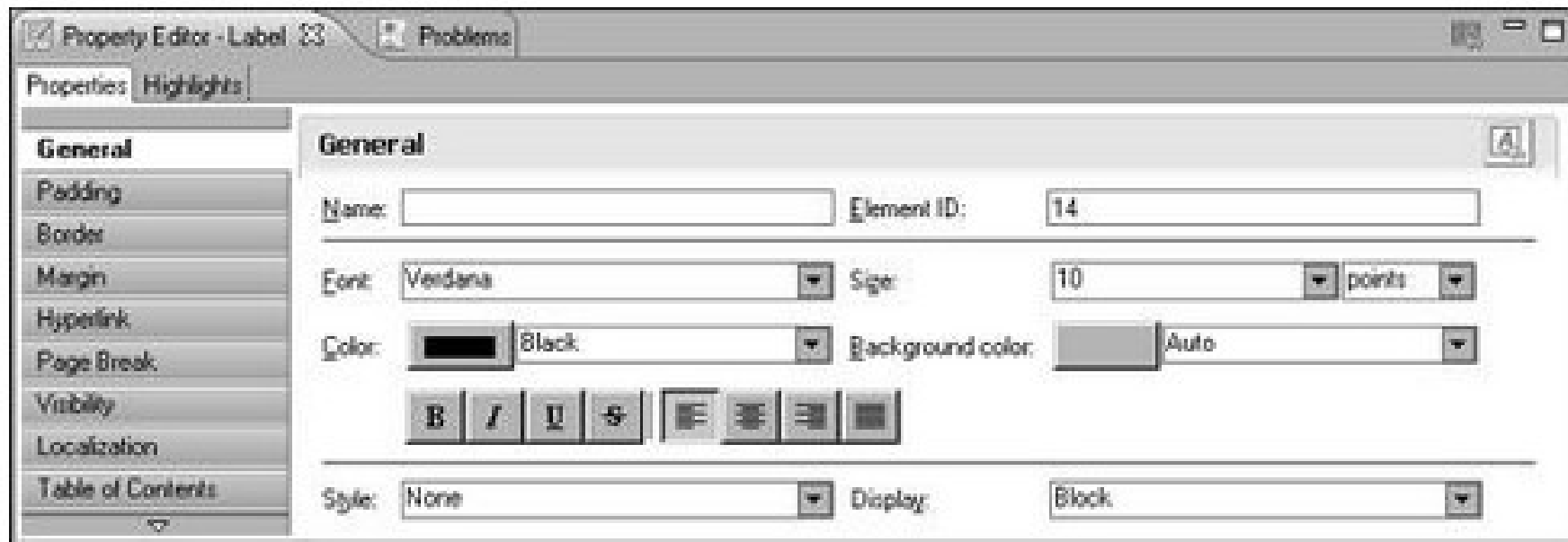
BIRT permet le copier/coller de format

Les options de formatage sont disponibles dans 2 vues :

- L'éditeur de propriété pour les propriétés les plus communément utilisées
- La vue Propriétés pour les propriétés plus complexes

*Windows → Show View → Properties*

# Vues pour le formatage





# Propriétés communes

---

Le formatage les plus communs sont :

- Formatage d'une valeur date, nombre ou chaîne de caractères. BIRT fournit les styles communs d'affichage
- Le type de police, la taille et la couleur : Attention la police doit être présente sur le poste de l'utilisateur visualisant le rapport
- Le style de texte (italique, gras, ...)
- L'alignement
- Les bordures, marges et padding
- Les sauts de page



# Formatage des nombres

---

BIRT propose 6 formats communs et un champ locale pour formater les nombres

Il est possible de spécifier son propre format en utilisant les symboles définis par la classe Java *DecimalFormat*



# Examples

---

Format pattern	Data in the data set	Result
0000.00	12.5	0012.50
	124.5	0124.50
	1240.553	1240.55
#.000	100	100.000
	100.25	100.250
	100.2567	100.257
\$#,###	2000.00	\$2,000
	20000.00	\$20,000
ID #	15	ID 15





# Formatage de date

---

BIRT propose plusieurs formats prédéfinis et permet de spécifier la locale

On peut fournir son propre format en utilisant les symboles définies par la classe Java *SimpleDateFormat*

Dans ce cas cependant, la locale n'est pas prise en compte



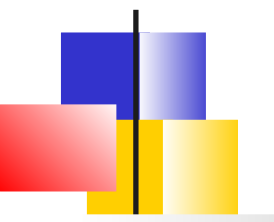
# Formatage des chaînes de caractères

---

BIRT propose 2 formatage de *String* (minuscule, majuscule)

On peut fournir son propre format en utilisant les symboles suivants :

- > : Passage en majuscule
- < : Passage en minuscule
- @ : Représente un caractère. Si la chaîne a moins de caractères, des blancs sont générés
- & : Représente un caractère. Si la chaîne a moins de caractères, pas de blancs
- ! : Inverse le sens de remplissage qui devient de la gauche vers la droite



# Examples

Format pattern	Data in data source	Result
(@#@) @@@-@@@@	6175551007 5551007	(617) 555-1007 ( ) 555-1007
(&&&) &&&-&&&&	6175551007 5551007	(617) 555-1007 ( ) 555-1007
!(@#@) @@@-@@@@	6175551007 5551007	(617) 555-1007 (555) 100-7
!(&&&) &&&-&&&&	6175551007 5551007	(617) 555-1007 (555) 100-7
!(@#@) @@@-@@@@ + ext 9	5551007	(555) 100-7 + ext 9
!(&&&) &&&-&&&& + ext 9	5551007	(555) 100-7 + ext 9
>&&&-&&&&&-&&	D1234567xy	D12-34567-XY
<&&&-&&&&&-&&	D1234567xy	d12-34567-xy



# Styles

---

BIRT supporte les styles permettant d'appliquer le même formatage à différents éléments

Les styles peuvent être définis :

- Dans le rapport
- Dans un fichier css
- Dans un thème d'une librairie BIRT



# Formattage via les styles

---

Il y a 2 façons d'obtenir un formattage via les styles :

- Créer un **nouveau style** puis l'appliquer à un élément
- Appliquer des propriétés de formatage à des **styles prédéfinis** (Ex : *report*, *table-header*)

Liste complète disponible à

[http://www.eclipse.org/birt/phoenix/ref/ROM\\_Styles\\_SPEC.pdf](http://www.eclipse.org/birt/phoenix/ref/ROM_Styles_SPEC.pdf)



# Styles prédéfinis

---

chart,  
crosstab, crosstab-cell, crosstab-column-header, crosstab-detail,  
crosstab-header, crosstab-row-header  
data  
grid  
image  
label, text, text-data  
list, list-detail, list-footer, list-group-footer-<n>, list-group-  
header-<n>, list-header,  
page  
report  
table, table-detail, table-detail-cell, table-footer, table-footer-  
cell, table-group-footer-<n>, table-group-footer-cell, table-group-  
header-<n>, table-group-header-cell, table-header, table-header-cell  
TOC-label-<n>



# Réutilisation de *css*

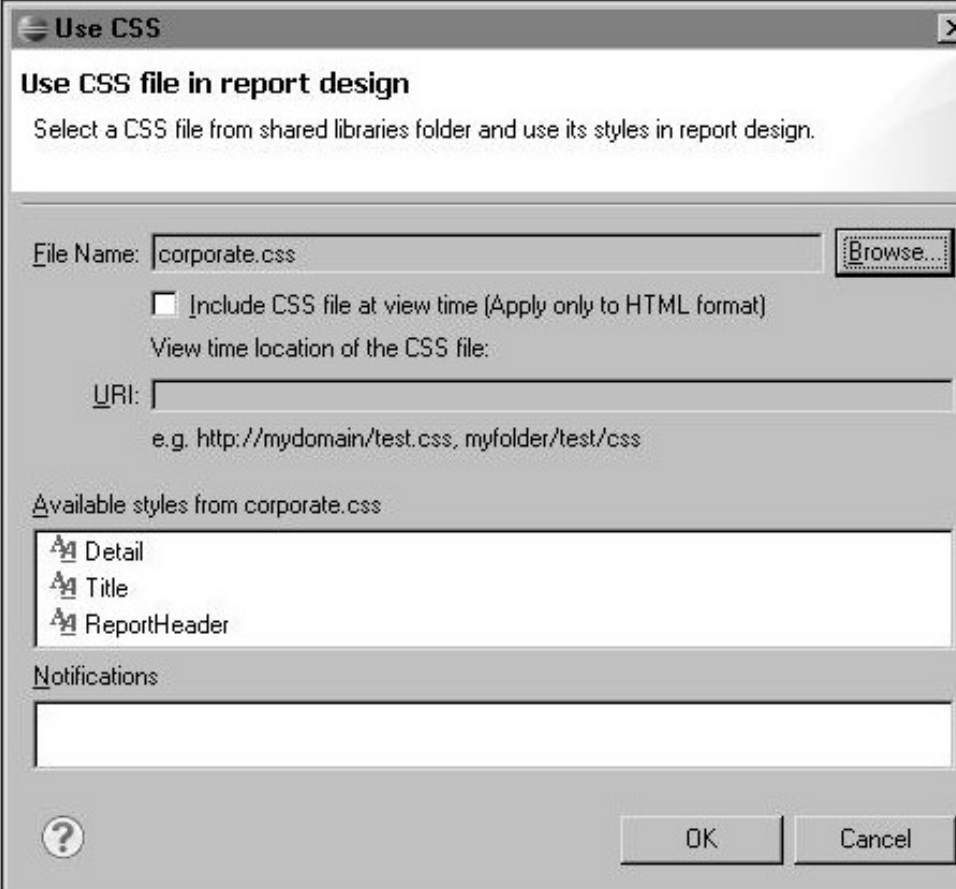
---

La réutilisation de fichier *css* peut se faire :

- Soit en important le fichier
- Soit en liant le rapport au fichier externe présent dans le dossier ressource

BIRT supporte CSS2 mais certaines propriétés ne sont pas supportées

# Réutilisation css



The image shows a 'Use CSS' dialog box with a title bar containing a minimize button, the text 'Use CSS', and a close button. The main area has a title 'Use CSS file in report design' and a subtitle 'Select a CSS file from shared libraries folder and use its styles in report design.' Below this, there is a 'File Name' text box containing 'corporate.css' and a 'Browse...' button. A checkbox labeled 'Include CSS file at view time (Apply only to HTML format)' is unchecked. Below the checkbox is the text 'View time location of the CSS file:' followed by a 'URI:' text box. Below the URI box is an example: 'e.g. http://mydomain/test.css, myfolder/test/css'. A section titled 'Available styles from corporate.css' contains a list box with three items: 'Detail', 'Title', and 'ReportHeader', each preceded by a small icon. Below this is a 'Notifications' section with an empty text box. At the bottom left is a help icon (a question mark in a circle). At the bottom right are 'OK' and 'Cancel' buttons.

**Use CSS**

**Use CSS file in report design**  
Select a CSS file from shared libraries folder and use its styles in report design.

File Name: corporate.css Browse...

☐ Include CSS file at view time (Apply only to HTML format)  
View time location of the CSS file:

URI:   
e.g. http://mydomain/test.css, myfolder/test/css

Available styles from corporate.css

- Detail
- Title
- ReportHeader

Notifications

? OK Cancel





# Styles conditionnels

---

Il est possible d'appliquer un style seulement sous certaines conditions

Il faut alors définir des règles de formatage via l'onglet « **Highlights** » de l'éditeur de propriétés et le formatage associé

L'onglet « *Highlights* » peut être défini au niveau :

- d'un élément
- d'un style




# Ajustement de l'espace

---

Pour ajuster l'espace du contenu, plusieurs techniques peuvent être utilisées :

- Dimensionner les lignes et les colonnes d'une table ou d'une grille
- Insérer une ligne/colonne vide et ajuster sa taille
- Dimensionner les marges et le padding (résultat moins maîtrisé)



# Affichage des données sur une ligne

---

Il est possible d'afficher toutes les données d'un jeu sur une seule ligne en utilisant une table

La technique consiste à utiliser la fonction d'agrégation **CONCATENATE** appliquée au champ que l'on veut afficher

L'élément d'affichage se place alors dans l'entête de la table



# Règle de correspondance

---

Il est possible d'afficher des valeurs alternatives aux données du Jeu

- Par exemple, Afficher « Masculin » si le champ contient la valeur « M »

Les associations se font dans l'onglet **Map**. Il faut alors préciser une condition et le texte de remplacement



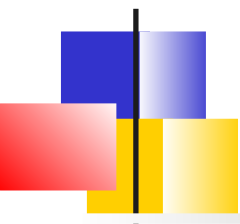
# Cacher conditionnellement des éléments

---

Il est possible de cacher des éléments sous certaines conditions.

La propriété « **Visibility** » permet de spécifier des conditions via une expression booléenne

Elle permet également de cacher l'élément pour des formats particulier



# Page maître et sauts de page



# Page maître

---

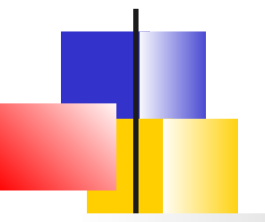
La page maître définit pour les pages du rapport :

- La dimension de la page
- Les marges
- L'entête
- Le bas de page.

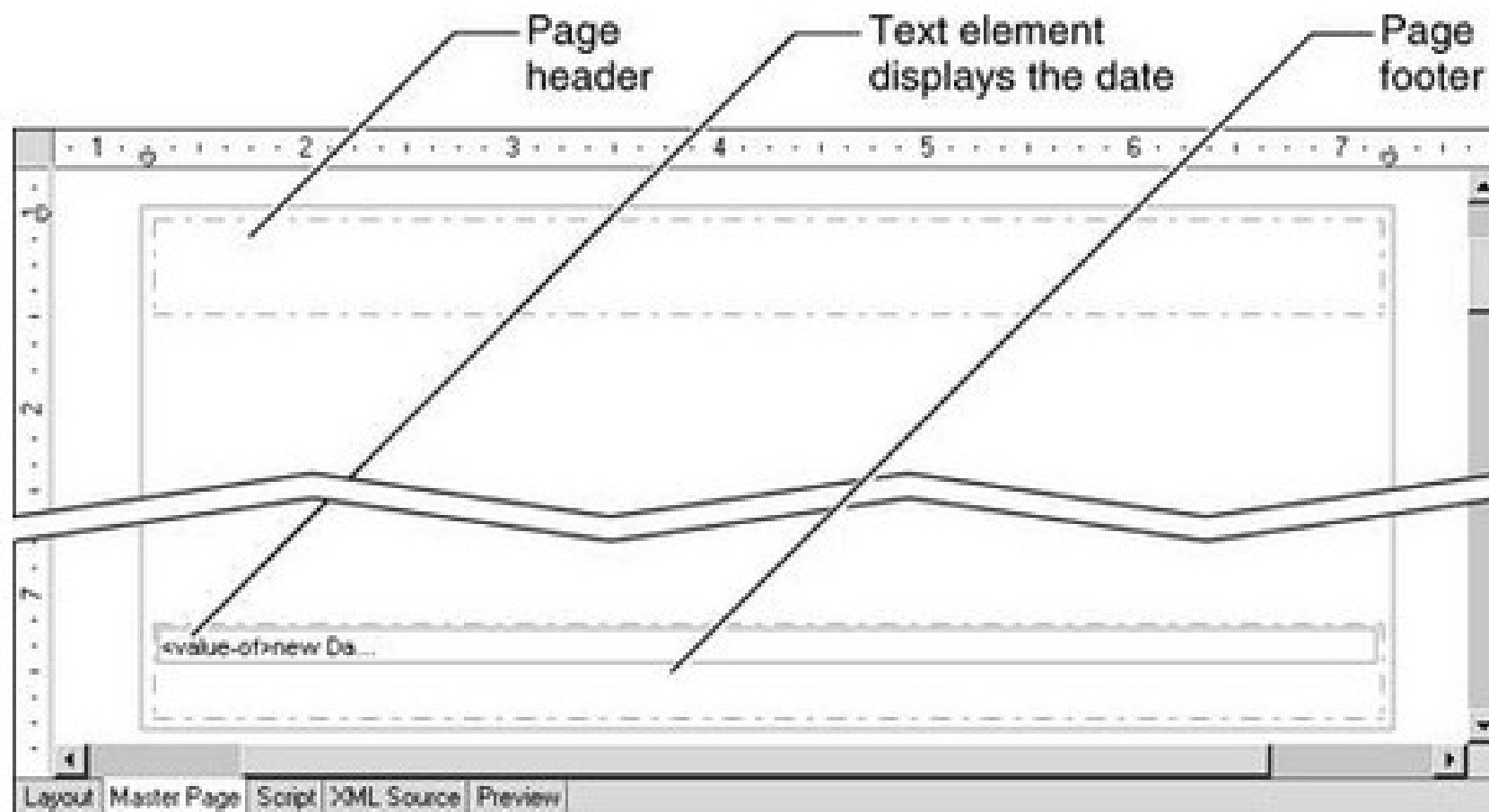
La taille de la page et les marges ne s'applique qu'aux formats PDF, DOC et PPT

Pour les documents HTML 2 dispositions sont disponibles :

- disposition fixe (par défaut), le résultat est identique au PDF
- disposition automatique (auto), la taille du rapport s'adapte à la taille de la fenêtre du navigateur



# Page maître par défaut

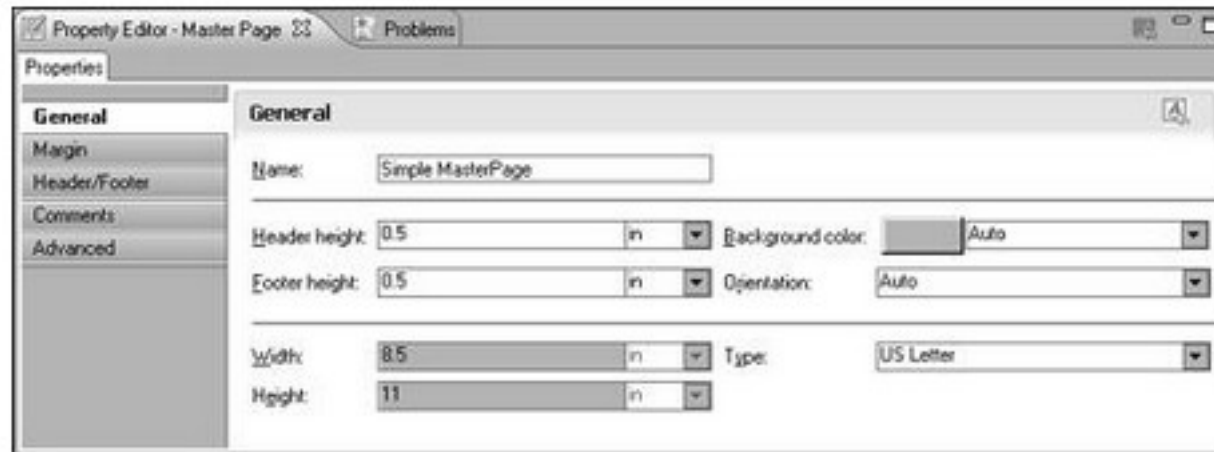






# Dimensions du rapport

Par défaut, le rapport utilise un format US letter, une orientation portrait et des marges de 0,25"





# Personnalisation entête et bas de page

---

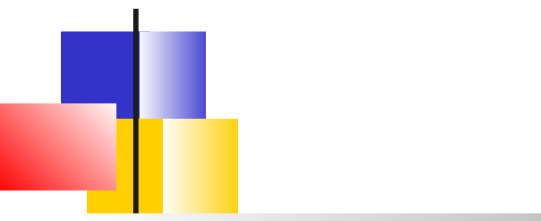
L'entête et le bas des page peuvent être personnalisés en ajoutant des éléments de rapport

- Pour insérer plusieurs éléments, utiliser une grille


BIRT fournit des éléments prédéfinis permettant d'inclure facilement des informations classiques comme le numéro de page, la date, etc.

- Ces éléments sont présents dans la palette lors de l'édition de la page maître

Une option permet d'exclure l'entête sur la première page (propriété « *Show header on first* » )



Use these items to display text that typically appears in the header or footer of a report



# Dimensions de l'entête et bas de page

---

Il est possible d'indiquer une dimension pour l'entête et le bas de page.

- Pour être prise en compte, la hauteur doit être plus grande que l'espace occupé par les éléments inclus dans l'entête ou le bas de page
- Ces dimensions ne sont pas prises en compte pour le format HTML



# Image de fond

---

Il est possible d'afficher une image de fond en filigrane

- Il faut cependant s'assurer que l'image ne perturbe pas la lisibilité du rapport
- Si l'image est plus petite, il est possible de répéter le motif, de fixer sa position

*Property Editor Master page → Properties → Advanced → Background*



# Plusieurs pages maître

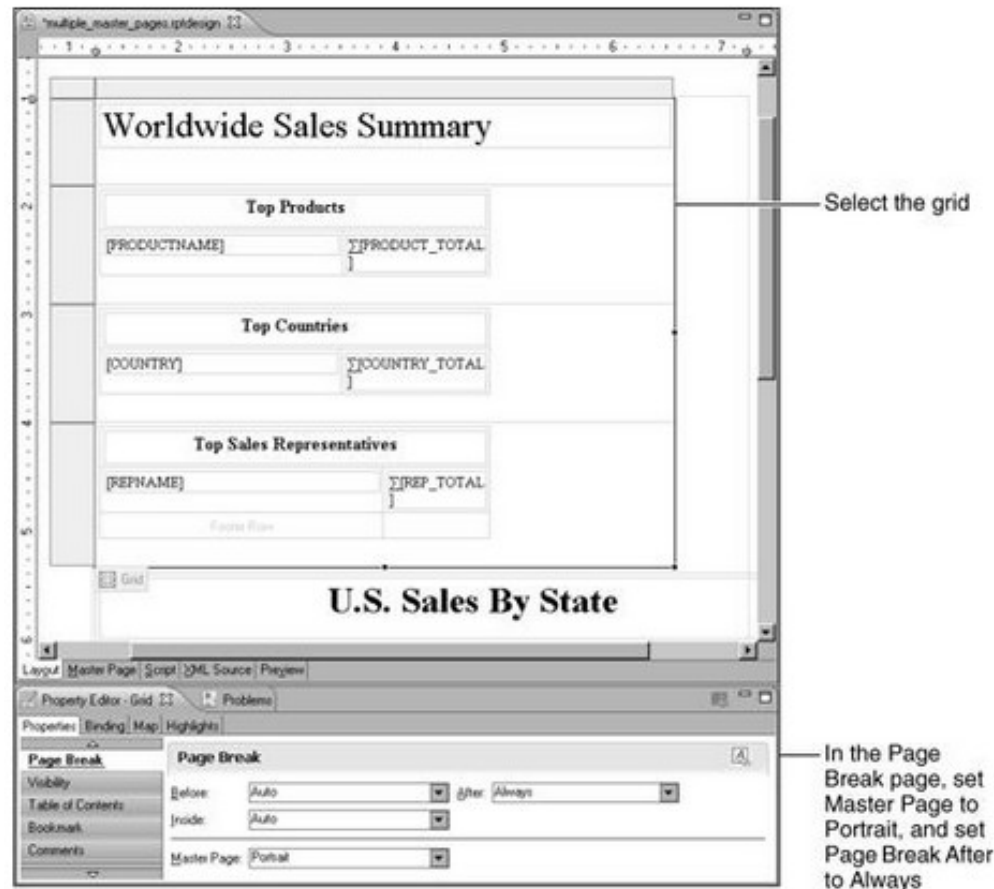
---

Il est possible d'utiliser plusieurs pages maître dans un rapport.

Les étapes de mise en place sont alors :

- Créer toutes les pages maîtres  
nécessaires : *Outline View → Master Pages → Insert Element*
- Insérer des sauts de pages séparant les différentes section du rapport
- Assigner à chaque section une page maître  
*Property Editor → Page Break*

# Sélection de la page maître





# Contrôle de la pagination

---

Par défaut, BIRT remplit les pages avec le maximum de données ; ce qui peut provoquer des sauts de page inopportuns (en plein milieu d'une section)

Il est alors possible de spécifier des sauts de page associés à certains éléments du rapport.

- Les éléments de premier-niveau (qui n'ont pas de conteneur)
- Les éléments de second-niveau (leur conteneur n'a pas de conteneur)
- Les groupes des éléments de premier et second niveaux
- Les lignes des grilles et les détail des tables





# Spécification des sauts de page

---

La spécification des sauts de page s'effectue via les propriétés **Page Break Before** et **Page Break After** et **Inside** qui peuvent prendre les valeurs suivantes :

- **Always** : Toujours insérer un saut
- **Auto** : Insérer un saut si nécessaire (valeur par défaut)
- **Avoid** : Eviter d'insérer un saut (si possible)
- **Always Excluding First** (Groupe et PBBefore) : Insère un saut avant chaque groupe sauf le premier
- **Always Excluding Last** (Groupe et PBAfter) : Insère un saut après chaque groupe sauf le dernier



# Entêtes et sauts de page

---

L'option ***Repeat Header*** présent sur différents éléments (Groupe, tables) est par défaut positionné à *true* provoquant l'affichage de l'entête de l'élément sur toutes les pages occupées

Il est possible de la désactiver



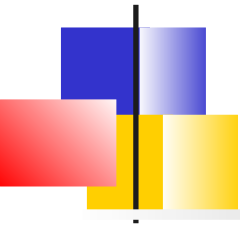
# Nombre de lignes par page

---

Une autre technique pour contrôler la pagination consiste à insérer un saut de page après un certain nombre de lignes de données

L'option ***Page Break Interval*** permet de spécifier ce nombre de lignes.

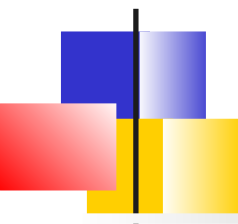
Par défaut, il est de 40.



# Manipulation des données

---

Expressions  
Tri et filtres de données  
Groupes et agrégation  
Paramètres



# Expressions



# Introduction

---

Les expressions sont des expressions Javascript qui retournent **une seule valeur typée**

Les expressions peuvent être multilignes .  
Chaque ligne étant séparée par « ; ». Seule la valeur de la dernière ligne est retournée

Attention,

- Javascript est sensible à la casse
- S'assurer que la valeur retournée est du bon type

BIRT fournit un assistant : l'expression builder accessible via 



# Types de données

---

Les valeurs manipulées par BIRT (champs de la base de données, valeurs d'un élément de rapport ou expression) sont typées

Les types doivent donc correspondre

Les types disponibles sont *Integer*, *Decimal*, *Float*, *String*, *Blob*, *Date Time*, *Date*, *Time*, *Boolean* et *Java Object*



# Expression Builder

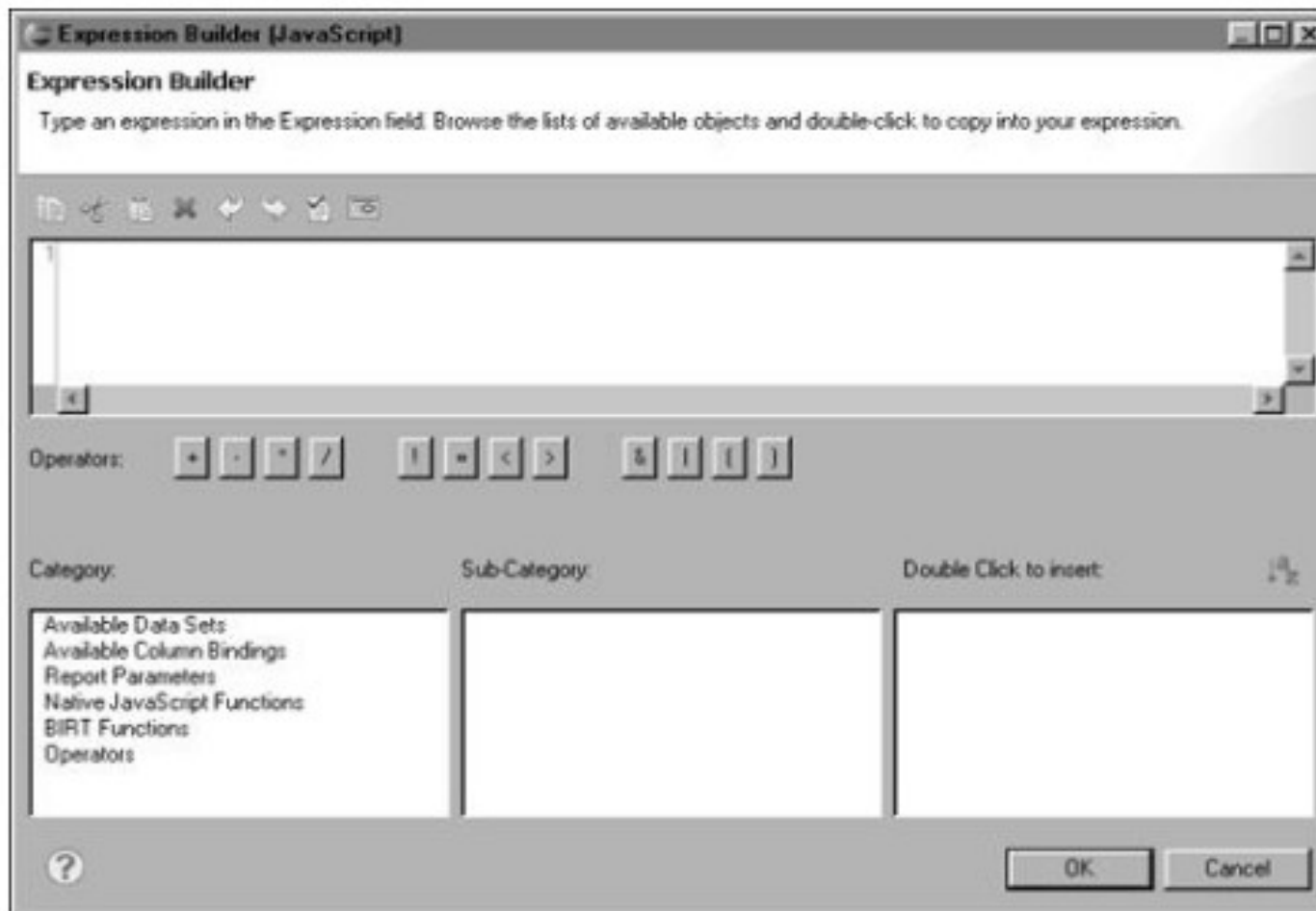
---

L'assistant comporte 3 zones :

- La barre de boutons haute propose entre autre un bouton permettant de valider l'expression finale
- La partie haute est la zone d'édition de l'expression. Les éléments de la partie basse sélectionnés y apparaissent et peuvent être édités
- La partie basse propose une vue hiérarchique des objets utilisable dans les expressions divisé en 3 sous-zones. La sélection d'un objet permet d'afficher les choix possible dans la zone suivante



# Expression Builder





# Objets sélectionnable

---

Les objets sélectionnable dans la partie basse de l'expression builder sont :

- Les **liaisons de données** du jeu associé à l'élément ou à son conteneur
- Les **jeux de données** permettant d'accéder aux liaisons d'un autre jeu de données du rapport
- Les **paramètres** du rapport
- Les **fonctions Javascript** (avec aide contextuelle)
- Les **fonctions BIRT** (l'api Javascript BIRT facilitant certaines opérations)
- Les **opérateurs** javascript
- Les **variables** du rapport défini via le Data Explorer



# Signature des méthodes

---

Le 3ème bloc de la partie basse indique les méthodes disponibles.

Il affiche les signatures de méthodes :

- Types des paramètres d'entrée
- Type du paramètre de sortie

Exemple :

```
year(date:Timestamp):Integer
```



# Manipulation de nombres

---

Les opérateurs classiques  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $($ ,  $)$  peuvent être utilisés.

D'autre part, Javascript et l'API BIRT fournissent de nombreuses fonction pour manipuler les nombres voir :

- Javascript : Number, Math
- BIRT : BirtMath, Finance

Un nombre peut être converti en String via la méthode *toString()* ou en ajoutant la chaîne vide ""



# Manipulation de chaînes

---

- ✓ Remplacement de chaîne : `replace()`
- ✓ Concaténation : `+`
- ✓ Suppression d'espace : `trim()`, `trimLeft()`, `trimRight()`
- ✓ Sous-ensemble : `substr()`, `length`, `indexOf()`, `lastIndexOf()`, `charAt()`
- ✓ Reconnaissance de motif : Expressions régulières
- ✓ Conversion en entier : `parseInt()`
- ✓ Conversion en date : `new Date()`
- ✓ Gestion de la casse : `toLowerCase()`, `toUpperCase()`



# Manipulation de dates

---

- ✓ Date du jour : `new Date()`,  
`BirtDateTime.now()`,  
`BirtDateTime.today()`
- ✓ Extraire une partie d'une date : `getDay()`,  
`getMonth()`, `getYear()`, ...
- ✓ Calcul du temps entre 2 dates :  
`BirtDateTime.diffDay()`, `diffHour()`, ...
- ✓ Calcul de date : `BirtDateTime.addDay()`,  
`addMonth()`, ...



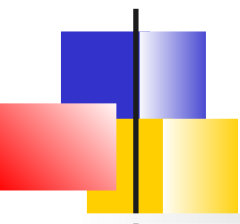
# Manipulation de booléens

---

Une expression booléenne retourne *true* ou *false*

Elles sont surtout utilisées pour afficher conditionnellement un élément, filtrer des données, etc.

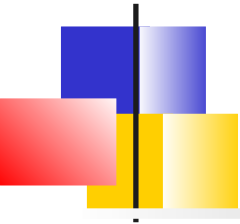
- Les expressions booléennes peuvent être combinées via les opérateurs `&&` et `||`
- La négation est exprimée par `!`
- Les opérateurs de comparaison sont `==`, `>`, `<`



# Tri et filtres de données



# Tri



Le tri des données du jeu peut se faire :

- Soit via la **requête** (Source de données JDBC).  
Méthode recommandée et la plus efficace
- Soit par le **moteur** (quand on ne peut pas faire autrement). La spécification des critères de tri et leur sens de tri s'effectue alors par l'onglet « *Sorting* »



# Spécification du tri BIRT

---

L'onglet « *Sorting* » permet de spécifier plusieurs critères de tri.

Pour chaque critère, il faut spécifier :

- ***Sort key*** : Le critère : soit un champ du jeu de données, soit une expression
- ***Sort direction*** : Ascendant ou descendant
- ***Sort locale*** : Le langage ou auto
- ***Sort strength*** : Indique si la casse, les accents ou la ponctuation sont pris en compte lors du tri



# Force du tri

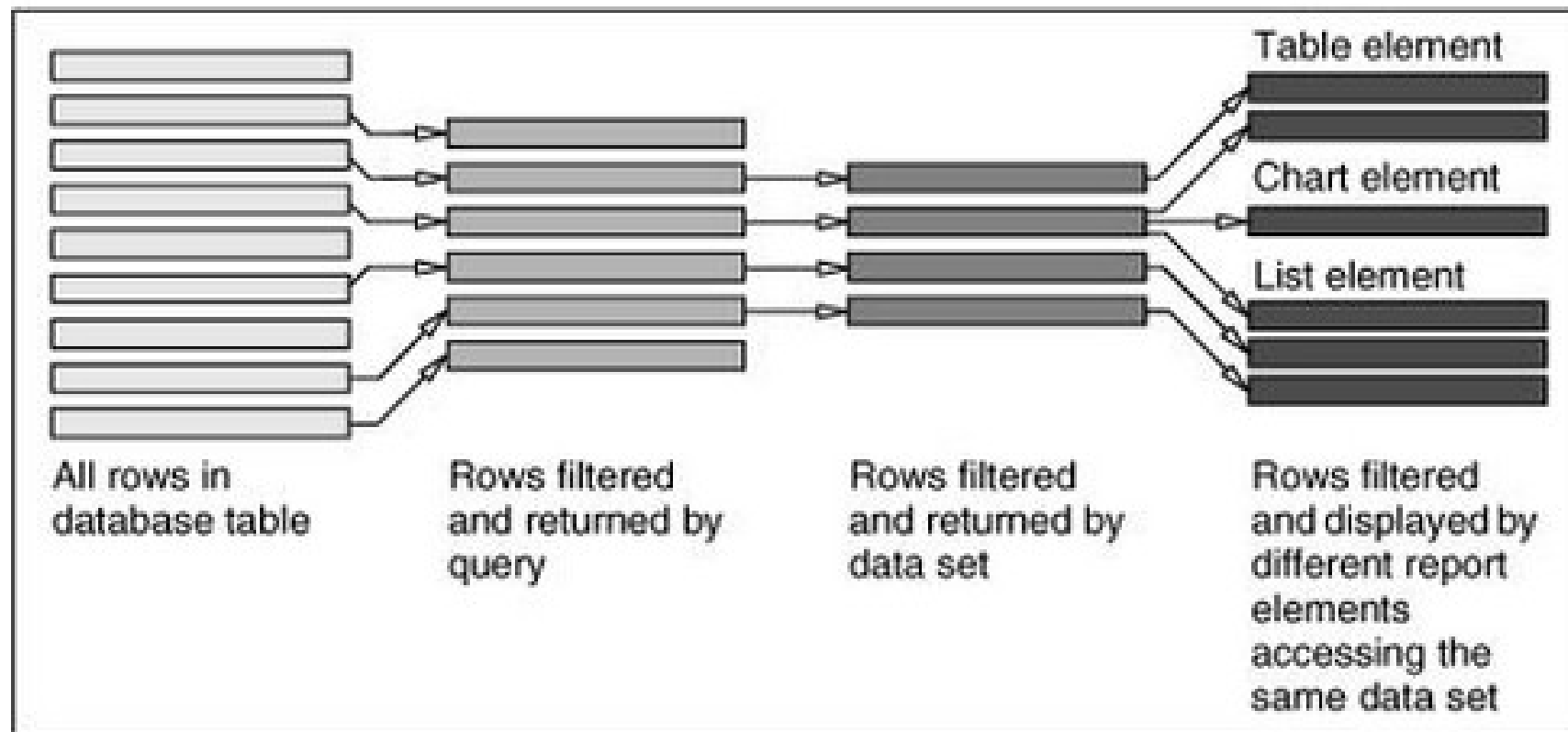
---

BIRT six niveaux de comparaison de chaîne de caractères :

- **ASCII** : Prise en compte de la casse et des accents  
 $A < B < a < à$
- **Primary** : Ignore la casse et les accents  
 $a = A = à = < b < c$
- **Secondary** : Ignore la casse  
 $ab = Ab < àb < ac$
- **Tertiary** : Prise en compte de la casse et des accents  
 $aa < Aa < aà$
- **Quaternary** : La ponctuation, la casse et les accents sont pris en compte  
 $ab < a - b < aB$
- **Identical** : Basé sur Unicode, permet de différencier des chaînes qui seraient égales via le mode Quaternary. Très gourmand en performance et peu utile dans la pratique



# Filtres





# Types de condition de filtre

---

- × Comparaison : *Less than, Equal, Greater than, etc..*
- × Valeur null : *Is Null, Is Not Null*
- × Intervalle (inclusif) : *Between, Not Between*
- × Logique conditionnelle : *==, >, ||, &&, Is True*
- × Expression régulière Javascript : *Match*
- × Recherche de motif à la SQL : *Like*
- × Les plus grandes ou plus basse valeurs : *Top n, Bottom n, Top percent n, Bottom percent n*

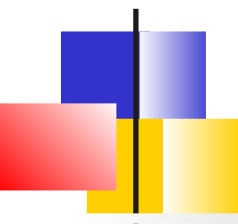


# Filtres multiples

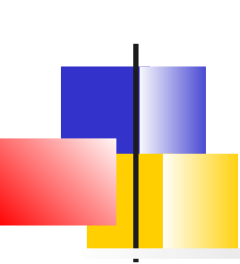
---

Lors de la définition de plusieurs filtres, les filtres sont combinés avec l'opérateur logique « ET »

Les données extraites remplissent donc les conditions de tous les filtres



# Groupes et agrégation



# Regroupement de données

---

- La création de groupes permet de :
  - Ajouter des titres au début de chaque groupe
  - Ajouter des informations résumés (fonction d'agrégation au début ou fin de groupe)
  - Insérer un saut de page à chaque changement de groupe
  - Générer une table des matières affichant les différentes valeurs du groupe et permettant la navigation





# Ajout de groupe

L'ajout de groupe se fait via l'éditeur de propriétés et l'onglet « *Groups* » sur un élément de type table ou liste






Les propriétés à renseigner sont alors :

- Le **nom** du groupe
- Le **champ** de regroupement
- **L'intervalle** de regroupement
- **Affichage du détail** ou non
- La **table des matières** : l'expression affichée dans la TDM et le style
- **Sort direction** : le sens du tri du champ de regroupement
- Les **sauts de page** et si les entêtes sont répétées à chaque page
- **Sorting** : Pour indiquer un autre critère de tri que le champ de regroupement
- Des **filtres** de données



# Lignes des groupes

L'ajout d'un groupe ajoute 2 nouvelles lignes à l'élément : l'entête et le bas de groupe

	Product	Price	Quantity	Total
	[PRODUCTCODE]			
	[PRODUCTCODE]	[PRICEEACH]	[QUANTITYORDERED]	[EXTENDEDPRICE]
	Group Footer Row			
	Footer Row			

Group header row

Group footer row

Il est possible d'ajouter d'autres lignes de ce type



# Hiérarchie des groupes

---

Les groupes d'une table ou d'une liste sont hiérarchiques.

L'interface du designer permet de modifier l'ordre des groupes, insérer un groupe entre 2 groupes, etc.



# Groupement par intervalle

---

Plutôt que le critère de groupe se base sur l'opération d'égalité, il est possible de spécifier un intervalle de regroupement.

- Par exemple, regrouper par mois, année, etc.

La propriété intervalle a une signification différente en fonction du type de données :

- *String* : séquence de caractères d'une longueur particulière.  
Par exemple : Regrouper selon les 2 premiers caractères
- *Numérique* : Pas de regroupement avec une valeur de base (par défaut, la première valeur trouvée). Par exemple regrouper par pas de 1000 en démarrant à 0
- *Date* : Période de regroupement (heure, jour, semaine, mois, trimestre, année)



# Filtre

---

Les filtres permettent de ne prendre en compte qu'un sous-ensemble du jeu de données

Un filtre est défini par :

- Une expression
- Un opérateur
- Éventuellement, 2 valeurs complétant l'opérateur

Par exemple, on peut facilement dans un groupe n'afficher que les valeurs les plus représentatives avec l'opérateur *Top n*



# Propriété *Drop*

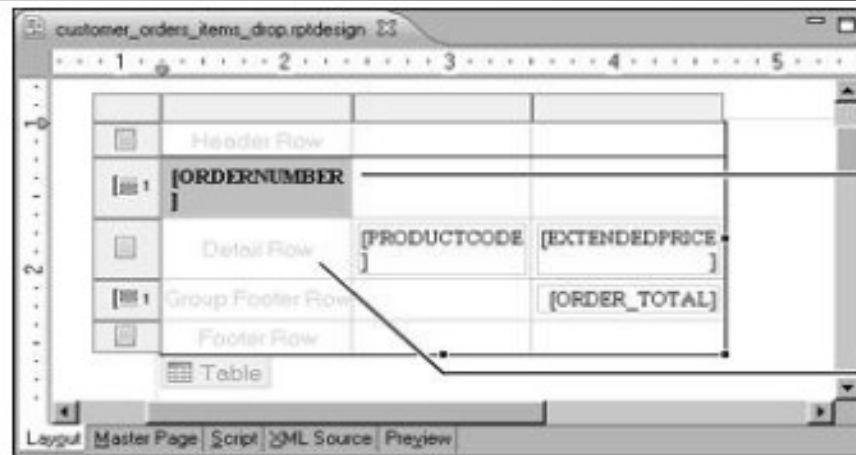
---

La propriété ***Drop*** d'une cellule d'entête permet de déplacer l'entête de groupe dans la première ligne détail.

- Cette propriété ne peut être utilisée que pour l'entête de groupe placée directement au dessus du détail.
- De plus, il faut qu'une cellule vide existe sous le titre

L'entête de groupe peut soit couvrir les ligne détail soit couvrir l'ensemble des lignes du groupe

# Example



Group heading in a cell having a background color. The cell's Drop property is set to either Detail or All.

Dropping a cell requires an empty cell below it

Group header cell (shown with gray background) dropped to the detail rows

Order 10200	S24_1785	\$3,285.81
	S32_1374	\$2,831.85
	S32_4289	\$1,764.45
		<hr/> \$7,882.11
Order 10202	S32_2206	\$901.53
	S32_4485	\$2,530.84
		<hr/> \$3,432.37

Group header cell dropped to all rows in the group

Order 10200	S24_1785	\$3,285.81
	S32_1374	\$2,831.85
	S32_4289	\$1,764.45
		<hr/> \$7,882.11
Order 10202	S32_2206	\$901.53
	S32_4485	\$2,530.84
		<hr/> \$3,432.37



# Agrégation de données

---

Les groupes permettent d'agréger des données.

L'agrégation consiste à effectuer des calculs sur un ensemble de données : moyenne, comparaison, somme, etc.

L'ensemble de données peut être celui d'un groupe ou du jeu de données complet.





# Types de calcul

---

BIRT propose 2 types de calcul :

- Les calculs **unique** qui ont la même valeur pour toutes ligne du jeu de données.  
Ce type de calcul s'affiche généralement dans les entêtes ou les bas de groupe/tableau/liste
- Les calculs **dynamiques ou flottant** : La valeur est calculée à chaque itération et est en général différente pour chaque ligne du jeu de données.  
Ce type de calcul s'affiche généralement dans la partie détail d'un groupe/tableau/liste



# Types de calcul

---

**AVERAGE** : Moyenne

**CONCATENATE** : Concaténation avec un caractère séparateur optionnel

**COUNT** : décompte

**COUNTDISTINCT** : décompte distinct

**FIRST/LAST** : Première/dernière des valeurs

**IS-BOTTOM-N** : retourne vrai si la valeur est avant la valeur N, faux sinon

**IS-BOTTOM-N-PERCENT** : Idem en pourcentage

**IS-TOP-N** : retourne vrai si la valeur est après la valeur N, faux sinon

**IS-TOP-N-PERCENT** : Idem en pourcentage

**MIN/MAX** : Valeur minimum/maximum

**MEDIAN** : Valeur médiane

**MODE** : Valeur apparaissant le plus de fois dans le jeu de données



# Types de calcul

---

**MOVINGAVE** : La moyenne mouvante sur un échantillon de données (intervalle)

**PERCENTILE, QUARTILE** : centile, quantile

**RANK** : Le rang de la valeur dans l'ensemble des valeurs

**PERCENTRANK** : Idem RANK avec une valeur comprise entre 0 et 1

**PERCENTSUM** : Le pourcentage sur la somme des valeurs

**RUNNINGCOUNT** : Décompte courant pour la ligne

**RUNNINGSUM** : Somme courante pour la ligne

**STDDEV, VARIANCE** : Déviation, variation

**SUM** : Somme

**WEIGHTEDAVE** : Moyenne pondérée, les poids sont indiqués dans une autre colonne du jeu de données



# Utilisation

---

Comme pour toute donnée dynamique, il est nécessaire de créer une liaison pour afficher une agrégation

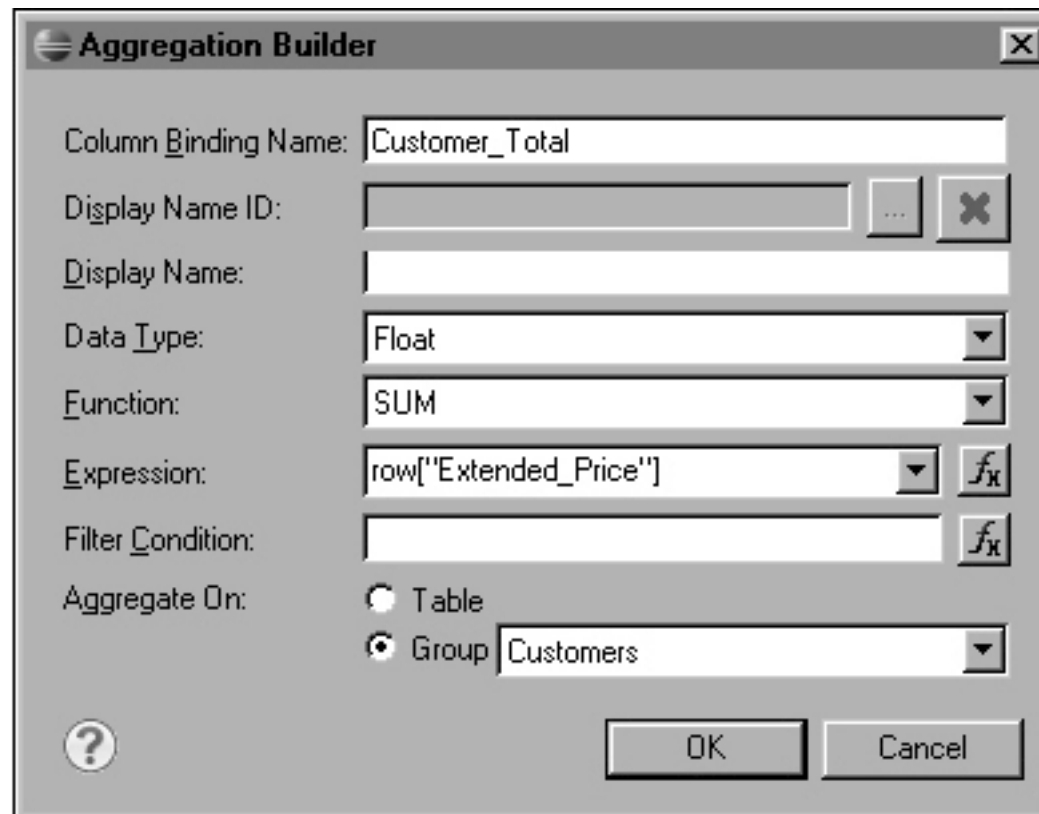
- On peut l'associer à un élément texte ou data

BIRT Designer propose l'assistant *Aggregation Builder* permettant de définir le nom de liaison et le calcul



# Aggregation Builder

---



The screenshot shows the 'Aggregation Builder' dialog box with the following fields and settings:

- Column Binding Name: Customer\_Total
- Display Name ID: (empty field with a selection button and a close button)
- Display Name: (empty field)
- Data Type: Float
- Function: SUM
- Expression: row["Extended\_Price"]
- Filter Condition: (empty field)
- Aggregate On: ☐ Table ☒ Group Customers

Buttons at the bottom: ? (help), OK, and Cancel.

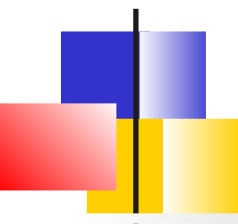


# Propriétés de l'agrégation

---

L'assistant permet de renseigner :

- Le nom de la liaison
- Le type de données
- Le type de calcul
- L'expression : Les valeurs sur lesquelles le calcul est effectué
- *Aggregate on* : L'ensemble du jeu de données ou un groupe particulier
- Éventuellement des conditions de filtre du jeu de données
- Éventuellement des champs supplémentaires en fonction du type de calcul



# Paramètres

# Paramètres de rapport



Les **paramètres** sont des données passées lors de la génération du document dynamique qui ne proviennent pas de la source de données:

- Par exemple des données de présentation comme le nom de l'utilisateur, le titre du rapport, ...
- Des paramètres de la requête SQL. Les paramètres agissent alors comme des filtres sur les données récupérées. Exemple : une période, un montant, etc ...
- Des paramètres servant à l'évaluation d'expression

Les paramètres ont une portée globale et peuvent être utilisés à tous les endroits du rapport





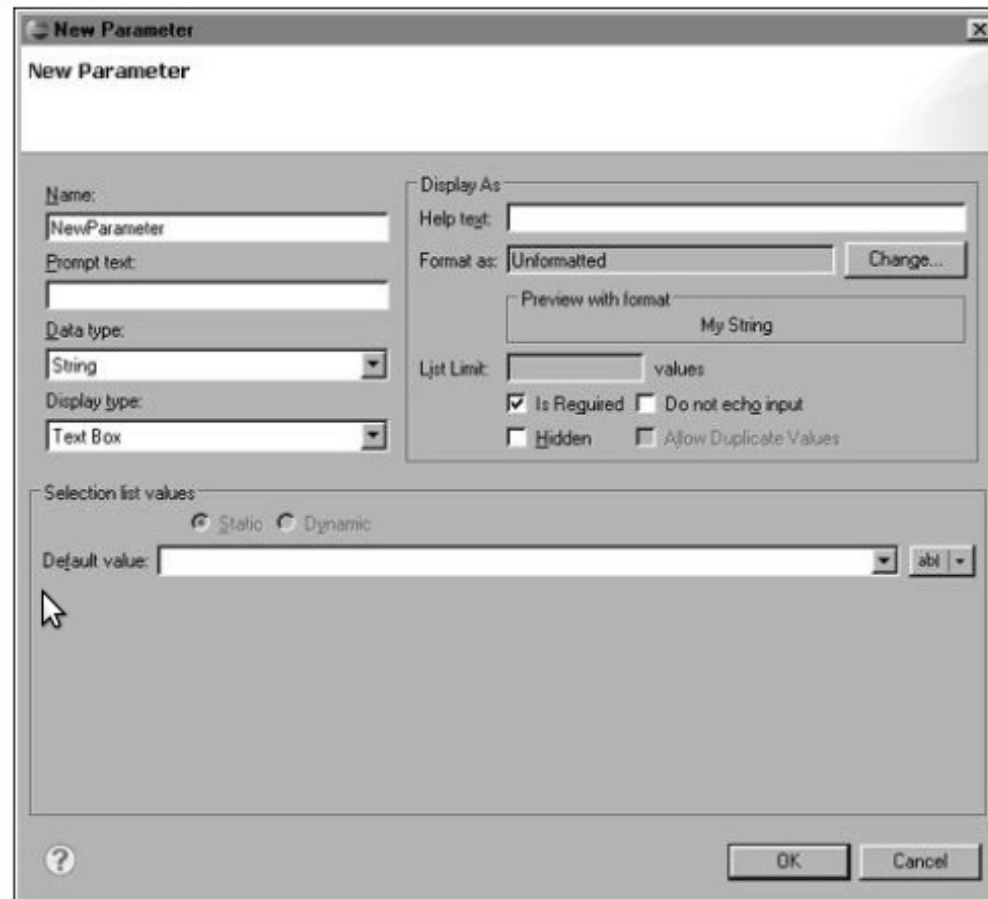
# Création

---

La création d'un paramètre nécessite :

- La définition de ses propriétés de base :  
**nom** et **type** de données
- La présentation du paramètre à l'utilisateur : le **type de contrôle** (champ, liste déroulante, etc.), la **valeur par défaut**, le texte de **description**, l'organisation des différents paramètres en **groupes** logiques

# Création



The image shows a 'New Parameter' dialog box with the following fields and options:

- Name:** NewParameter
- Prompt text:** (empty text box)
- Data type:** String (dropdown menu)
- Display type:** Text Box (dropdown menu)
- Display As:**
  - Help text:** (empty text box)
  - Format as:** Unformatted (dropdown menu) with a 'Change...' button.
  - Preview with format:** My String (text box)
  - List Limit:** (empty text box) values
  - ☒ Is Required ☐ Do not echo input
  - ☐ Hidden ☐ Allow Duplicate Values
- Selection list values:**
  - ☒ Static ☐ Dynamic
  - Default value:** (empty text box) [abi] (dropdown menu)

Buttons at the bottom: ? (help), OK, and Cancel.



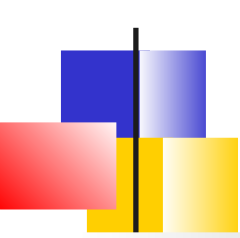
# Paramètres de requête

---

Des paramètres peuvent également être définis au niveau d'un jeu de données.

Cela s'effectue en ajoutant le caractère ? dans la requête SQL

En général, il faut ensuite associer le paramètre du jeu de données à un paramètre de rapport



# Utilisation dans une requête

---

Pour effectuer une requête paramétrée :

1) Utiliser le caractère marqueur « ? » dans la requête SQL

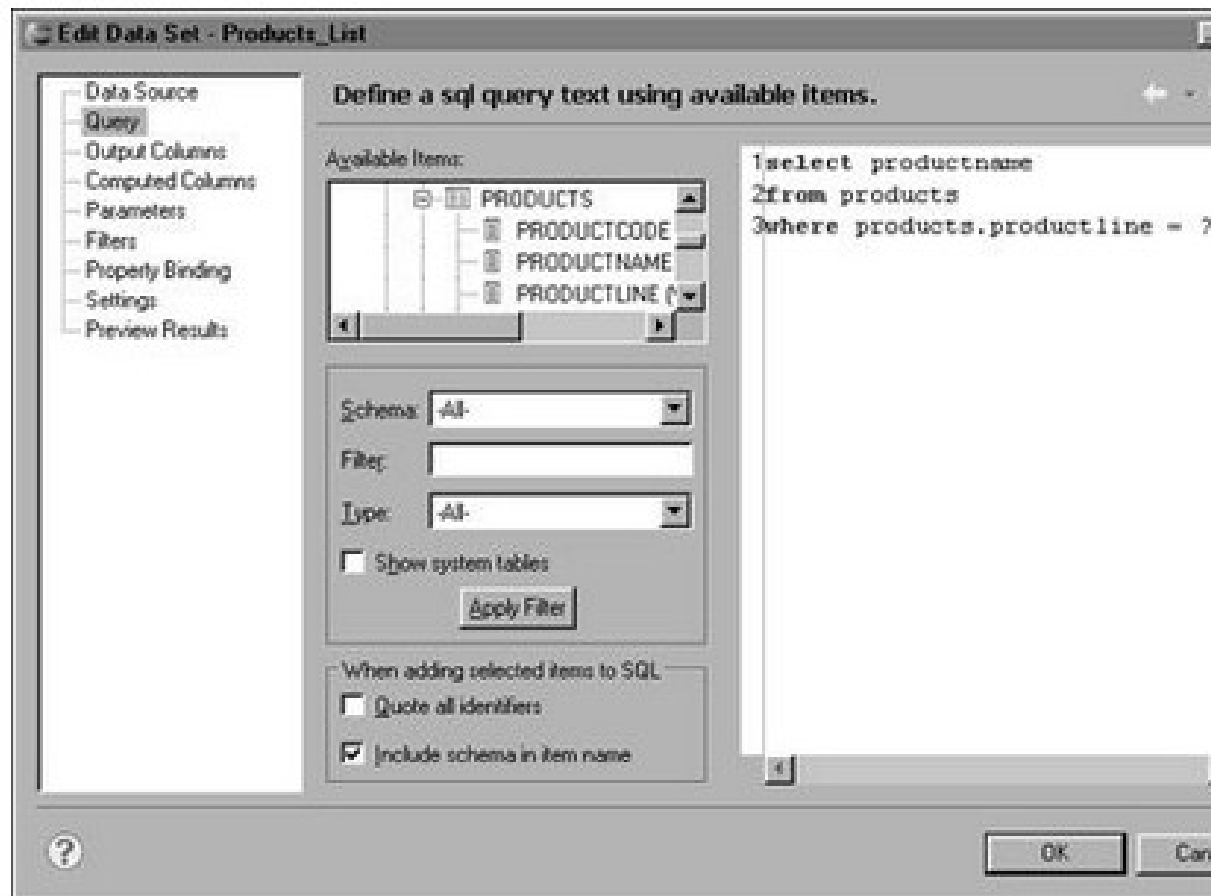
*SELECT \* FROM Orders WHERE CustomerID = ?*

2) Définir un paramètre au niveau du jeu de donnée

*A chaque marqueur ?, doit correspondre un paramètre du jeu de données. L'ordre de création est alors crucial lorsque plusieurs marqueurs sont utilisés.*

3) Associer le paramètre du jeu de donnée à un paramètre de rapport


# Exemple








# Exemple

---

 **Edit Parameter**

<u>N</u> ame	dsp_productline
<u>N</u> ative Name	
<u>D</u> ata <u>T</u> ype	String
<u>D</u> irection	Input
<u>D</u> efault <u>V</u> alue	N/A
<u>L</u> inked To Report Parameter	rp_ProductLine



# Paramètre du dataset

Edit Data Set - Products List

Parameters

	Name	Data Type	Direction	Default Value	Linked To Report Parameter
1	DSP_quantityinstock	Integer	Input	N/A	RP_quantityinstock
2	DSP_productvendor	String	Input	N/A	RP_productvendor

New... Edit... Remove Up Down

OK Cancel



# Construction de requête

---

Les paramètres peuvent également servir à construire la requête SQL

Il faut alors utiliser l'onglet « *Property Binding* » et utiliser la notation ***params[]***

Par exemple :

```
"Select customerName from Customers  
where customers.city IN ('" +  
params['city1']  
+"', '" + params['city2'] + "'" )"
```



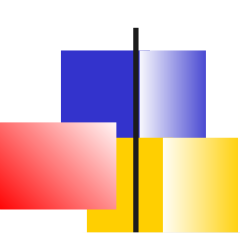


# Utilisation dans les filtres

---

Les paramètres du rapport peuvent également être utilisés dans les expressions des filtres

La notation ***params["nom"].value*** est alors utilisée



# Présentation des paramètres

---

Plusieurs propriétés jouent sur la présentation du paramètre :

- **Prompt text** : Le texte affiché expliquant le paramètre
- **Default value** : La valeur par défaut utilisée si l'utilisateur ne renseigne pas le paramètre (peut utiliser une expression)
- **Help text** : Texte d'aide
- **Display type** : Le type de contrôle utilisé pour saisir le paramètre
- **Format As** : Le format utilisé pour les listes déroulantes, boutons radio ou liste sélectionnable



# Listes de valeur

---

Les listes de valeur utilisées pour les liste déroulante, liste ou radio-button peuvent être statiques ou dynamiques (valeurs extraites de la source de données)

- Les listes dynamiques ne sont pas supportées par les radio-button



# Liste statique

---

La création d'une liste statique peut se faire :

- Soit en saisissant chaque valeur
- Soit en important les valeurs d'un champ du jeu de données

Pour chaque valeur, il est possible de définir le texte alternatif affiché à l'utilisateur, une clé utilisée pour la localisation



# Liste dynamique

---

Les valeurs proviennent d'un jeu de données.

- Le nombre maximal de valeurs peut être précisé
- Une expression pour le texte alternatif également



# Sélection multiple

---

Avec le contrôle liste, l'utilisateur peut saisir plusieurs valeurs pour un paramètre

- Les valeurs multiples sont utiles pour les filtres utilisant l'opérateur ***In***

# Exemple



**New Filter Condition**

Please fill following contents to create a filter condition.

Enter a filter condition:

row["PRODUCTVEN"  $f_x$  In

Value productvendor".value Add

params["RP\_productvendor"].value

Edit...  
Remove  
Remove All

? OK Cancel

The image shows a 'New Filter Condition' dialog box. It has a title bar with a close button. The main area contains a text prompt 'Please fill following contents to create a filter condition.' followed by a section 'Enter a filter condition:'. This section includes a dropdown menu with 'row["PRODUCTVEN', a function button 'fx', and another dropdown menu with 'In'. Below this is a 'Value' section with a dropdown menu showing 'productvendor".value' and an 'Add' button. To the right of the 'Value' section are three buttons: 'Edit...', 'Remove', and 'Remove All'. At the bottom, there is a help icon '?', an 'OK' button, and a 'Cancel' button. A list box below the 'Value' section contains the text 'params["RP\_productvendor"].value'.



# Paramètres en cascade

---

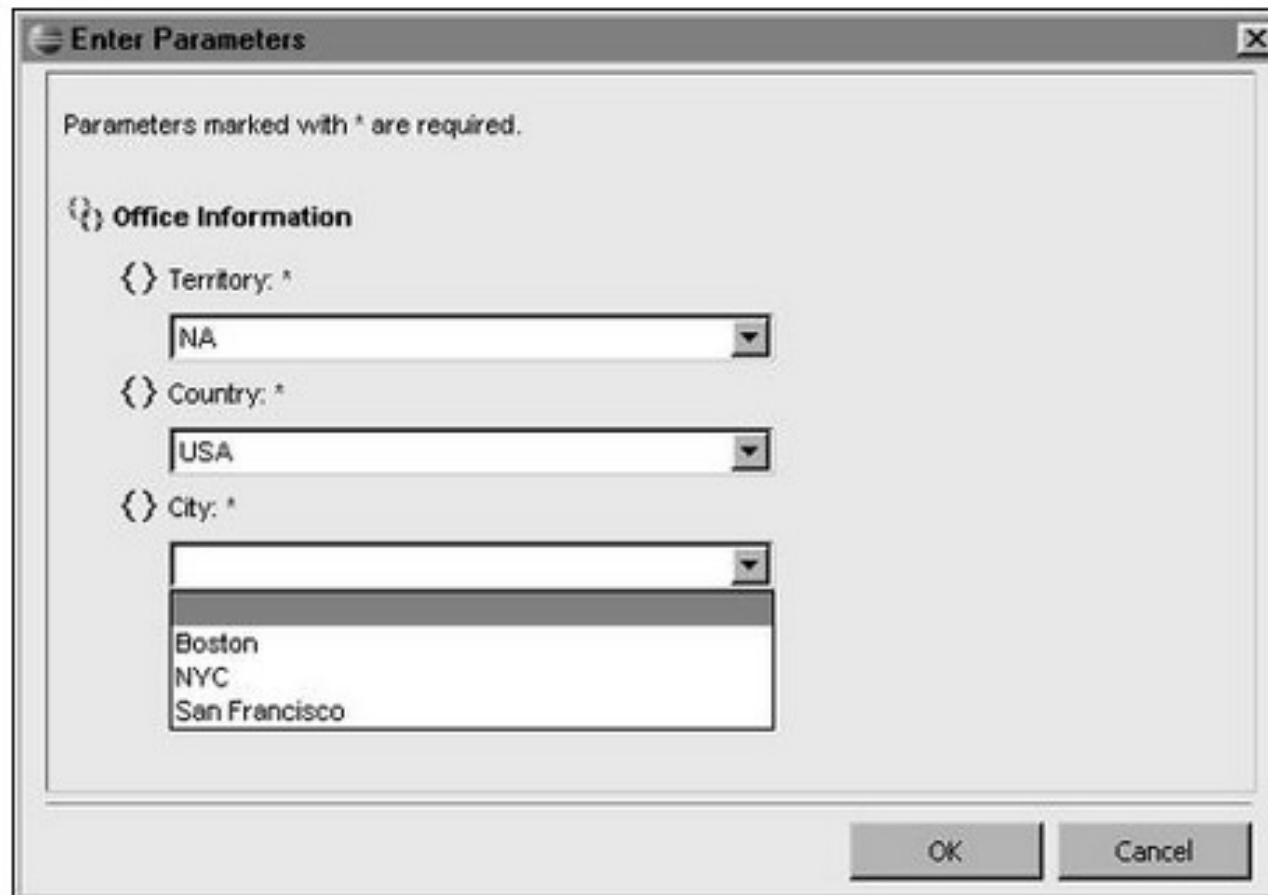
Les **paramètres en cascade** sont des paramètres qui ont une relation hiérarchique

- Quand l'utilisateur sélectionne une valeur d'un niveau, les valeurs possibles pour le sous-niveau se mettent à jour

Les paramètres en cascade utilisent soit le même jeu de données soit différents jeux liés via un champ commun.




# Exemple




The screenshot shows a standard Windows-style dialog box titled "Enter Parameters". Inside the dialog, there is a message "Parameters marked with \* are required." followed by a section titled "Office Information" which is preceded by a folder icon. Under this section, there are three required fields, each marked with a curly brace icon and an asterisk: "Territory: \*", "Country: \*", and "City: \*". The "Territory" field is a dropdown menu currently showing "NA". The "Country" field is a dropdown menu currently showing "USA". The "City" field is a dropdown menu that is currently open, showing a list of three options: "Boston", "NYC", and "San Francisco". At the bottom right of the dialog, there are two buttons: "OK" and "Cancel".


Enter Parameters

Parameters marked with \* are required.


 **Office Information**

 Territory: \*

NA

 Country: \*

USA

 City: \*

Boston  
NYC  
San Francisco

OK Cancel



# Un seul jeu de données

---

*DataExplorer → New Cascading Parameter Group*

Sélectionner « *Single Data Set* »

Ajouter les paramètres un à un en indiquant le champ correspondant du jeu de données

# Example

**New Cascading Parameter Group**

General

Cascading Parameter Group Name:

Prompt text:

☒ Single Data Set ☐ Multiple Data Set

Parameters

Name	Data Set	Value	Display Text
tp_Territory	Territory_Country_...	TERRITORY	<Display Column not s...
tp_Country		COUNTRY	<Display Column not s...
tp_City		CITY	<Display Column not s...

Add Edit Delete

Properties

Name:

Prompt text:

Data type:

Display type:

Default value:

Sort

Sort by:  Sort direction:

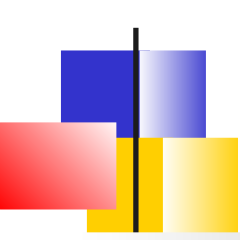
More Options

Help Text:

Format as:

List Limit:  values ☒ Is Required ☐ Allow Multiple Values

OK Cancel



# Plusieurs jeux de données

---

*DataExplorer → New Cascading Parameter Group*

Sélectionner «*Multiple Data Set* »

Ajouter les paramètres un à un en indiquant le champ correspondant du jeu de données

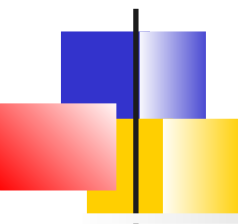
Pour les jeux de données dépendant ajouter une clause *WHERE* avec un marqueur et le lier au paramètre du niveau supérieur



# Éléments avancés

---

Sous-rapport  
Liens hypertextes  
Graphiques  
Tableaux croisés



## Sous-rapport



# Introduction

---

Un **sous-rapport** est un rapport apparaissant à l'intérieur d'un autre rapport

BIRT permet de disposer les sous-rapports dans le rapport parent grâce à ses éléments *Grid*, *List* et *Table*

Les sous-rapports peuvent être indépendants ou liés entre eux



# Structure du rapport

---

Liste : Un rapport contenant plusieurs sous-rapport liés utilise typiquement une liste comme conteneur de haut niveau

Grille : Permet de positionner des sous-rapports indépendants





# Rapports indépendants

## Top 10 Products

1992 Ferrari 360 Spider red	\$276,839.98
2001 Ferrari Enzo	\$190,755.86
1952 Alpine Renault 1300	\$190,017.96
2003 Harley-Davidson Eagle Drag Bike	\$170,686.00
1968 Ford Mustang	\$161,531.48
1969 Ford Falcon	\$152,543.02
1980s Black Hawk Helicopter	\$144,959.91
1998 Chrysler Plymouth Prowler	\$142,530.63
1917 Grand Touring Sedan	\$140,535.60
2002 Suzuki XREO	\$135,767.03

## Top 10 Sales Representatives

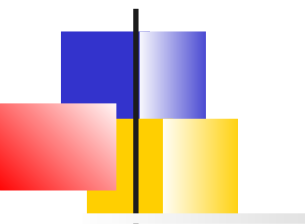
Gerard Hernandez	\$1,258,577.81
Leslie Jennings	\$1,081,530.54
Pamela Castillo	\$868,220.55
Larry Bott	\$732,096.79
Barry Jones	\$704,853.91
George Vanauf	\$669,377.05
Peter Marsh	\$584,593.76
Louie Bondur	\$569,485.75
Andy Foster	\$562,582.59
Steve Patterson	\$505,875.42

## Top 10 Customers

Euro+ Shopping Channel	\$820,689.54
Mini Gifts Distributors Ltd.	\$591,827.34
Australian Collectors, Co.	\$180,585.07
Muscle Machine Inc	\$177,913.95
La Rochelle Gifts	\$158,573.12
Dragon Souvenirs, Ltd.	\$156,251.03
Down Under Souvenirs, Inc	\$154,622.08
Land of Toys Inc.	\$149,085.15
AV Stores, Co.	\$148,410.09
The Sharp Gifts Warehouse	\$143,536.27

## Top 10 Cities

Madrid	\$979,880.77
San Rafael	\$591,827.34
NYC	\$497,941.50
Auckland	\$292,082.87
Singapore	\$263,997.78
Paris	\$240,649.68
San Francisco	\$199,051.34
New Bedford	\$190,500.01
Nantes	\$180,887.48
Melbourne	\$180,585.07

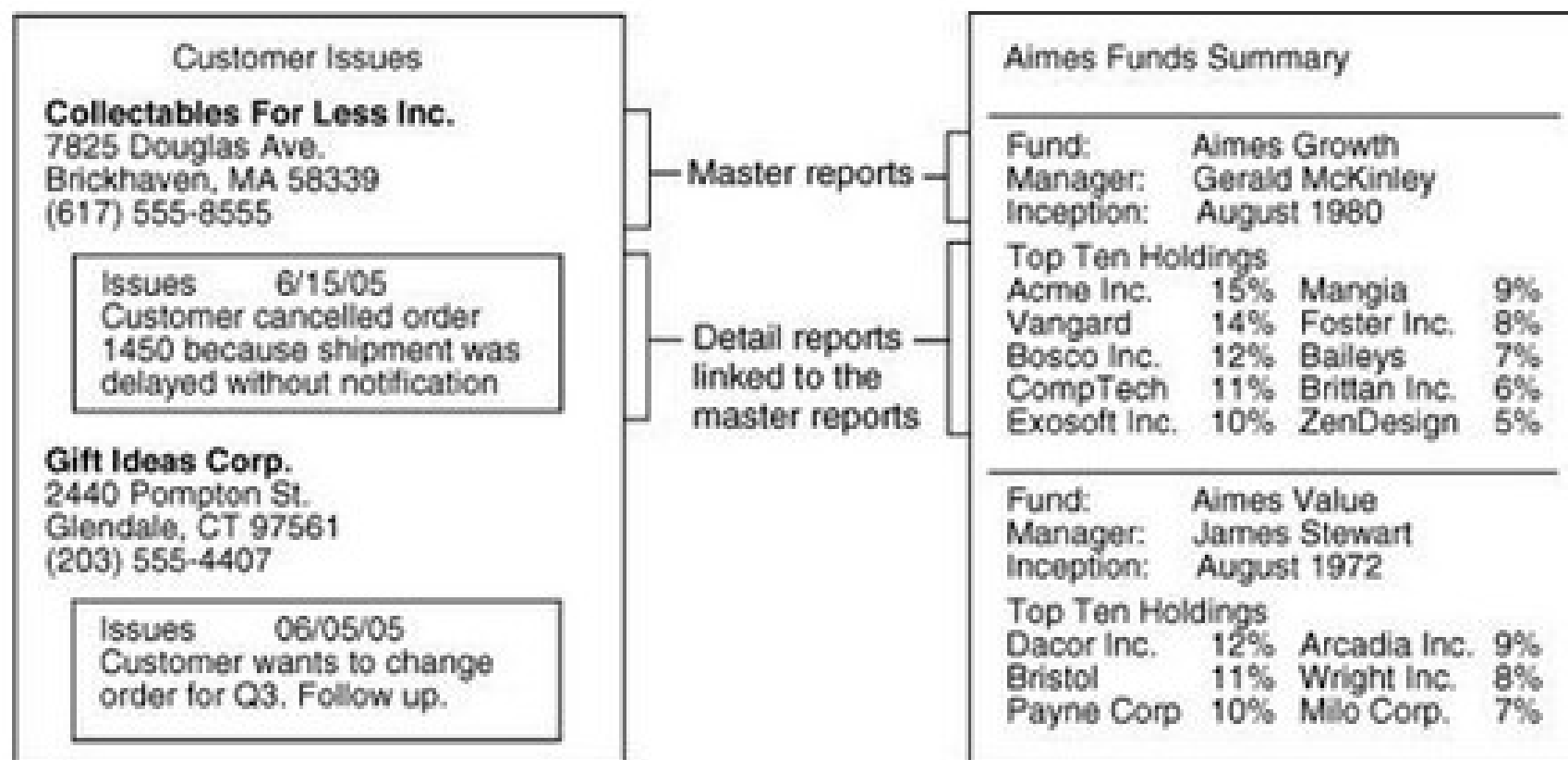


# Design

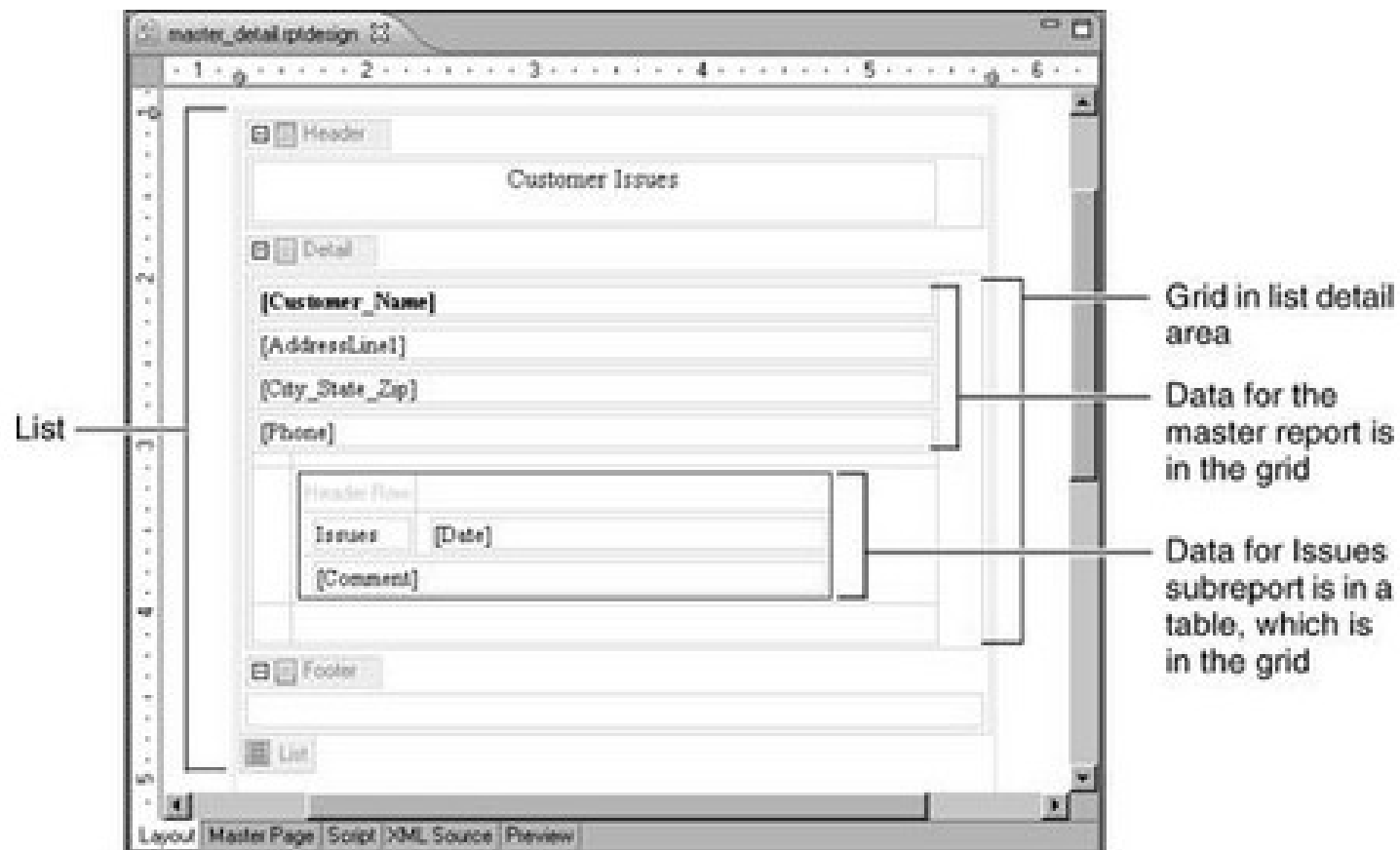
A fixed-height empty grid row adds space between the top and bottom subreports

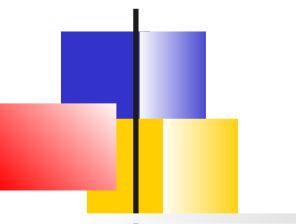
A fixed-width empty grid column adds space between the subreports

The screenshot shows a window titled 'subreports\_orig.rptdesign' with a horizontal ruler at the top (0 to 8) and a vertical ruler on the left (0 to 3). The main area contains a 2x2 grid of subreports. The top-left subreport is 'Top 10 Products' with fields '[PRODUCTNAME]' and '[Total\_Product]'. The top-right is 'Top 10 Sales Representatives' with '[REPRESENTATIVE]' and '[Total\_Rep]'. The bottom-left is 'Top 10 Customers' with '[CUSTOMERNAME]' and '[Total\_Customers]'. The bottom-right is 'Top 10 Cities' with '[CITY]' and '[Total\_City]'. A 'Grid' button is at the bottom left. The bottom of the window has tabs for 'Layout', 'Master Page', 'Script', 'HTML Source', and 'Preview'.



# Design avec Liste





# Design avec des tables

The screenshot shows a report design window titled "master\_detail\_in\_table.rptdesign". The report is structured as follows:

- Outer Table:** A table with a header row "Customer Issues" and detail rows containing fields: [Customer\_Name], [AddressLine1], [City\_State\_Zip], and [Phone].
- Inner Table:** A subreport table located within a detail row of the outer table. It has a header row "Issues" and detail rows containing fields: [Date] and [Comment].

Annotations with leader lines point to the following elements:

- Table:** Points to the outer table structure.
- Data for the master report in detail rows of the outer table:** Points to the detail rows of the outer table.
- Data for Issues subreport is in a table, which is in a detail row of the outer table:** Points to the inner subreport table.

The bottom of the window features a tabbed interface with "Layout" selected, and other tabs for "Master Page", "Script", "XML Source", and "Preview".



# Lier les rapports

---

Pour lier le sous-rapport au rapport parent par exemple :

- la requête du sous-rapport doit contenir un paramètre
- ce paramètre doit être lié à l'enregistrement courant du rapport parent.

# Exemple

**Edit Data Set - Data Set**

Parameters

	Name	Native Name	Data Type	Direction	Default Value	Linked To Report Parameter
1	custID		String	Input	112	None

New... Edit... Remove Up Down

OK Cancel

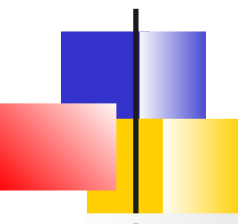
**Data Set Parameter Binding**

Data Set: Issues

Parameter	Data Type	Value
custID	String	row["customerNumber"]

Edit...

OK Cancel



# Liens hypertextes





# Introduction

---

Il est possible d'ajouter des éléments interactifs permettant une meilleure navigation dans le rapport.

- Des liens hypertexte ou des tables des matières peuvent être ajoutées
- Les graphiques peuvent également être interactifs
- Dans des rapports HTML, des check box, des boutons peuvent être ajoutés

Les fonctions interactives ne sont pas toujours disponible en fonction des formats de sortie

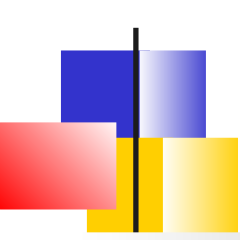


# Liens hypertexte

---

Les liens hypertexte peuvent pointer vers un élément du même rapport ou d'un autre rapport

L'élément source du lien doit être un élément *data*, *label*, *image* ou *graphique*



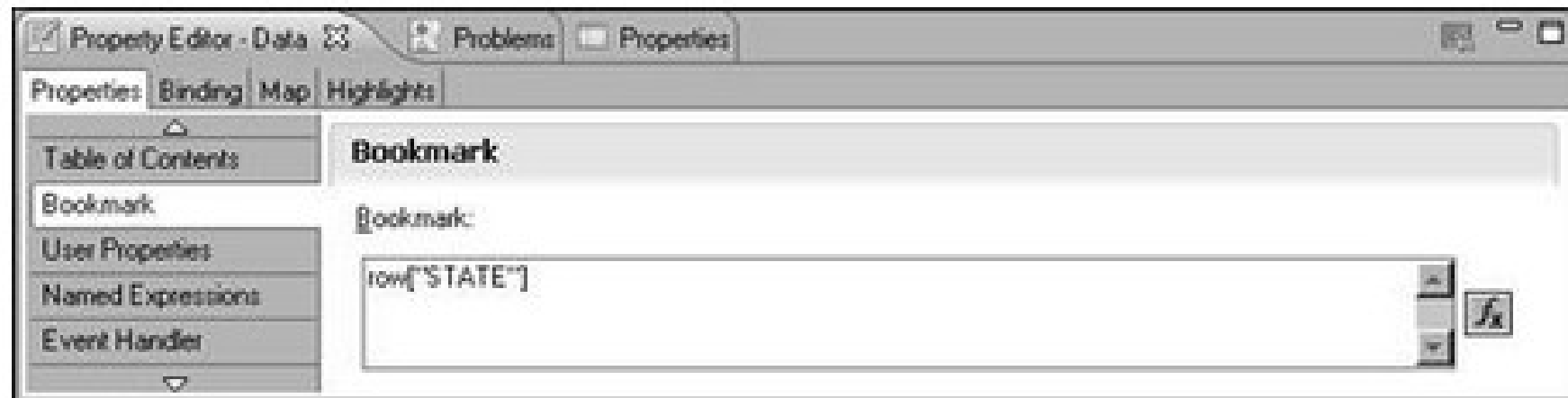
# Lien dans le même rapport

---

Pour réaliser un lien hypertexte dans le même rapport, il faut définir dans l'ordre :

- Un **signet** (bookmark ou cible) sous forme d'une expression
- Un **lien** hypertexte (source) qui a pour type *Internal Bookmark* et une expression qui doit se résoudre en un *bookmark*

# Examples



# Exemple hyperlink

The screenshot shows a report design tool interface. The main report area contains two sections: 'Sales Summary' and 'Sales Details'.

**Sales Summary:** This section includes a text box labeled '[STATE]' and a data element '[STATETOTAL]'. To the right is a 3D pie chart with five segments labeled A, B, C, D, and E. The values for these segments are: A: 6, B: 4, C: 12, D: 8, and E: 10.

**Sales Details:** This section contains a table with four columns: '[STATE]', '[STATETOTAL]', '[STATEPERCENT]', and an unlabeled column. The table has two data rows. The first row has headers: 'Product Code', 'Total Units', 'Amount', and '% of Total Amount'. The second row has data: '[PRODUCTCODE]', '[UNITTOTAL]', '[PRODUCTTOTAL]', and '[PRODUCTPERCENT]'.

Annotations in the image:

- A line points from the text 'Source data element' to the '[STATETOTAL]' data element in the 'Sales Summary' section.
- A line points from the text 'Choose to open Hyperlink Options' to the 'Edit...' button in the 'Hyperlink' property editor at the bottom.

The 'Hyperlink' property editor at the bottom shows the 'Link To' field set to 'row("STATE")'.



# Lien vers un autre rapport

---

Pour effectuer un lien vers un autre rapport, il faut définir un bookmark de la même façon mais choisir pour le lien le type ***drill-through***

Ce type de lien nécessite des informations supplémentaires :

- Le nom du rapport cible (.rptdesign ou .rptdocument)
- Le bookmark dans le rapport cible ou une entrée de la table des matières
- Si le lien doit provoquer l'ouverture d'une nouvelle fenêtre ou non
- Le format de sortie du rapport

Lors de l'activation du lien, il est également possible de passer un paramètre au rapport cible.

**Hyperlink Options**

Select Hyperlink Type:

- ☐ No Link
- ☐ URI
- ☐ Internal Bookmark
- ☒ Drill-through

---

Step 1: Select a target report

☒ Report Design: SalesDetails.rptdesign

☐ Report Document:

Report Parameters:

Parameters	Required	Data Type	Values
RP_State	✓	String	row["STATE"]

---

Step 2: Select a target anchor

☒ Target Bookmark

☐ Table of Contents Entry in Target Report

row["STATE"]

---

Step 3: Create a link expression that matches the target bookmark.

Link Expression: row["STATE"]

---

Step 4: Show target report in

☒ New Window ☐ Parent Frame

☐ Same Frame ☐ Whole Page

---

Step 5: Select a format for target report

☐ Format the target report in:

---

Step 6: Tool Tip

Tool Tip: Click to link to detailed sales information

---

?

OK Cancel



# Lien vers un contenu externe

---

Le lien peut également pointer vers une ressource externe.

- Le type de lien est alors **URI**
- Il faut indiquer l'adresse complète cible et si le lien doit ouvrir une nouvelle fenêtre ou non





# Table des matières

---

Par défaut, un rapport groupé inclut une table des matières qui affiche les valeurs des groupes

Les entrées sont générées pour tous les groupes (organisés hiérarchiquement)

Pour ne pas inclure un groupe particulier dans la table des matières , il faut supprimer l'expression dans la propriété « *Table Of Contents Item Expression* » de l'éditeur de groupe



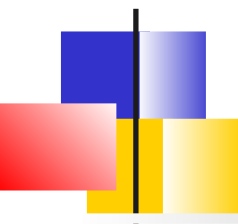
# Entrée de la TOC

---

Il est possible de générer des entrées pour d'autres éléments de rapport que les groupes

*Property Editor → Table of Contents*

Spécifier alors une expression qui correspond à la valeur à afficher dans la table des matières



# Graphiques



# Introduction

---

BIRT propose un assistant « *Chart builder* » permettant de sélectionner un type et sous-type de graphique, d'associer les données et de les formater.

L'assistant affiche une barre de progression des différentes étapes de mise au point du graphique

Les graphiques peuvent afficher plusieurs séries correspondant à des couleurs différentes

Les graphiques peuvent être rendus en différents formats (SVG, BMP, PNG)



# Types de graphiques

---

**Zone** : Affiche un ensemble de points reliés par des lignes et formant une zone

**Barre** : Les valeurs sont affichées sous forme de barres verticales ou horizontales.  
Permet des comparaisons faciles

**Tube, cône, pyramide** : Identiques au barre avec des formes différentes

**Ligne** : Points reliés par des lignes

**Mètre** : Utilise une aiguille pour positionner les valeurs sur un cadran organisé en zones

**Camembert** : Les valeurs représentent une portion du camembert

**Dispersion** : Juste des points. Graphique scientifique

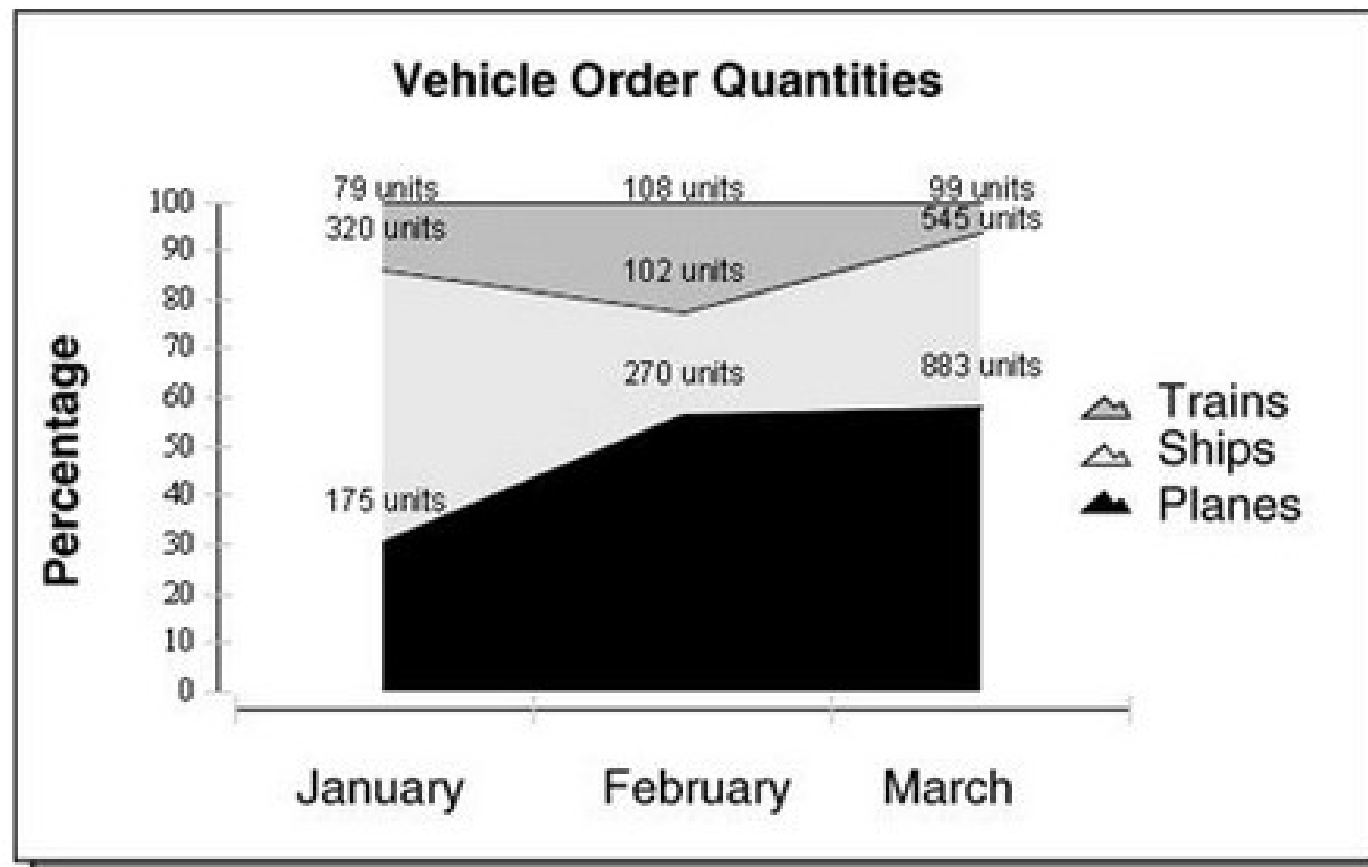
**Bulles** : Comme la dispersion mais les points sont des bulles plus ou moins grosses  
(3 axes)

**Boursier** : 4 données (Ouverture, Fermeture, Haute et basse) affichées sous forme de chandelier

**Différence** : Affiche les variations entre 2 ensembles de données en coloriant les zones entre les points de comparaison

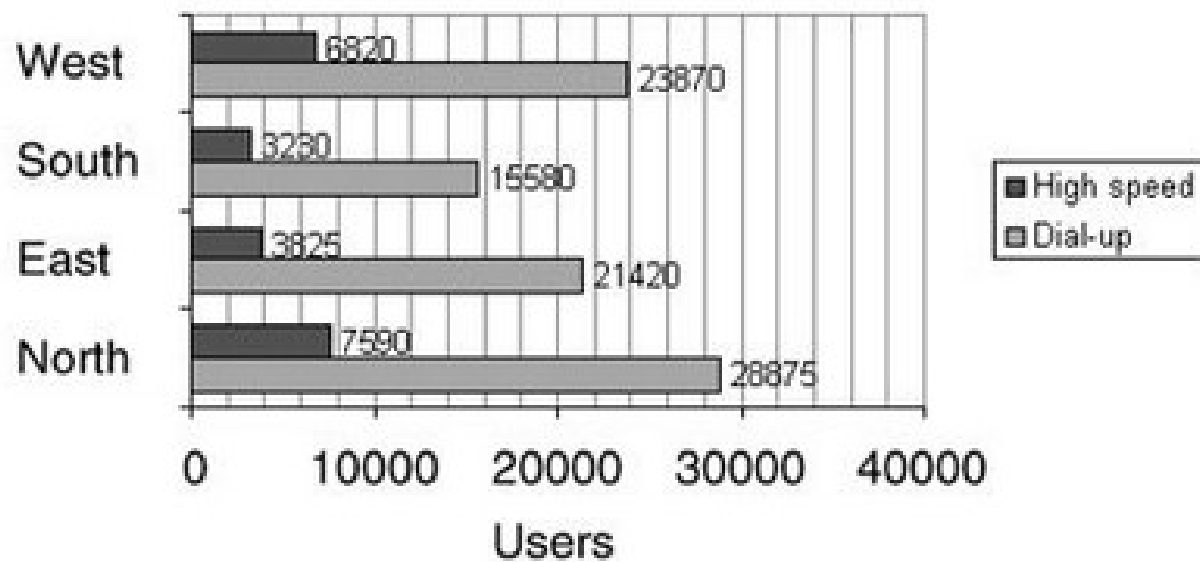
**Gantt** : Planification de projet

# Area

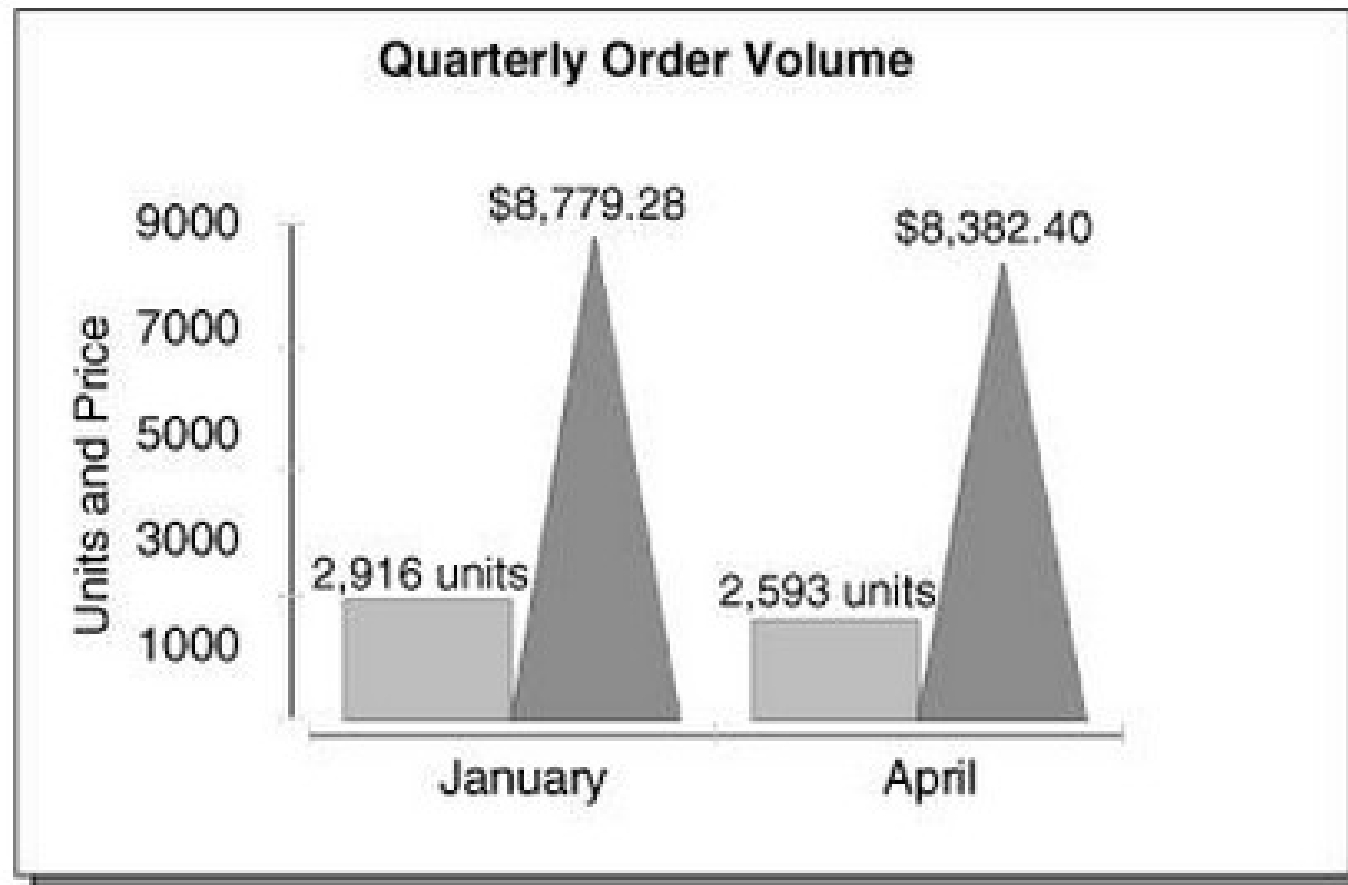


# Barre

**Types of Internet Connectivity in La Vern, MS**



# Cône

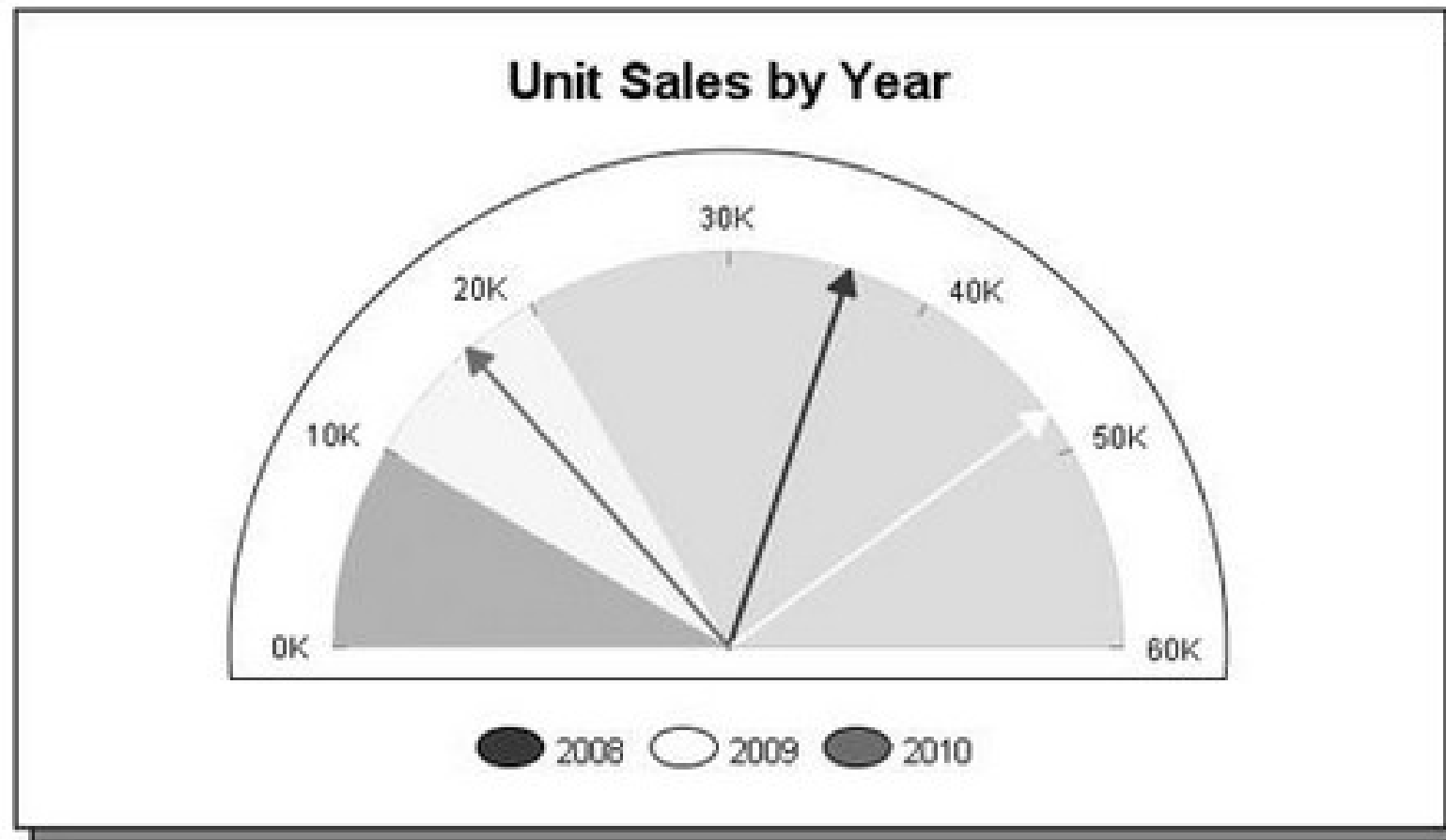




# Ligne

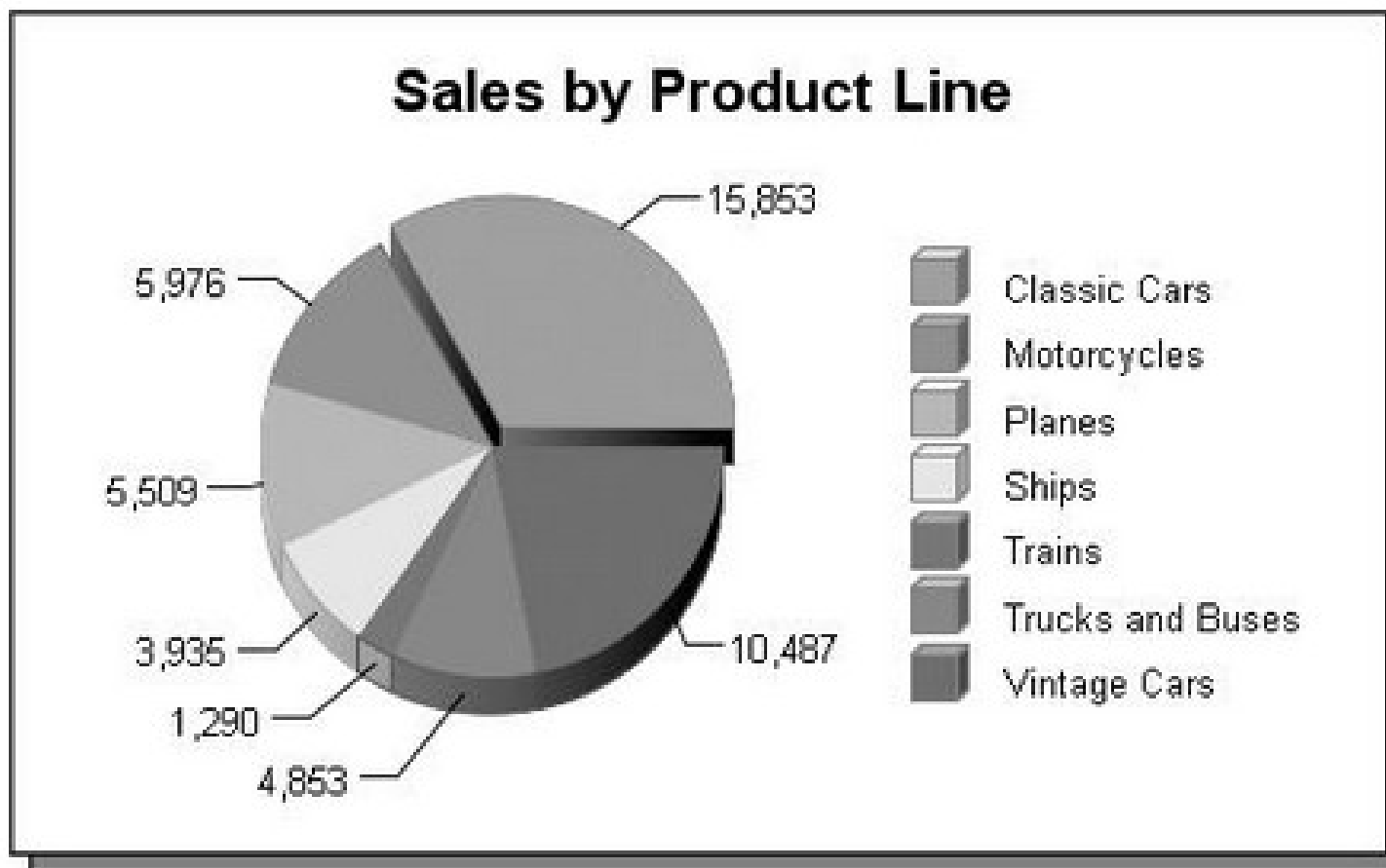


# Mètre

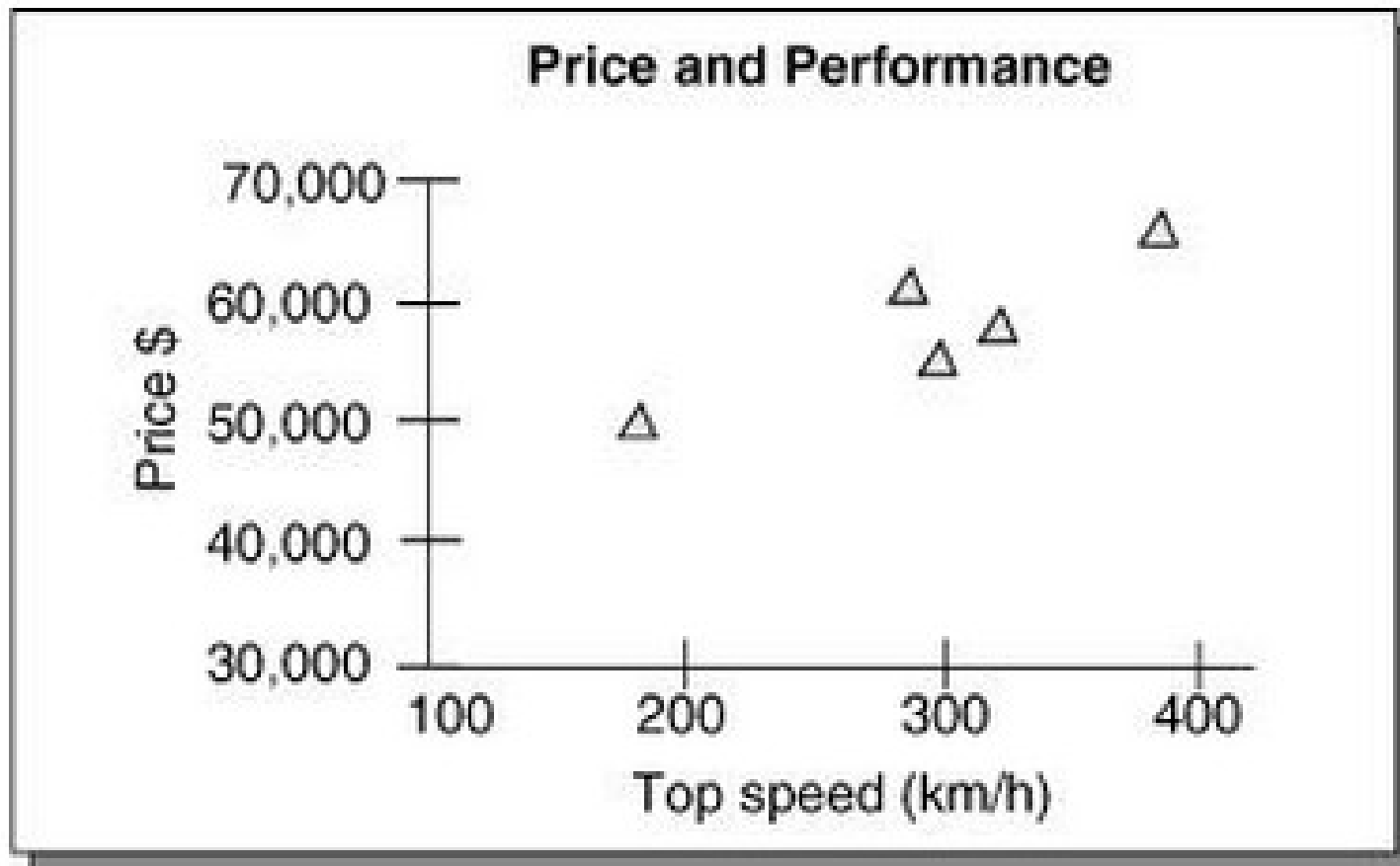




# Camembert



# Dispersion

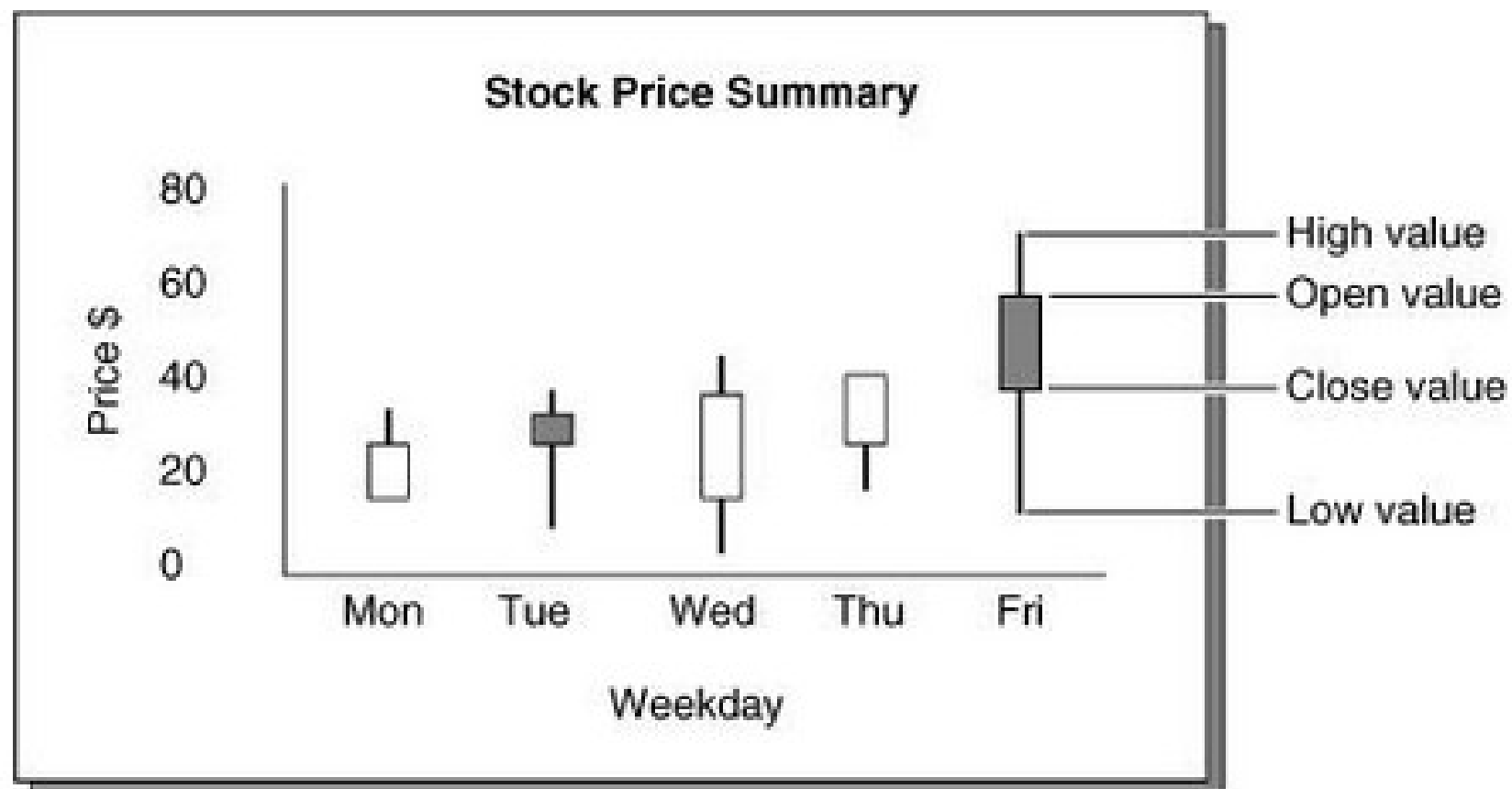


# Bulle

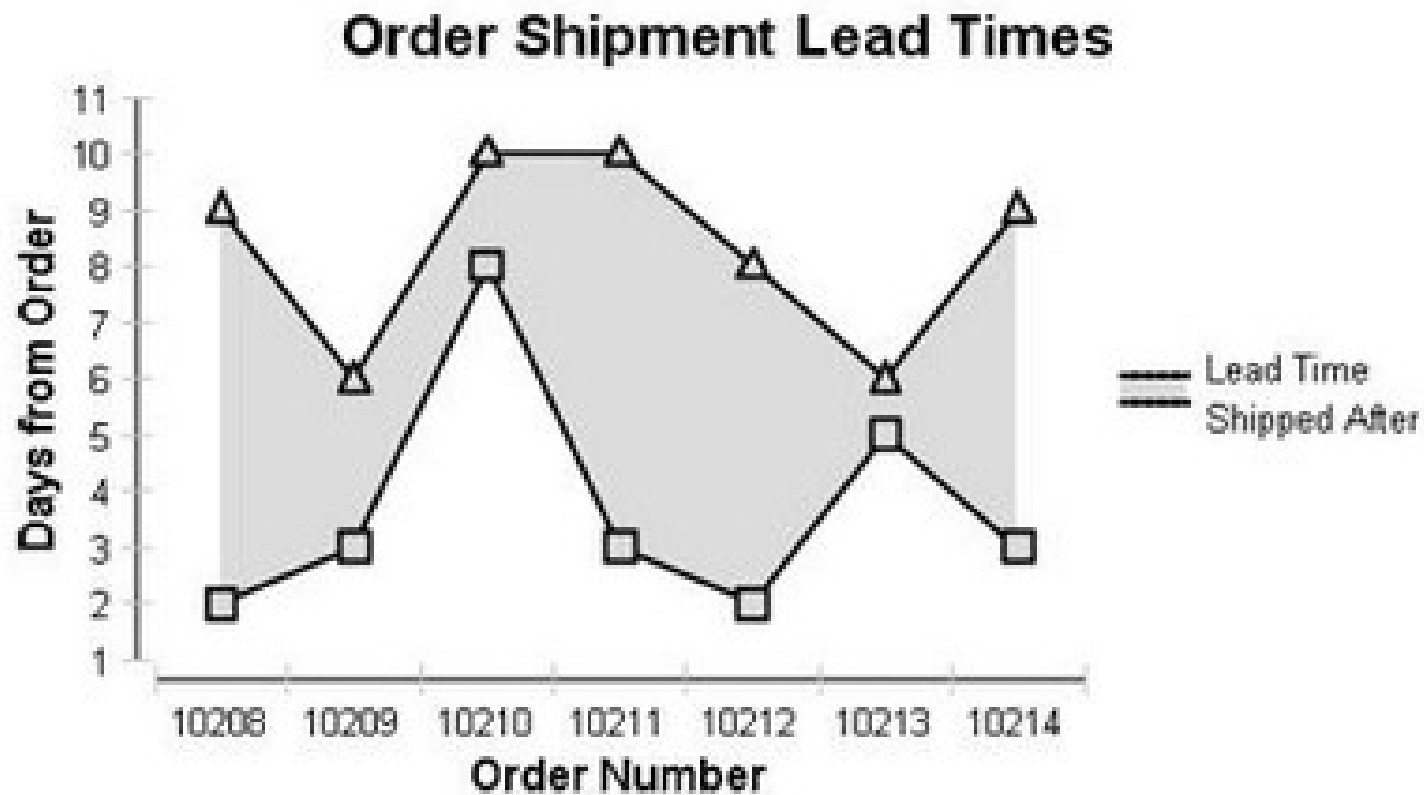




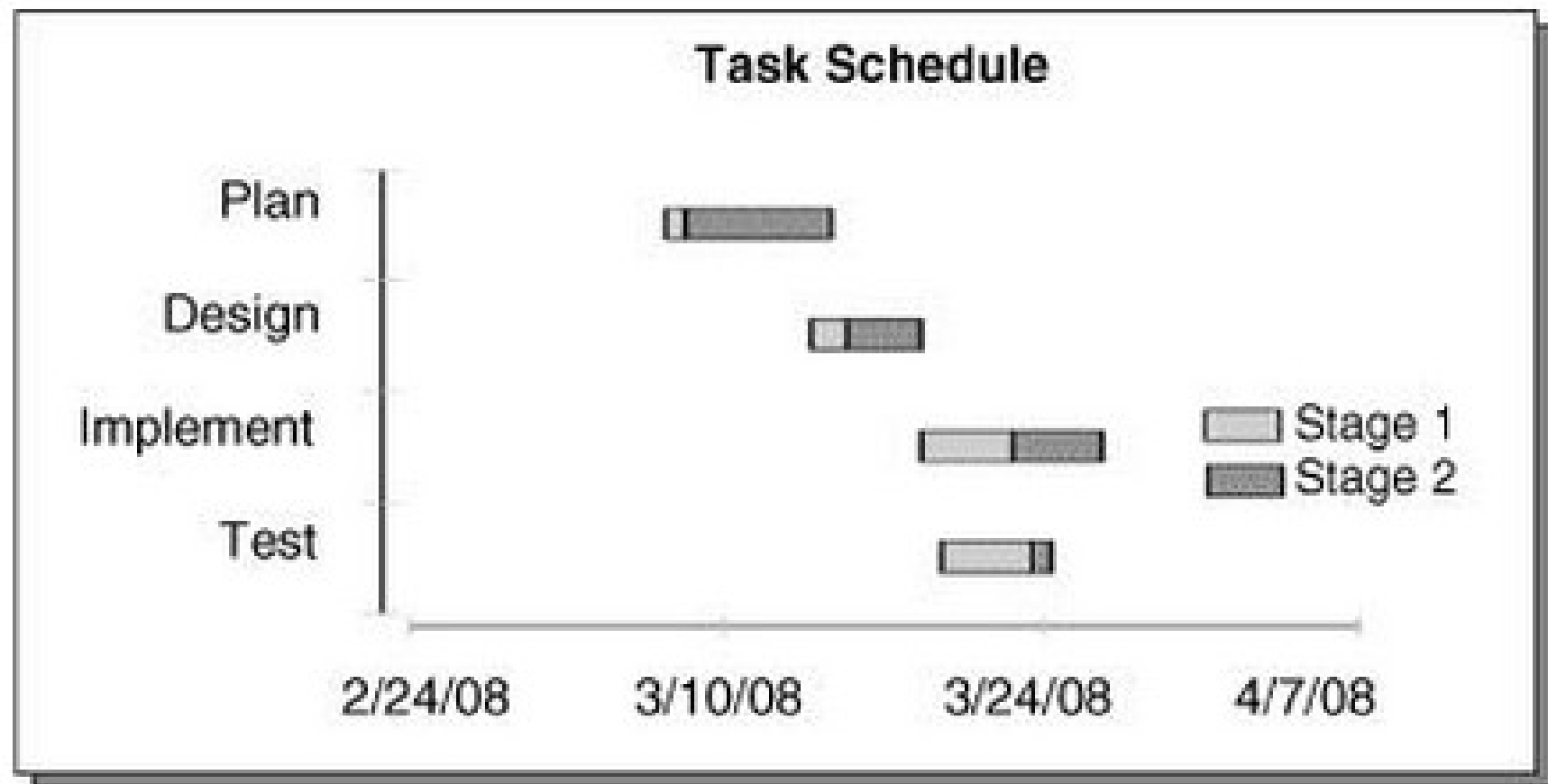
# Boursier



# Différence



# Gantt







# *Chart Builder*

---

L'assistant *Chart Builder* comporte 3 volets principaux

- La sélection du type de graphique
- La sélection des données
- Le formatage du graphique



# Première étape

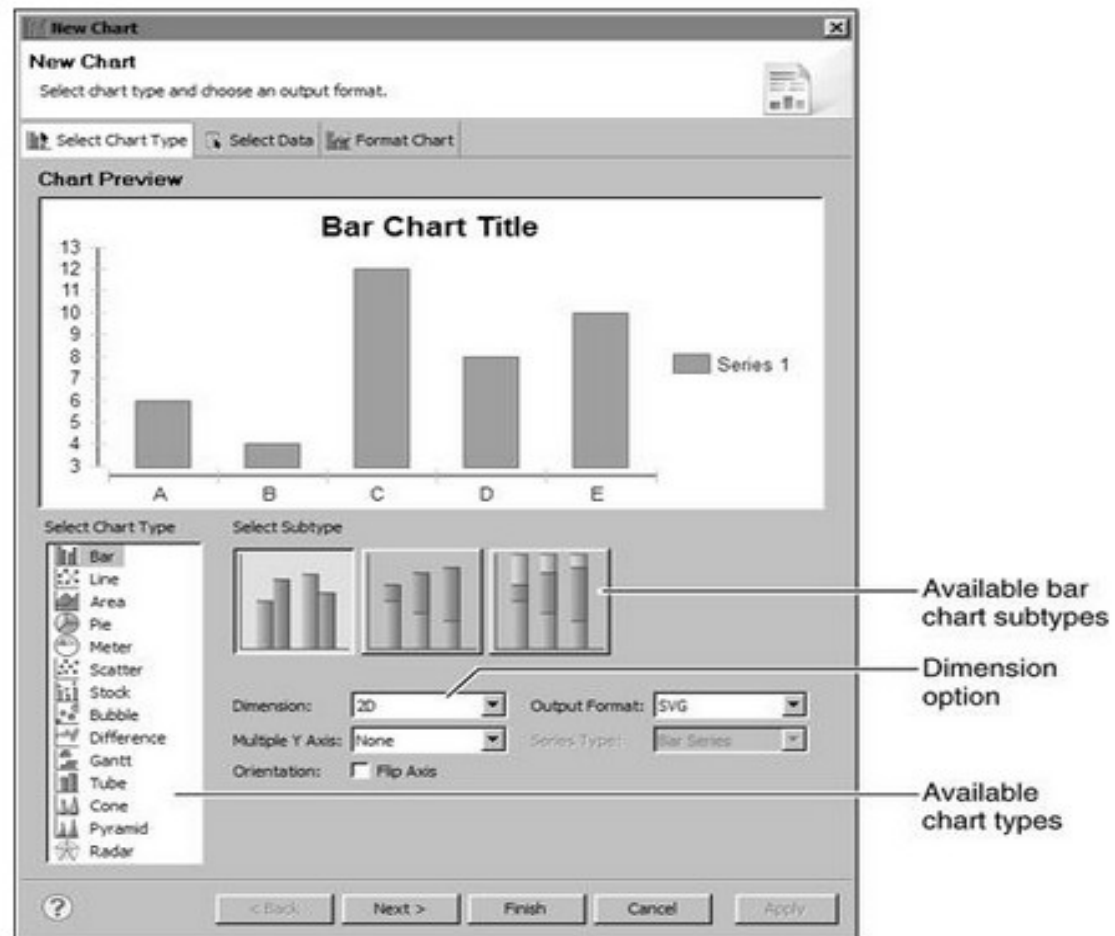
---

La première étape de l'assistant démarre lorsque l'on effectue un glisser/déposer de l'élément graphique.

Elle consiste à sélectionner le type de graphique et son sous-type

Le sous-type correspond généralement à l'utilisation des effets 3D, le basculement des axes

# Sélection du graphique





# Sous-types

---

Chart type	Dimension options		
	2D	2D with Depth	3D
Area	✓	✓	✓
Bar	✓	✓	✓
Bubble	✓	–	–
Cone	✓	✓	✓
Difference	✓	–	–
Gantt	✓	–	–
Line	✓	✓	✓
Meter	✓	–	–
Pie	✓	✓	–
Pyramid	✓	✓	✓
Scatter	✓	–	–
Stock	✓	–	–
Tube	✓	✓	✓



# Formats de sortie

---

4 formats de sortie sont disponibles :

- SVG (défaut) : Format XML vectoriel léger permettant l'ajout de plus d'interactivité
- JPEG, PNG : Adapté aux photos
- BMP : Format BitMap le plus lourd

Pour certains graphiques, il est possible de définir plusieurs axes Y, un effet 3D



# Jeu de données

---

La deuxième étape consiste à sélectionner les données du graphique.

Les graphiques ont le choix entre :

- utiliser leur propre jeu de données
- ou utiliser le jeu de données de leur conteneur (Liste, tableau ou grille)

Le jeu de données peut également être filtré ou paramétré et de nouvelles liaisons de données peuvent être définies

L'assistant permet de prévisualiser les premières données du jeu

Un texte alternatif peut être affiché, si aucune donnée n'est disponible.



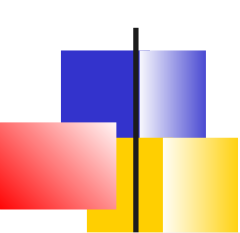
# Propre jeu de données

---

Pour utiliser un jeu de données indépendamment du conteneur, l'option à choisir est « *Use Data From* »

Puis choisir :

- Soit un jeu de données défini au niveau d'un rapport
- Soit un cube



# Jeu de données du conteneur

---

Pour utiliser les données d'un conteneur, le graphique doit être placé dans la cellule d'une table ou d'une liste.

2 options sont alors disponibles :

- ***Inherit columns and Groups*** : Le graphique utilise les groupes du conteneur (et les agrégations définies pour le groupe). Il ne peut pas redéfinir ni d'autres groupes, ni de filtres supplémentaires
- ***Inherit columns*** : Le graphique peut définir ses propres groupes, filtres et agrégation





# Association de données

---

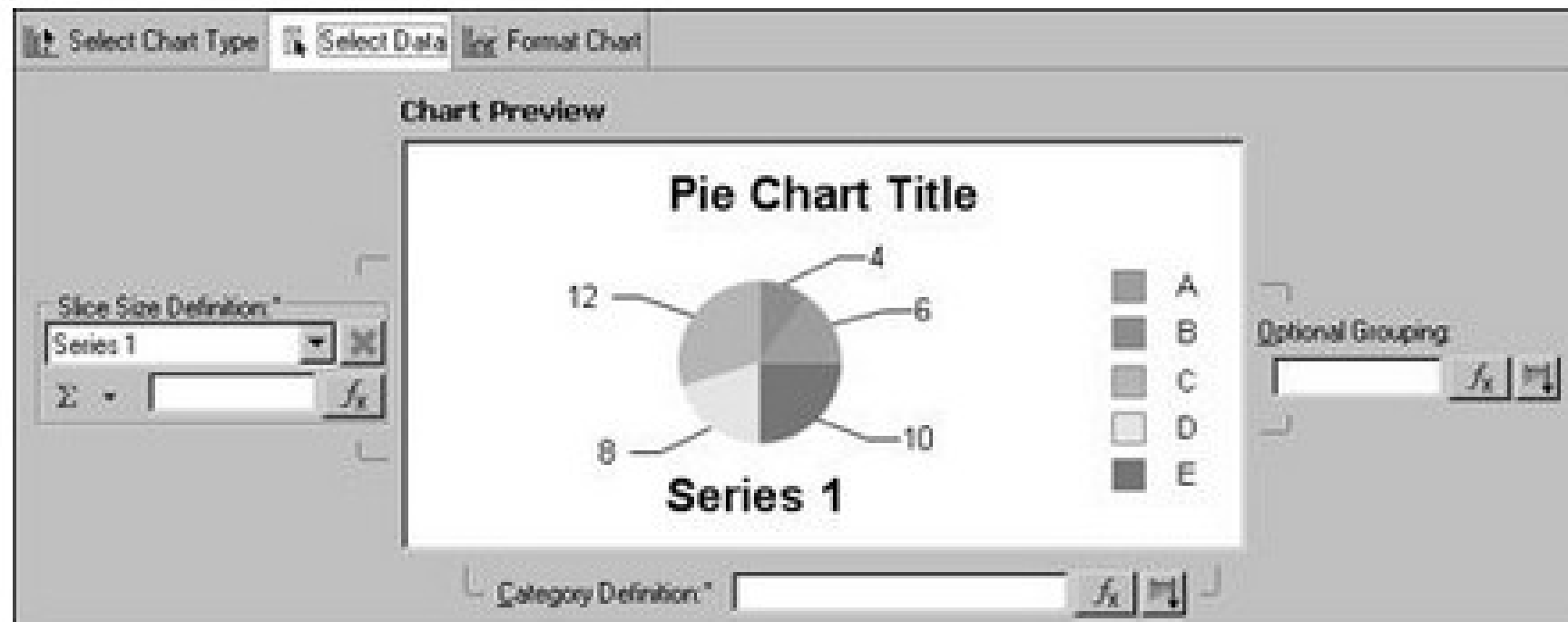
Chaque type de graphique utilise les données du jeu différemment

Il faut alors renseigner les expressions BIRT demandées par chaque type de graphique

Par exemple, pour un camembert :

- L'identification d'une part (la catégorie)
- L'expression donnant la taille de la part
- Un regroupement éventuel

# Exemple





# Axes d'un graphique

---

La plupart des graphiques ont 2 axes :

- L'**axe-x** généralement utilisé pour les catégories peut être de n'importe quel type de données
- L'**axe-y** généralement utilisé pour les valeurs de type numériques

Généralement, un graphique affiche une unique catégorie sur l'axe-x et plusieurs séries de valeurs sur l'axe-y

- Si le nombre de catégories ou de valeurs est trop important pour être affiché, il est possible de les regrouper au niveau du graphique

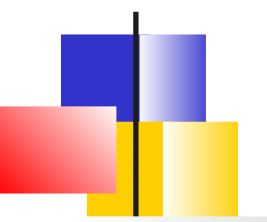


Chart type	Option name	Description
Area, bar, cone, difference, line, pyramid	Category (X) Series	Arranges data on <i>x</i> -axis. Can group, sort, and aggregate data.
	Value (Y) Series	Plots values on <i>y</i> -axis.
Bubble	Category (X) Series	Plots values on the <i>x</i> -axis. Can group, sort, and aggregate data.
	Y Value and Size	Plots values on <i>y</i> -axis and defines the size of the bubbles.
Meter	Category Definition	Requires a blank string " ": quotation mark, space, quotation mark.
	Meter Value Definition	Defines values of the dial and position of the needle. Creates multiple meters.
Alternate meter subtype	Category Definition	Requires a blank string " ": quotation mark, space, quotation mark.
	Meter Value Definition	Defines values of the dial and position of the needle.
Pie	Category Definition	Defines what slices represent.
	Slice Size Definition	Defines size of sectors. Creates multiple pies.
Scatter	Category (X) Series	Plots markers along <i>x</i> -axis. Groups data along <i>x</i> -axis.
	Value (Y) Series	Defines intersection of ( <i>x-y</i> ) value pairs. Defines multiple ( <i>x-y</i> ) value pairs.
Stock	Category (X) Series	Plots values along <i>x</i> -axis.
	Value (Y) Series	Defines four levels of data: high, low, open, close. Defines multiple sets of candlesticks.



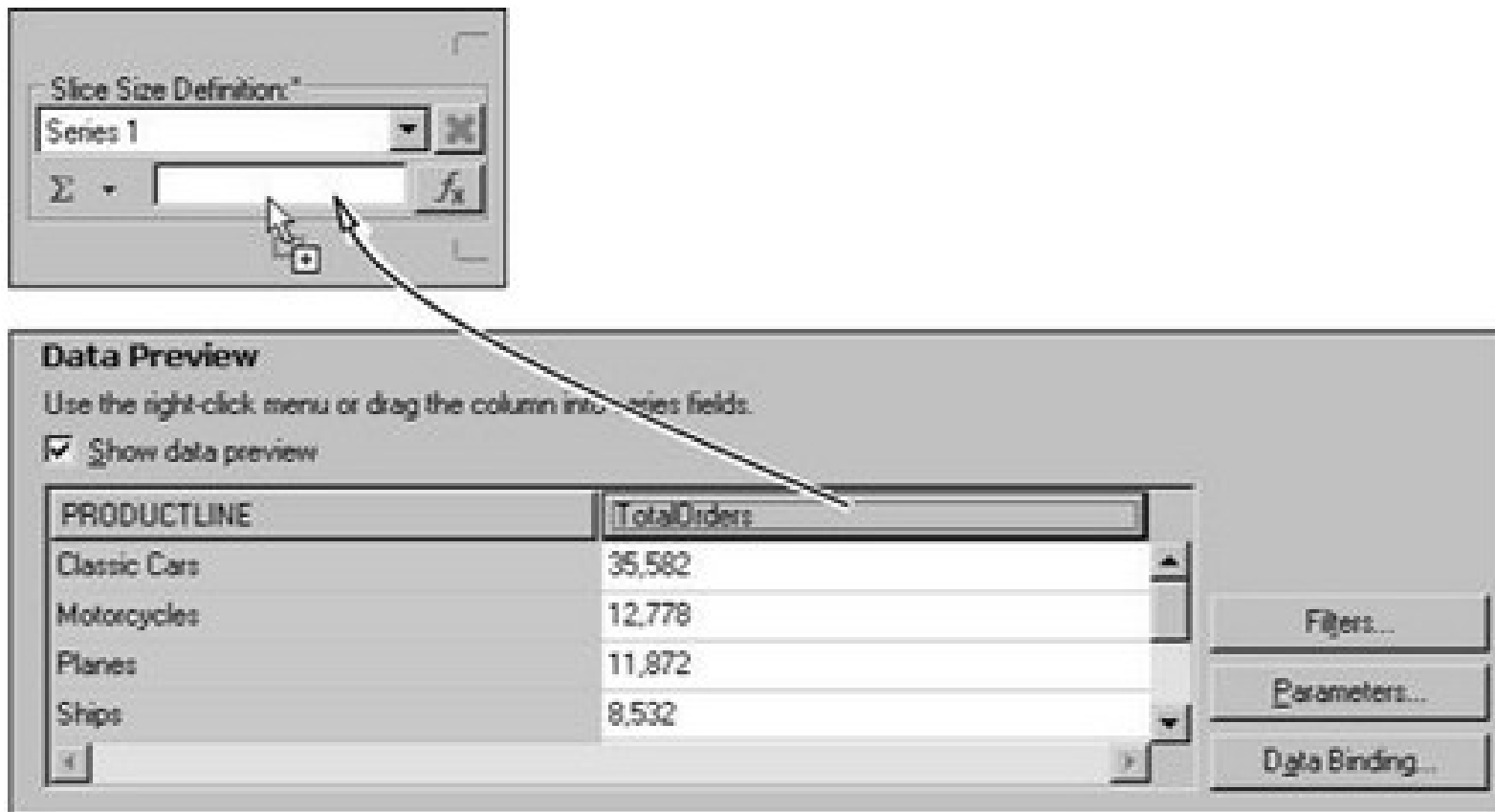
# Saisie des expressions

---

Plusieurs techniques sont possible pour renseigner les expressions d'un graphique :

- Effectuer un glisser/déposer à partir de la prévisualisation des données
- Utiliser l'expression builder
- Écrire directement l'expression

# Drag And Drop



The image shows a software interface with two main components. The top component is a 'Slice Size Definition' dialog box. It has a text field labeled 'Series 1' with a dropdown arrow and a close button (X). Below this is a field with a summation symbol ( $\Sigma$ ) and a formula button ( $f_x$ ). The bottom component is a 'Data Preview' window. It contains a table with two columns: 'PRODUCTLINE' and 'TotalOrders'. The table lists four product lines: 'Classic Cars', 'Motorcycles', 'Planes', and 'Ships', with their respective 'TotalOrders' values. To the right of the table are three buttons: 'Filters...', 'Parameters...', and 'Data Binding...'. An arrow points from the 'TotalOrders' column header in the table to the 'Series 1' field in the 'Slice Size Definition' dialog.

**Slice Size Definition:**

Series 1

$\Sigma$   $f_x$

**Data Preview**

Use the right-click menu or drag the column into series fields.

☒ Show data preview

PRODUCTLINE	TotalOrders
Classic Cars	35,582
Motorcycles	12,778
Planes	11,872
Ships	8,532

Filters...  
Parameters...  
Data Binding...



# Regroupement

---

Les données des axes peuvent être regroupées et triées

- Le tri doit indiquer le sens (ou le champ de tri si différent de la valeur), la locale et la force
- Le groupe doit indiquer le type de donnée, l'unité, l'intervalle (nombre de lignes à regrouper), et le calcul d'agrégation à effectuer

# Tri et groupe



The image shows a 'Group and sorting' dialog box with two main sections: 'Sorting' and 'Grouping'. The 'Sorting' section includes 'Data Sorting' (Ascending), 'Sort On' (row['ORDERDATE']), 'Locale' (Auto), and 'Strength' (TERTIARY). The 'Grouping' section includes a checked 'Enabled' checkbox, 'Type' (DateTime), 'Unit' (Months), 'Interval' (1), and 'Aggregate Expression' (Sum). At the bottom are buttons for '?', 'OK', and 'Cancel'.

Section	Property	Value
Sorting	Data Sorting	Ascending
	Sort On	row['ORDERDATE']
	Locale	Auto
	Strength	TERTIARY
Grouping	Enabled	<input checked="" type="checkbox"/>
	Type	DateTime
	Unit	Months
	Interval	1
	Aggregate Expression	Sum

Enable grouping

Group by month

Show every month on x-axis

Sum monthly data





# Regroupement / axes

---

Le regroupement de données a des conséquences différentes en fonction des axes :

- Le regroupement de catégorie (axe-x) permet d'agréger des valeurs
- Le regroupement de valeurs (axe-y) permet de différencier des valeurs (*Optional Y grouping*) de la même façon que les séries



# Regroupement de catégorie de type date

---

Les valeurs de type date peuvent être regroupées par secondes, minutes, heures, jours, semaines, mois et année

L'option *intervall* permet d'effectuer un second regroupement

- Par exemple, pour grouper par trimestre choisir un groupement par mois et un intervalle de 3



# Regroupement de catégorie de type texte

---

Les valeurs de type texte ne peuvent être regroupées que par l'option intervalle.

L'intervalle donne alors le nombre de valeurs regroupées



# Regroupement catégorie de type numérique

---

Les valeurs numériques sont regroupées via l'option intervalle

- Un intervalle de 10 regroupe les valeurs de 1 à 10, puis 11 à 20, etc.
- Si il n'existe pas de valeurs pour un intervalle, celui-ci n'est pas affiché



# Regroupement de valeurs

Chart type	Reason for using optional Y grouping
Bar, cone, line, pyramid, tube	To summarize data into multiple sets of risers in the chart.
Area, difference	To summarize data into multiple areas in the chart.
Bubble	To identify bubbles using the legend.
Meter	To plot multiple meters.
Alternate meter subtype	To plot multiple dials.
Pie	To plot multiple pies.
Scatter	To plot multiple ( $x$ - $y$ ) value pairs.
Stock	To plot multiple sets of candlesticks.

# Exemple Barre



Series	Expression
Value (Y) Series	row["QUANTITYORDERED"]
Category (X) Series	row["ORDERDATE"]
Optional Y Series Grouping	row["PRODUCTLINE"]



# Axes-y multiples

---

Il est possible de définir plusieurs axes-y utilisant une échelle différente

L'option ***Multiple Y Axis*** doit alors être positionné à *Secondary Axis* ou *More Axes*

Il faut ensuite spécifier les données pour chaque axe-y



# Combinaison de graphiques

---

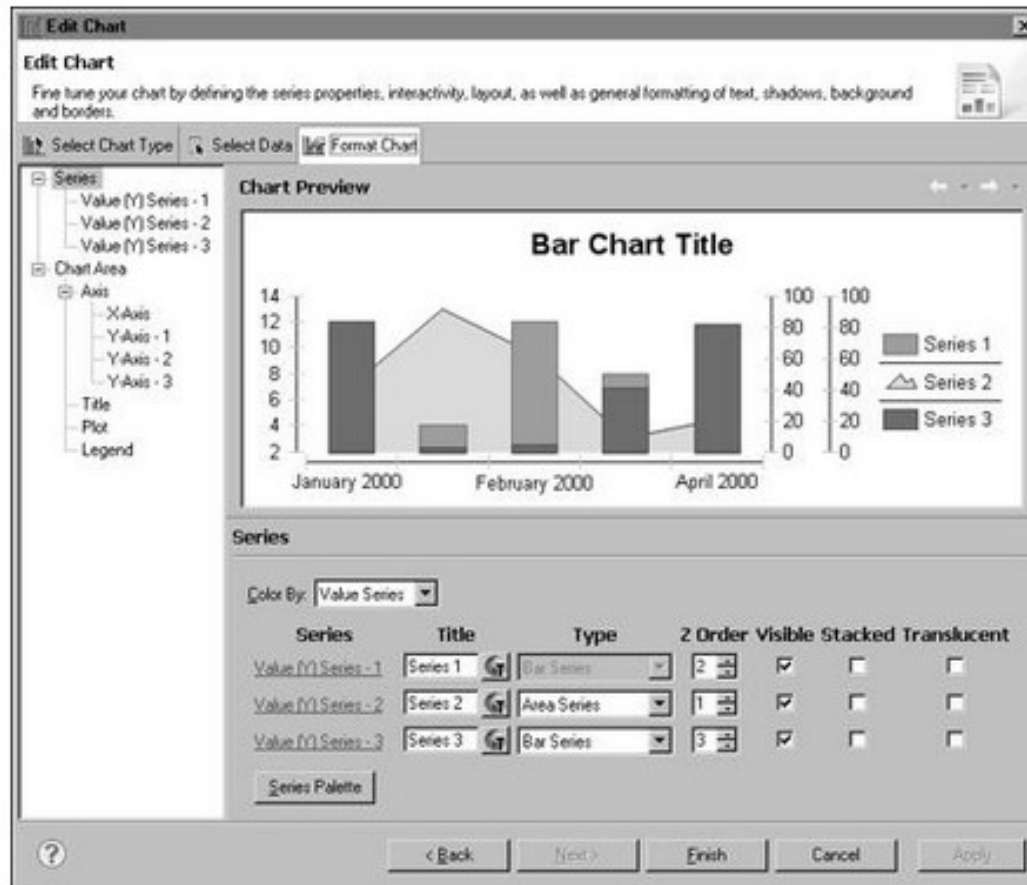
Il est possible de combiner plusieurs types de graphiques sur le même graphique.

Pour cela, il faut définir une série pour chaque type de graphique et associer un graphique différent pour l'une des séries dans l'onglet *Format Chart*

Un index  $z$  est quelque fois nécessaires



# Example





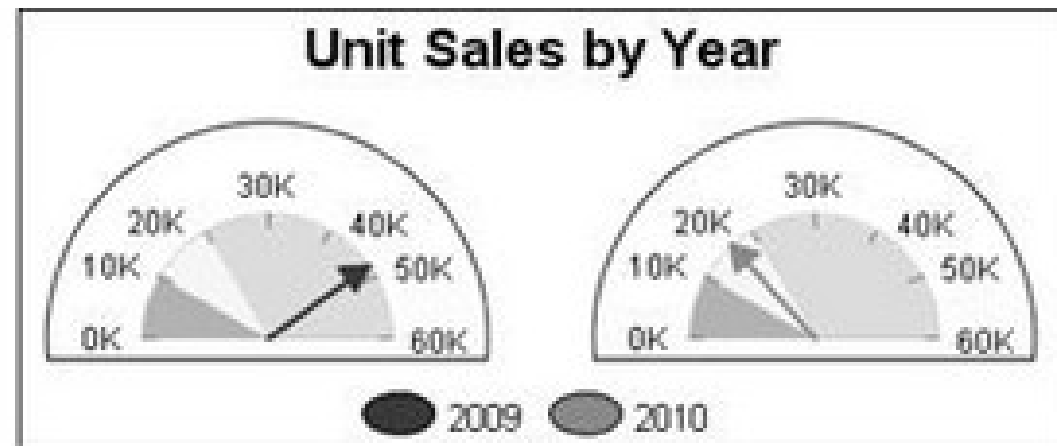
# Graphique mètre

---

Il est possible de créer des graphiques mètres affichant plusieurs aiguilles sur le même cadran ou plusieurs cadrans avec chacun une seule aiguille

Cela se fait en définissant plusieurs séries ou en utilisant le regroupement d'axe-y

# Example





# Prévisualisation

---

L'assistant permet de prévisualiser les données du jeu de données et également (après avoir renseigné les expressions) prévisualiser grossièrement la forme du graphique.

- La prévisualisation utilise soit un sous-ensemble des données du jeu soit des données aléatoires
- Les options de prévisualisation peuvent être configurées dans  
*Windows → Preferences → Report Design → Chart → Field assist*



# Formatage

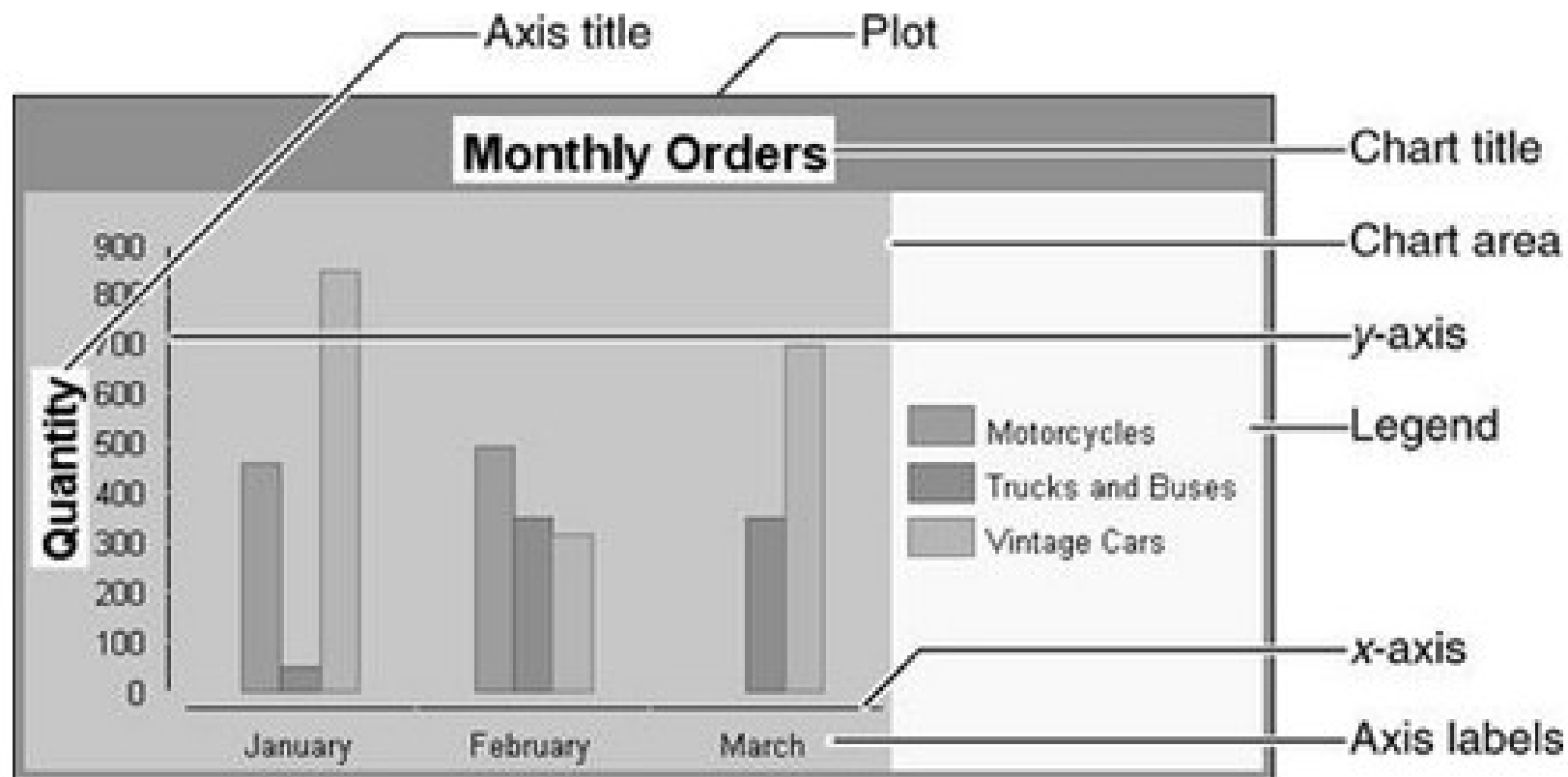
---

De nombreuses options de formatage sont possibles :

- Modification de la zone dédiée au graphique :  
Couleur de fond, padding, ...
- Formatage des valeurs : numériques, dates
- Formatage des libellés du graphique (titres, légendes, étiquettes) : couleur police, taille, ...
- Formatage des axes : Style de ligne, étiquettes, couleurs
- Formatage des séries : forme, couleur, etc
- Ajout d'hyperliens, de la surbrillance



# Zones d'un graphique





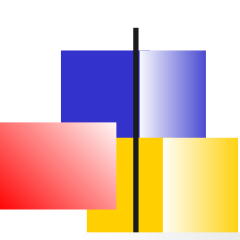
# Interactivité

---

Les graphiques peuvent contenir des liens hypertextes, être définis comme signet ou apparaître dans la table des matières comme d'autres éléments de rapport.

Certains graphiques apportent en plus d'autres types d'interactivité associés à des éléments du graphique (légende, titre, axe)

*Format Chart → <Chart element> → Interactivity*



# Spécification de l'interactivité

---

Il faut alors définir :

- le type d'événement déclenchant l'interactivité
- L'action résultante

Toutes les actions ne sont pas disponibles en fonction des formats.  
Seul le format SVG supporte toutes les actions proposées



# Example





# Événements

---

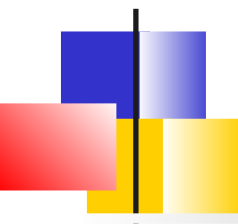
Event type	Description
Mouse Click	Click the selected chart element.
Mouse Double-Click	Double-click the selected element.
Mouse Down	Press and hold the mouse button down over the selected element.
Mouse Up	Release the mouse button above the selected chart element.
Mouse Over	Move the mouse pointer onto the selected element and leave it there.
Mouse Move	Pass the mouse pointer over the selected element.
Mouse Out	Move the mouse pointer off the selected element.
Focus	Put UI focus on the selected element with the mouse or tab navigation.
Blur	Remove UI focus from the selected element using either the mouse or tab navigation.
Key Press	Press a key while the mouse pointer is over the selected element.
Key Down	Press and hold a key down while the mouse pointer is over the selected element.
Key Up	Release a key while the mouse pointer is over the selected element.
Load	Load the chart in the viewer.



# Actions

---

Action name	Result
Hyperlink	Links to a web page, a document, or an image. Also used to link to another report.
Invoke Script	Invokes a client-side script inside the viewer.
Show Tooltip	Displays explanatory text over a chart element.
Toggle Visibility	Changes the visibility of a chart element, typically a series.
Highlight	Highlights a chart element, such as a data point.



# Tableaux croisés



# Tableau croisé

---

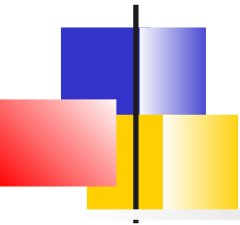
- Les tableaux croisés sont des tableaux spécifiques dans lesquels les nombres de lignes et de colonnes ne sont pas connus à l'avance. Par exemple, les ventes des différents produits par années
- Ni le nombre d'années, ni le nombre de produits ne sont connus au moment du design
- Ils sont utilisés pour afficher des données agrégées avec de multiples niveaux de groupements de colonnes et de lignes
- Les calculs courants sont des totaux, des pourcentages, des moyennes



# Tableau croisé

---

	2007	2008	2009
Fraises	45	21	39
Cerises	40	25	42
Pommes	35	36	38

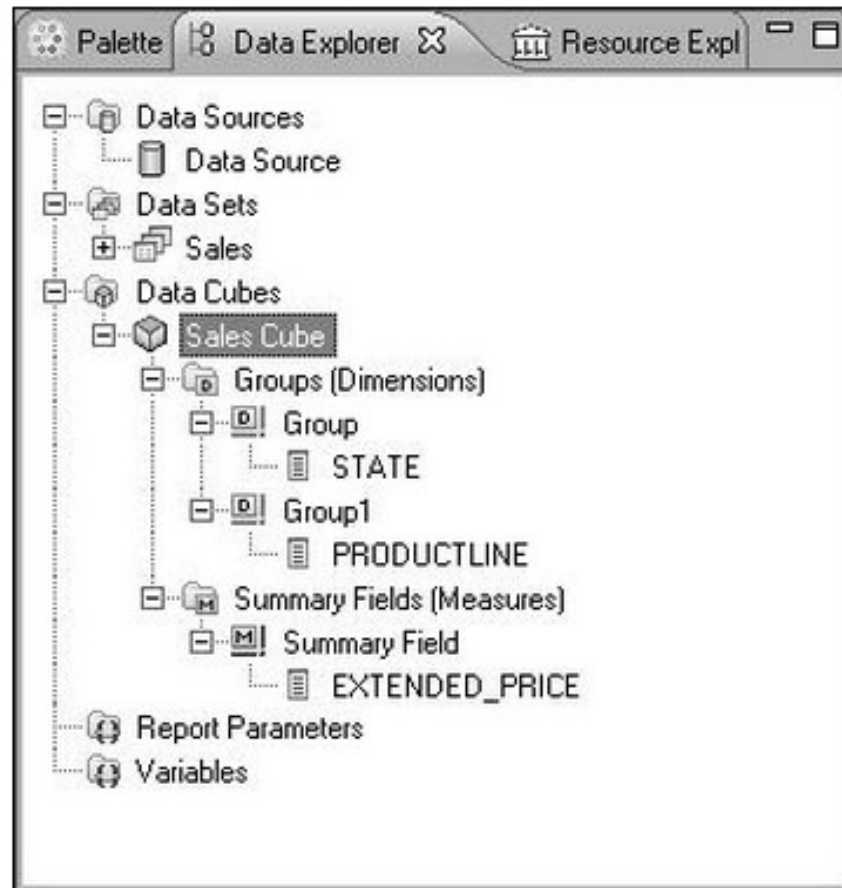


# Cubes

Les tableaux croisés utilisent des jeux de données spécifiques appelés cube (OLAP)

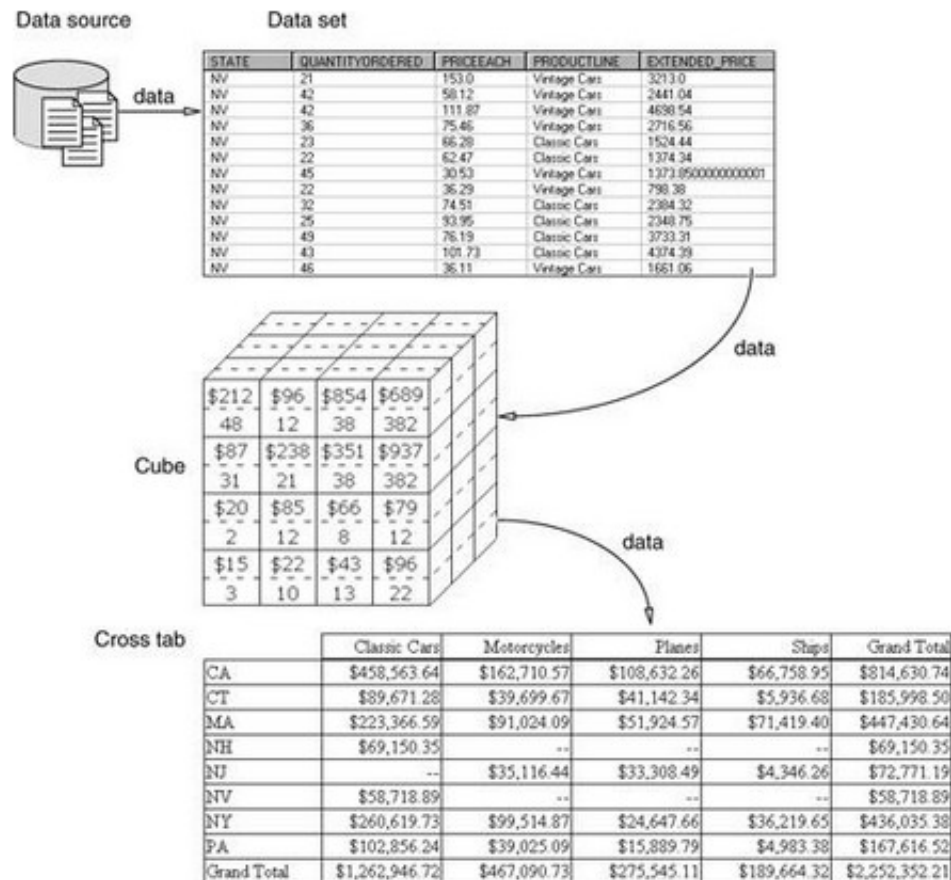
- Un cube agrège une ou plusieurs mesures ou champ résumé selon différentes dimensions (x,y) ou regroupements
- Les dimensions peuvent être hiérarchiques

# Exemple





# Données





# Construction des cubes

---

Les cubes peuvent être construits à partir d'un unique jeu de données ou de plusieurs jeux

- Dans le cas de plusieurs jeux de données, le premier jeu de données doit contenir la mesure



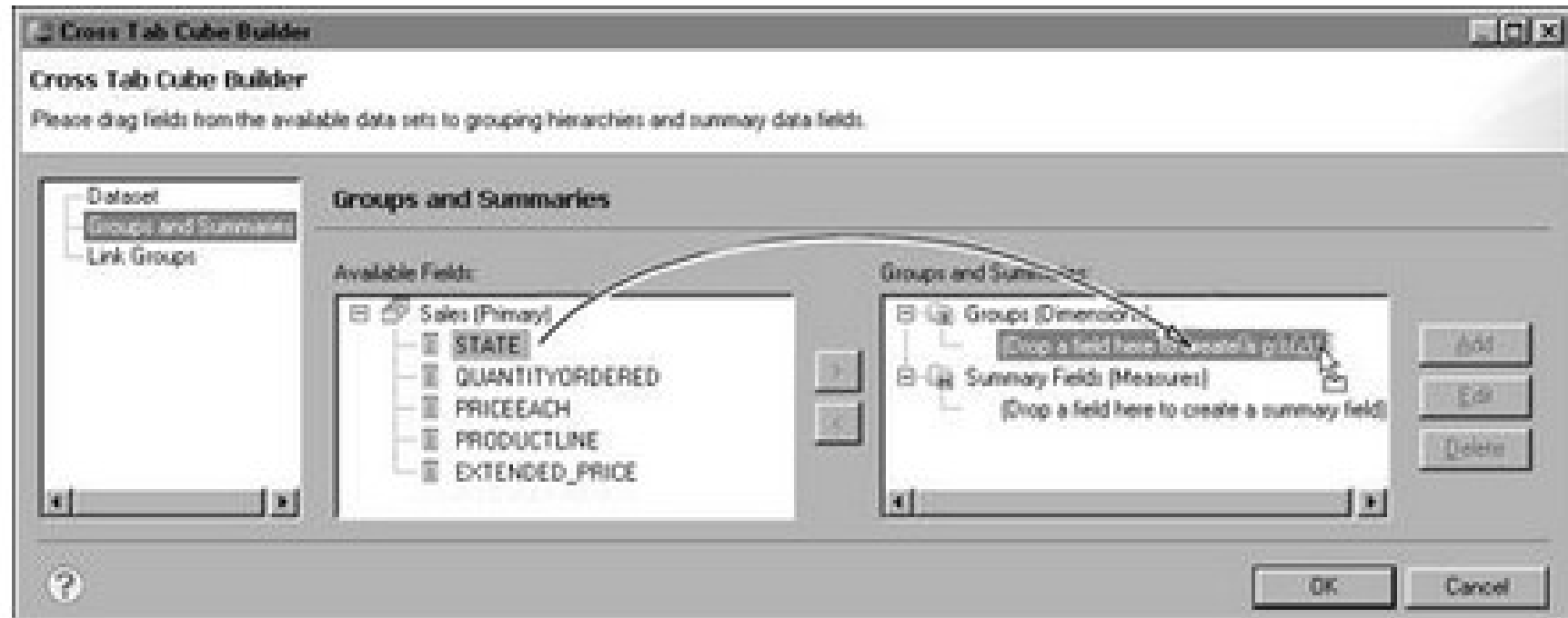
# Un seul jeu de données

---

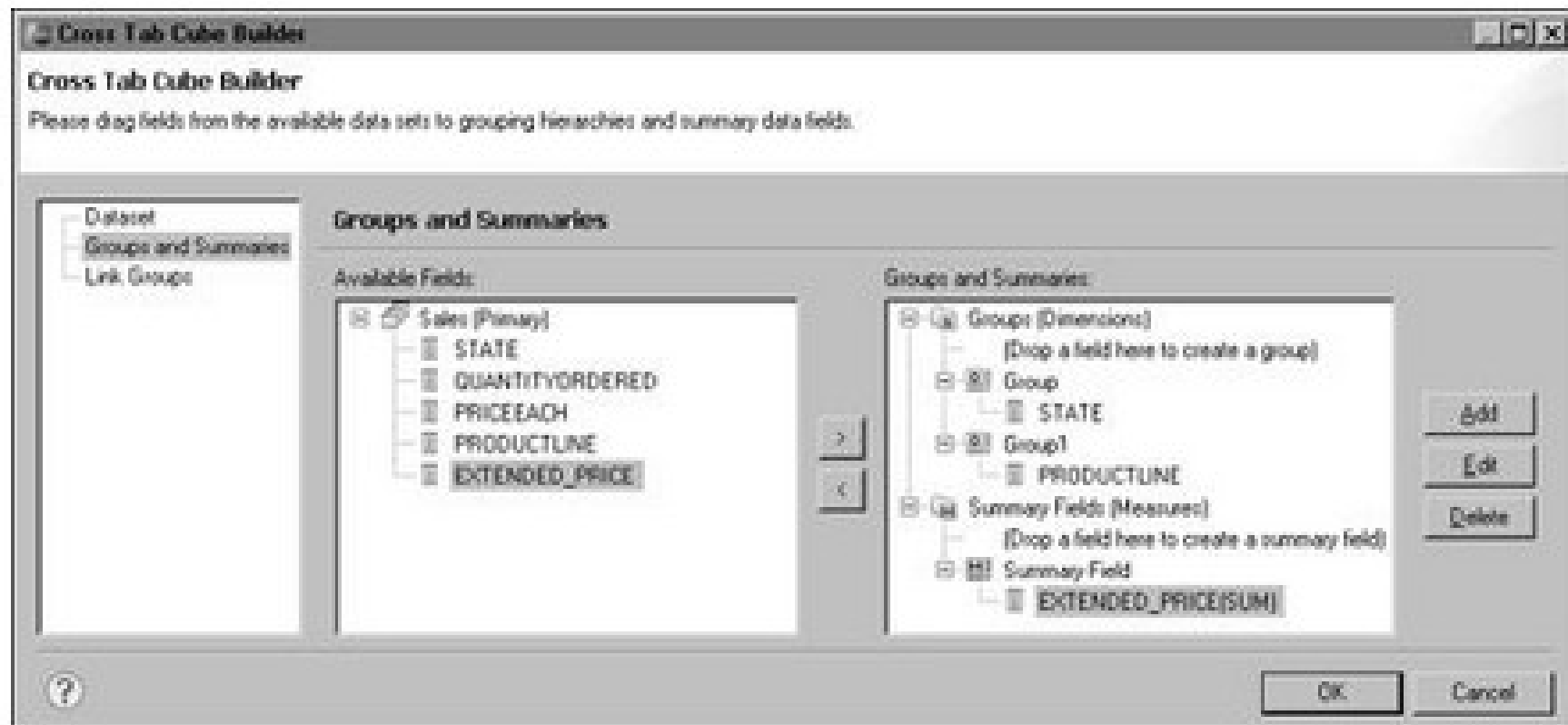
Dans ce cas, on sélectionne le jeu de données comme jeu de donnée primaire

Ensuite, les groupes et mesures (summaries) sont créés en glissant/déposant les champs adéquats

# Ajout d'un groupe



# Ajout d'une mesure





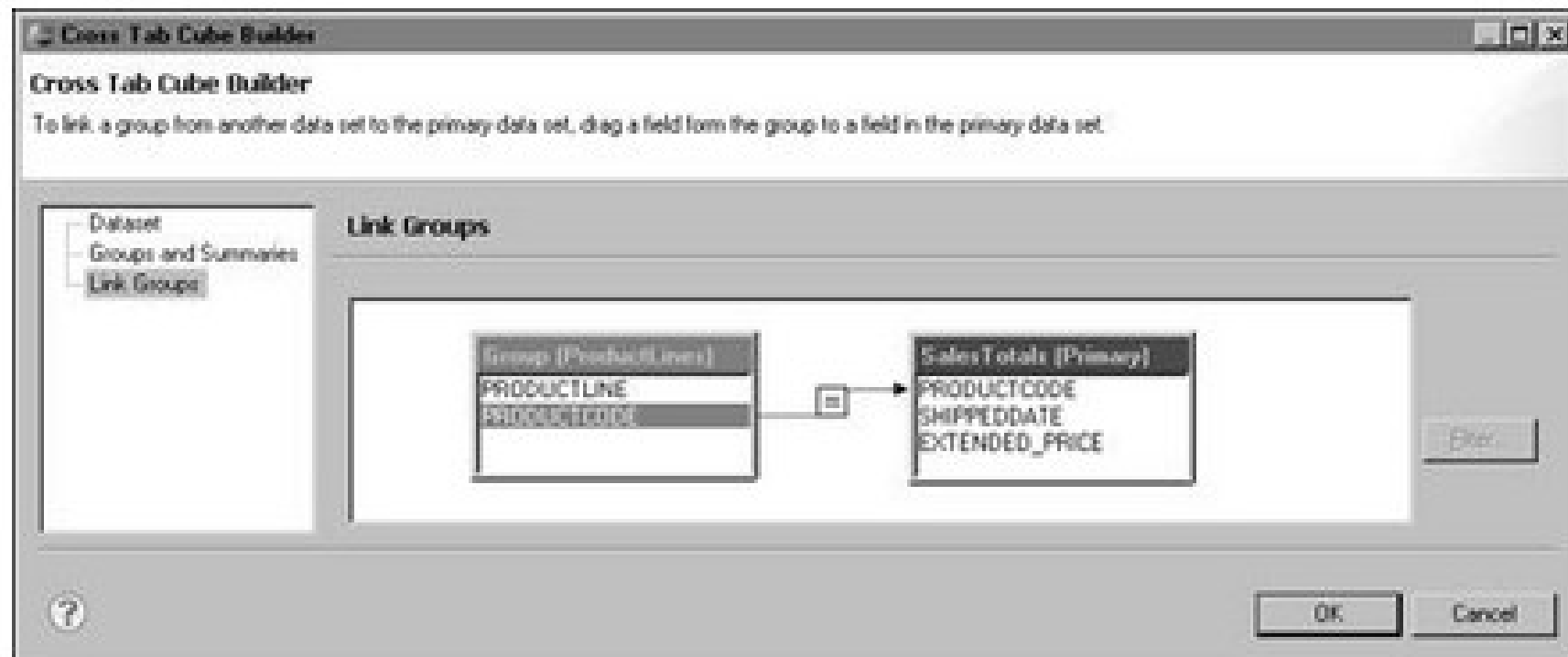
# Plusieurs jeux de données

---

La procédure est alors identique et les champs pour les dimensions peuvent être choisis des différents jeux de données

Une étape supplémentaire liant les dimensions avec le jeu de données primaires est nécessaire

# Lien des dimensions





# Zones d'un tableau croisé

---

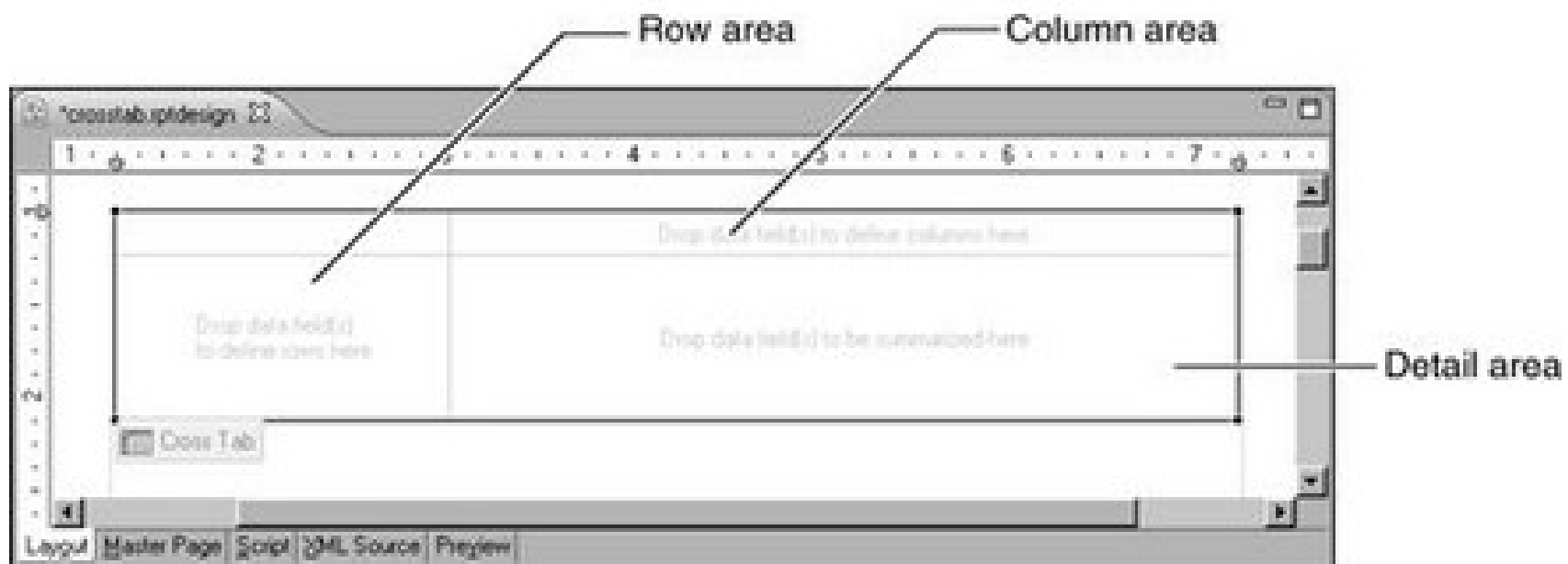
Un tableau croisé contient 3 zones :

- La zone **ligne** destiné à afficher des dimensions en abscisse
- La zone **colonne** destinée à afficher les dimensions en ordonnée
- La zone **détail** destinée à afficher les mesures





# Zones





# Remplissage du tableau

---

Le renseignement des zones se fait par glisser/déposer des données du cube dans la zone

- Il est possible de positionner plusieurs dimensions dans une ligne et colonne ; l'ordre de position est alors important
- Il est également possible d'afficher plusieurs mesures



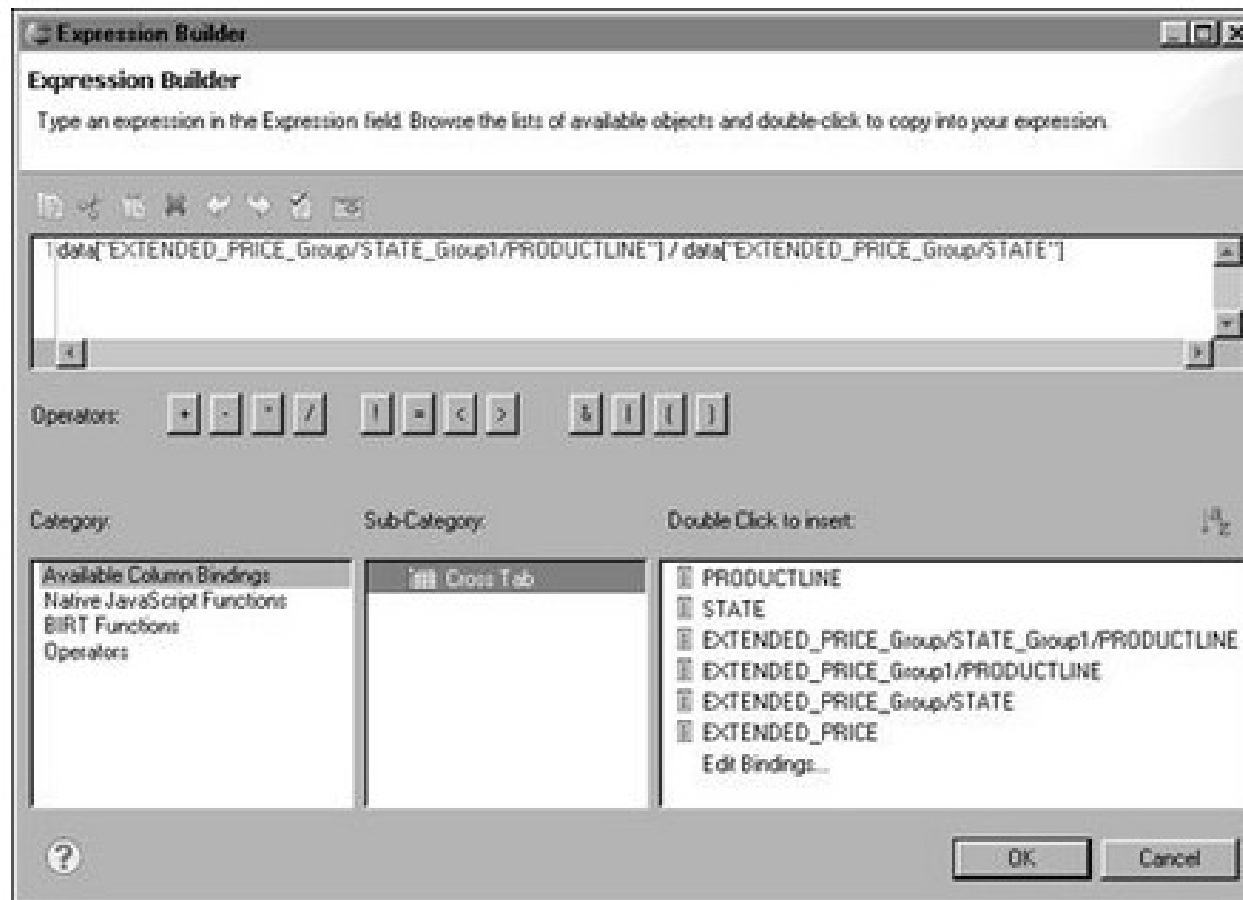
# Mesure dérivée

---

Le tableau croisé fournit un ensemble de variables qui peuvent être utilisées pour calculer des mesures dérivées

- Les variables fournis le tableau croisé sont typiquement les totaux des mesures par toutes les combinaisons de dimensions possibles
- Une mesure dérivée classique est un calcul de pourcentage par rapport à un total

# Exemple





# Ajout des sous-totaux et grand total

---

Il est possible d'afficher les totaux pour chaque combinaison de dimension

Par défaut, BIRT ne les affiche pas. Il faut explicitement les spécifier dans l'éditeur de propriété et les onglets *Row Area* et *Column Area*

- Pour définir un sous-total, il faut fournir la dimension d'agrégation
- Les totaux peuvent également être affichés sous forme de graphiques



# Gestion des lignes ou colonnes vides

---

Par défaut, BIRT n'affiche pas les lignes ou colonnes n'ayant pas de mesures

Il est possible de forcer l'affichage



# Filtre et Tri des données

---

Par défaut, BIRT trie les données par les valeurs des dimensions dans l'ordre ascendant

- Il est possible de modifier la direction de l'ordre et même le critère d'ordre qui peut être une mesure ou un total par exemple
- Il est également possible de définir un filtre au niveau du tableau croisé

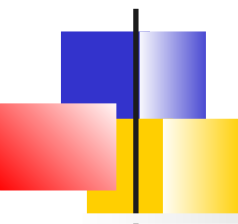


# Pour aller plus loin

---

Localisation  
Développement collaboratif





# Localisation



# Introduction

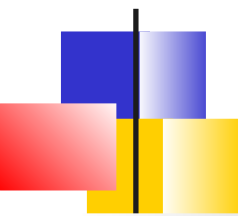
---

Lorsqu'il est nécessaire de générer le rapport en différentes langues.

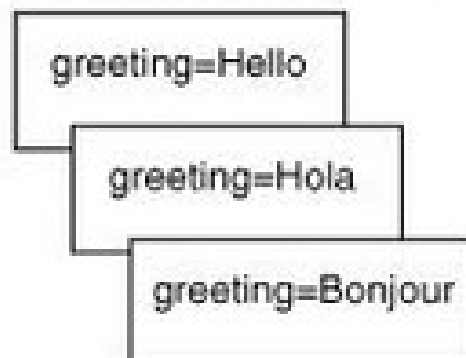
Les chaînes statiques dans le rapport.

A la place, sont utilisées des clés qui font référence à des fichiers de traduction externes au rapport

Lors de la génération, BIRT utilise la locale de la machine pour utiliser la bonne traduction



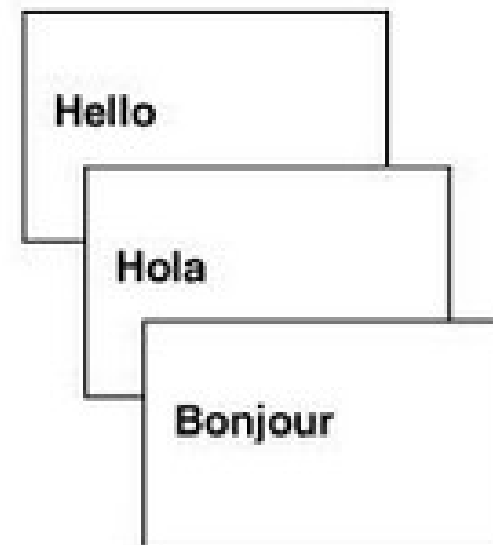
**Resource files** for English, Spanish, and French locales. Each file contains the resource key, greeting, and the localized version.



**Report design** uses the resource key, greeting, in a label element.



**Report output** when run in English, Spanish, and French locales, respectively.





# Éléments localisables

---

Les éléments pouvant être localisés et acceptant une clé de ressources sont :

- Les textes statiques, labels du rapport mais également des graphiques
- Les noms d'affichage des des champs du jeu de données
- Les valeurs textes venant d'un jeu de données
- Les textes d'aides associés aux paramètres



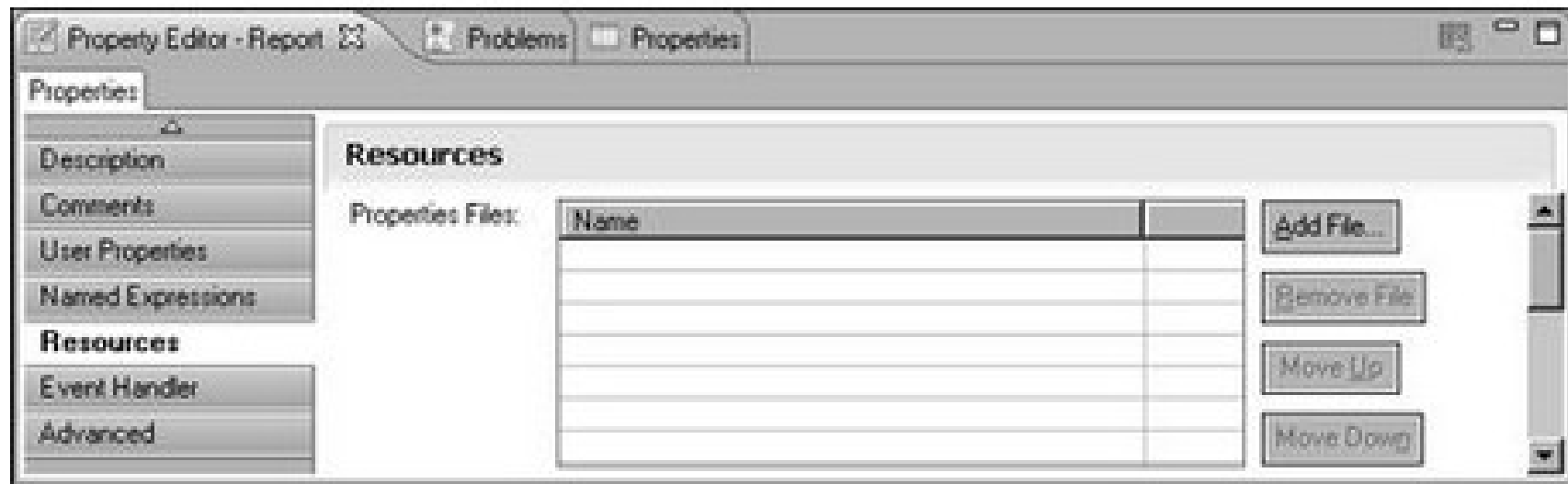
# Étapes pour la localisation

---

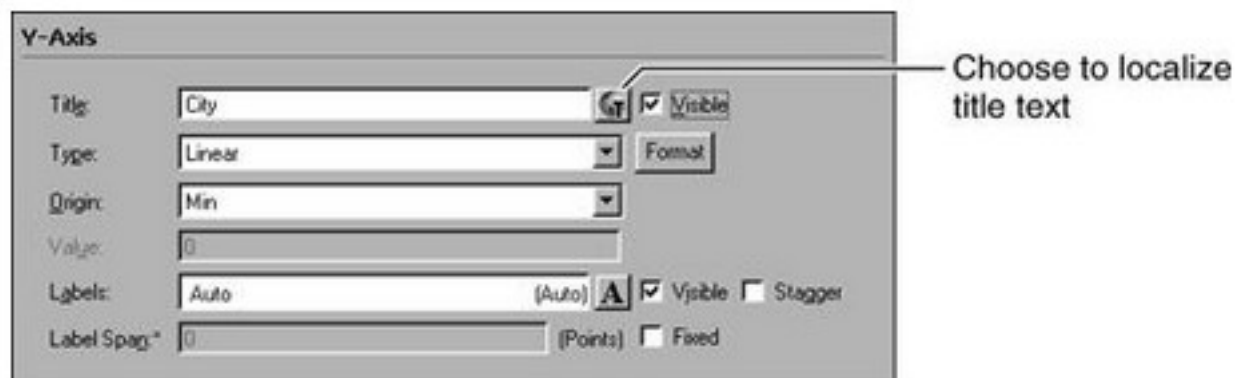
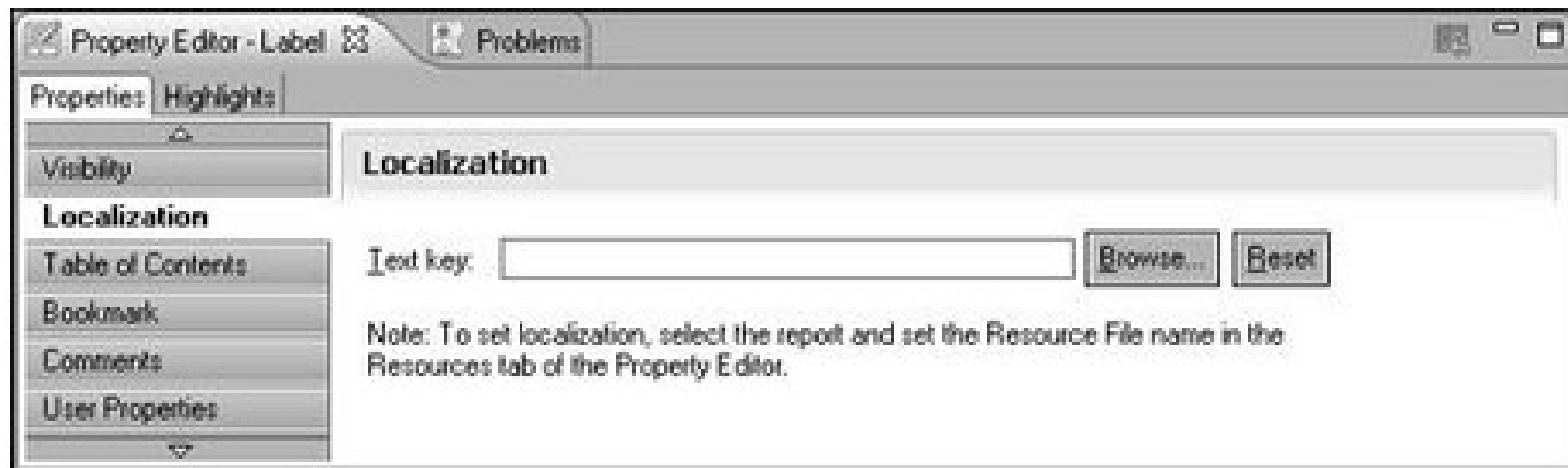
Pour localiser un rapport, il est nécessaire de :

- Créer les fichiers ressources : Ce sont des fichiers définissant une clé par ligne et respectant le nommage suivant :  
*<bundlename>\_<langue\_iso\_code>\_<pays\_iso\_code>.properties*
- Les placer dans le répertoire ressource
- Affecter le bundle au rapport
- Utiliser les clés dans les labels

# Affectation du bundle



# Affectation de clé



# Affectation de clé



A screenshot of a 'Select Key' dialog box. It features a table with two columns: 'Key' and 'Value'. The table lists several key-value pairs. Below the table is a 'Quick Add' section with input fields for 'Key' and 'Value', and 'Add' and 'Delete' buttons. At the bottom, there is a help icon, 'OK', and 'Cancel' buttons.

Key ^	Value
contact name	Contact
customer address	Customer Address
customer ID	Customer ID
customer name	Customer
order date	Order Date
order ID	Order Number
sales rep	Sales Representative

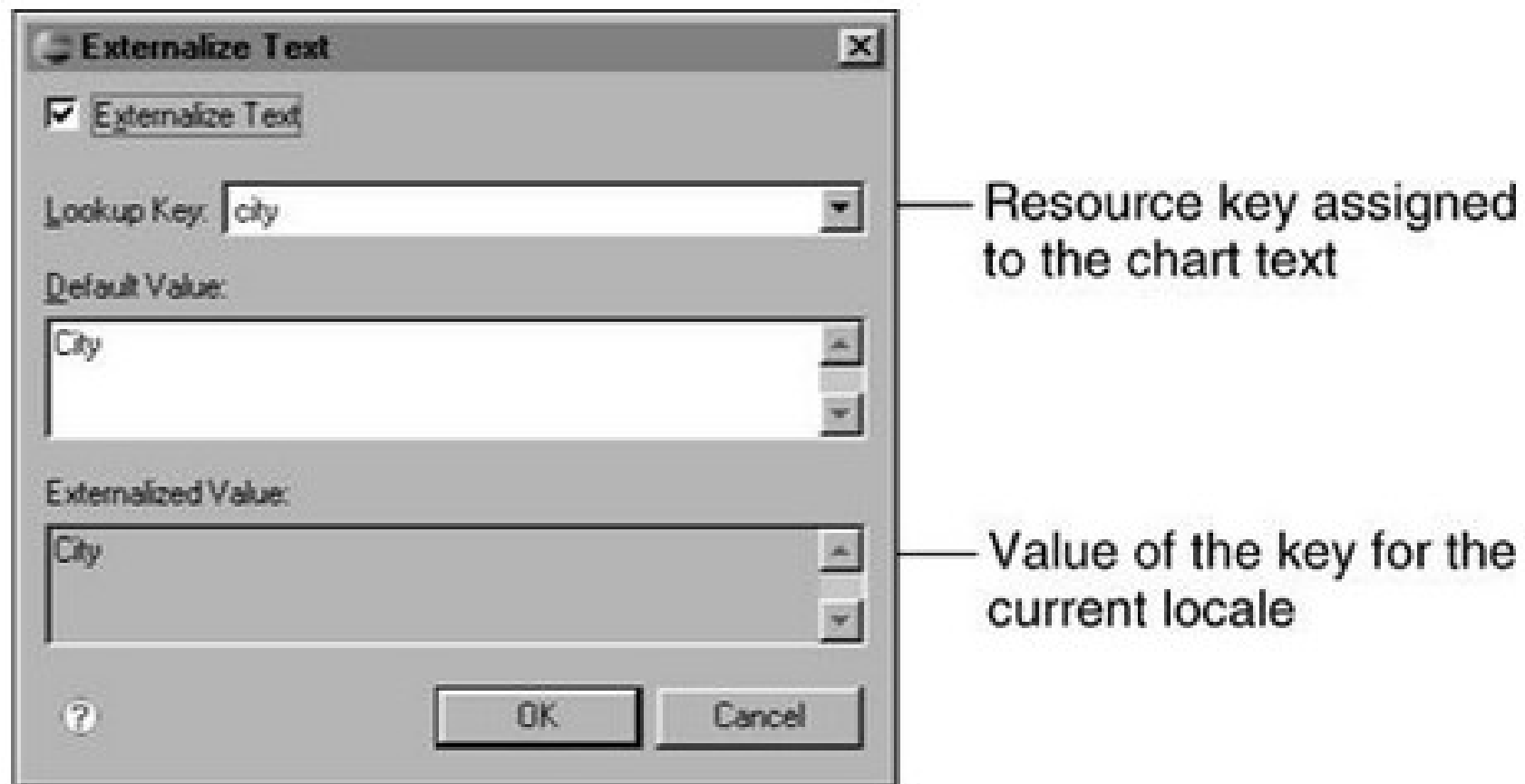
Quick Add

Key  Value

Quick Add will save the key to the current resource file.



# Affectation de clé (graphique)



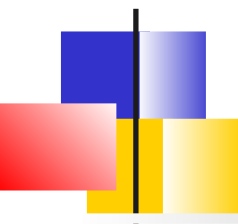


# Sélection de locale

---

Pour tester le rapport dans les différentes langues, il suffit de modifier la locale dans les préférences de prévisualisation.

*Window → Preferences → Preview*



# Développement collaboratif



# Introduction

---

BIRT facilite le développement collaboratif grâce à plusieurs types de ressources :

- Les **bibliothèques** fournissent un repository d'éléments de rapport (source de données, jeux de données, styles, etc.).  
L'extension des fichiers est **.rptlibrary** et ils sont situés dans le répertoire *resource* partagé par les développeurs
- Les **gabarits** (templates) fournissent une modèle de départ pour créer un rapport.  
L'extension est **.rpttemplate** et les fichiers sont situés dans le répertoire *template*. BIRT fournit des gabarits standards
- Les fichiers **CSS** qui mutualisent des informations de style

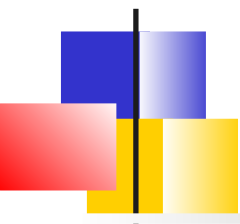


# SCM

---

BIRT bénéficie des capacités de **gestion de projet d'Eclipse** pour organiser les rapports, en particulier il peut facilement se connecter à un SCM (CVS/SVN/Git) pour la gestion des sources

Les éléments mutualisés (bibliothèque, gabarits, css) et les rapports peuvent donc être commités et mis à jour via un SCM



# Bibliothèques et thèmes



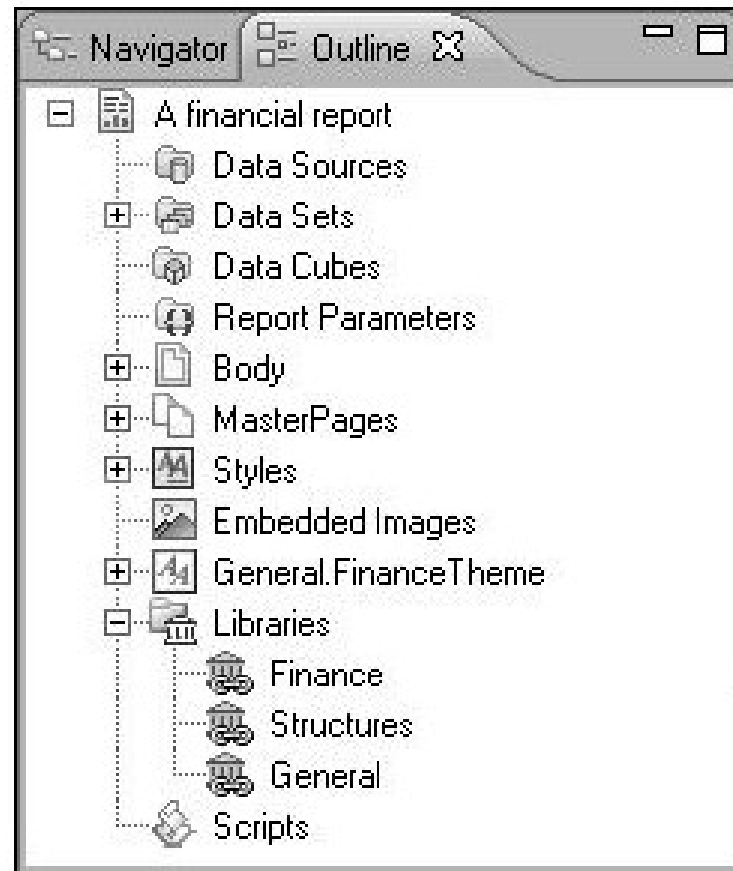
# Bibliothèque

---

Une bibliothèque est un composant **dynamique** du rapport.

- Lorsqu'un développeur effectue un changement dans la bibliothèque, il est répercuté dans tous les rapports qui utilisent la bibliothèque
- Une bibliothèque stocke des éléments de rapport comme des sources de données, des jeux de données, des pages maître, des styles, des éléments simples ...
- Un rapport peut utiliser 0 ou plusieurs bibliothèques

# Rapport utilisant plusieurs bibliothèques







# Création

---

BIRT propose 3 techniques pour créer une bibliothèque :

- A partir d'un rapport :

  - Outline View → Clic-droit sur le rapport → Export to Library*

- Créer une bibliothèque vide :

  - File → New → Library*

- A partir d'un élément du rapport :

  - Outline View → Clic-droit sur l'élément → Export to Library*

Une fois créée, des éléments peuvent y être ajoutés ou édités avec les mêmes éditeurs/assistants que ceux d'un rapport



# Édition de la bibliothèque

---

L'édition des éléments est identique à l'édition d'éléments dans un rapport.

Sont disponible :

- L'explorateur de données
- La palette permettant d'ajouter des éléments visuels
- L'éditeur de propriétés



# Thèmes

---

Une bibliothèque permet de regrouper des styles dans des thèmes.

- Chaque thème intègre tous les styles utilisés dans un rapport et un rapport ne peut utiliser qu'un seul thème
- Dans une nouvelle bibliothèque, il existe déjà un thème nommé *defaultTheme*
- On peut créer plusieurs thèmes dans une bibliothèque
- Les thèmes apportent les même fonctionnalités que les fichiers css mutualisés



# Priorité des styles

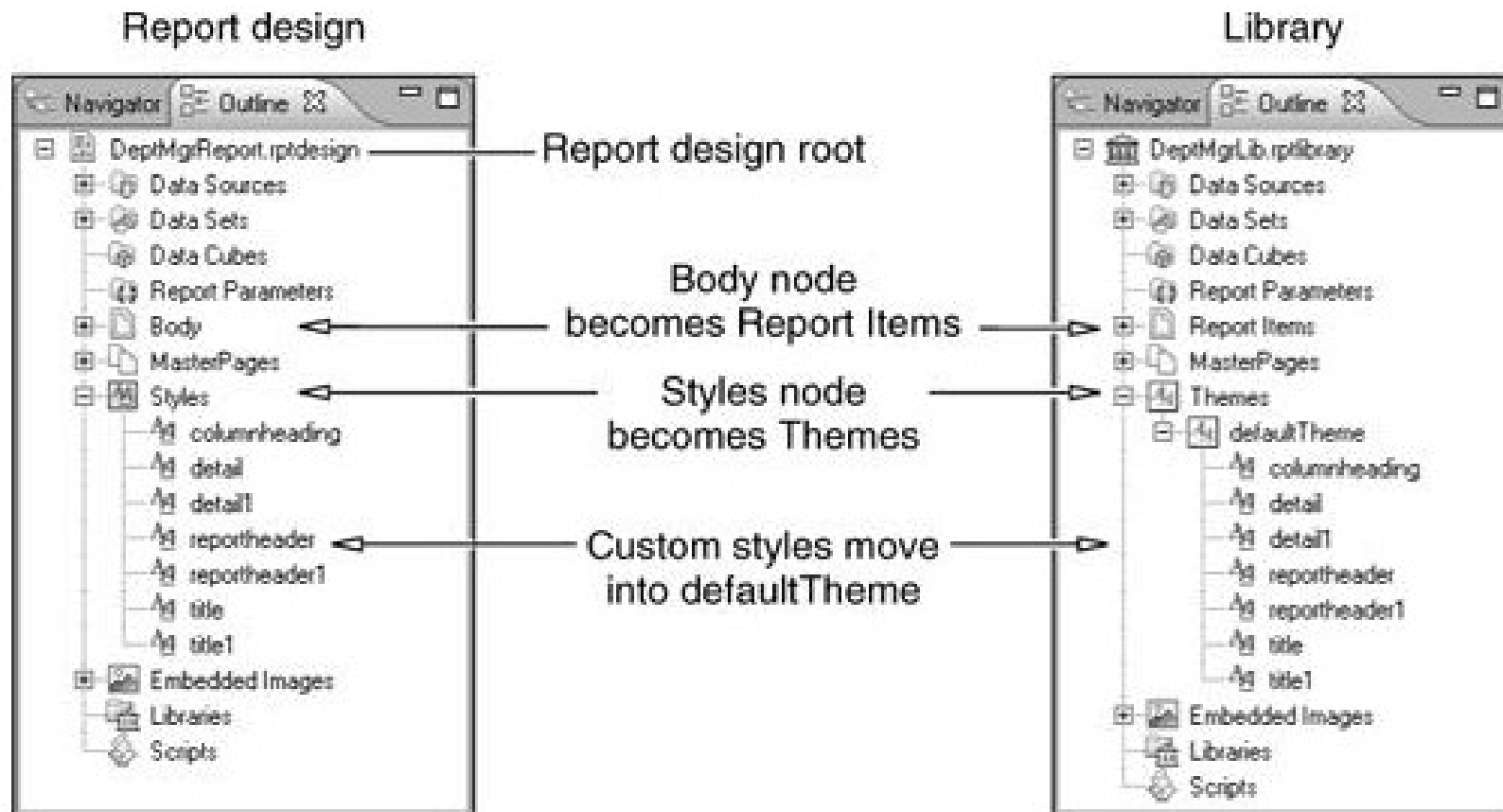
---

Finally, a report can use styles coming from a theme, a css file or the report itself.

In case of identical name, the order of priority is :

- The report
- The css file
- The theme

# Outline d'une bibliothèque





# Partage d'une bibliothèque

*Alternative à un SCM*

---

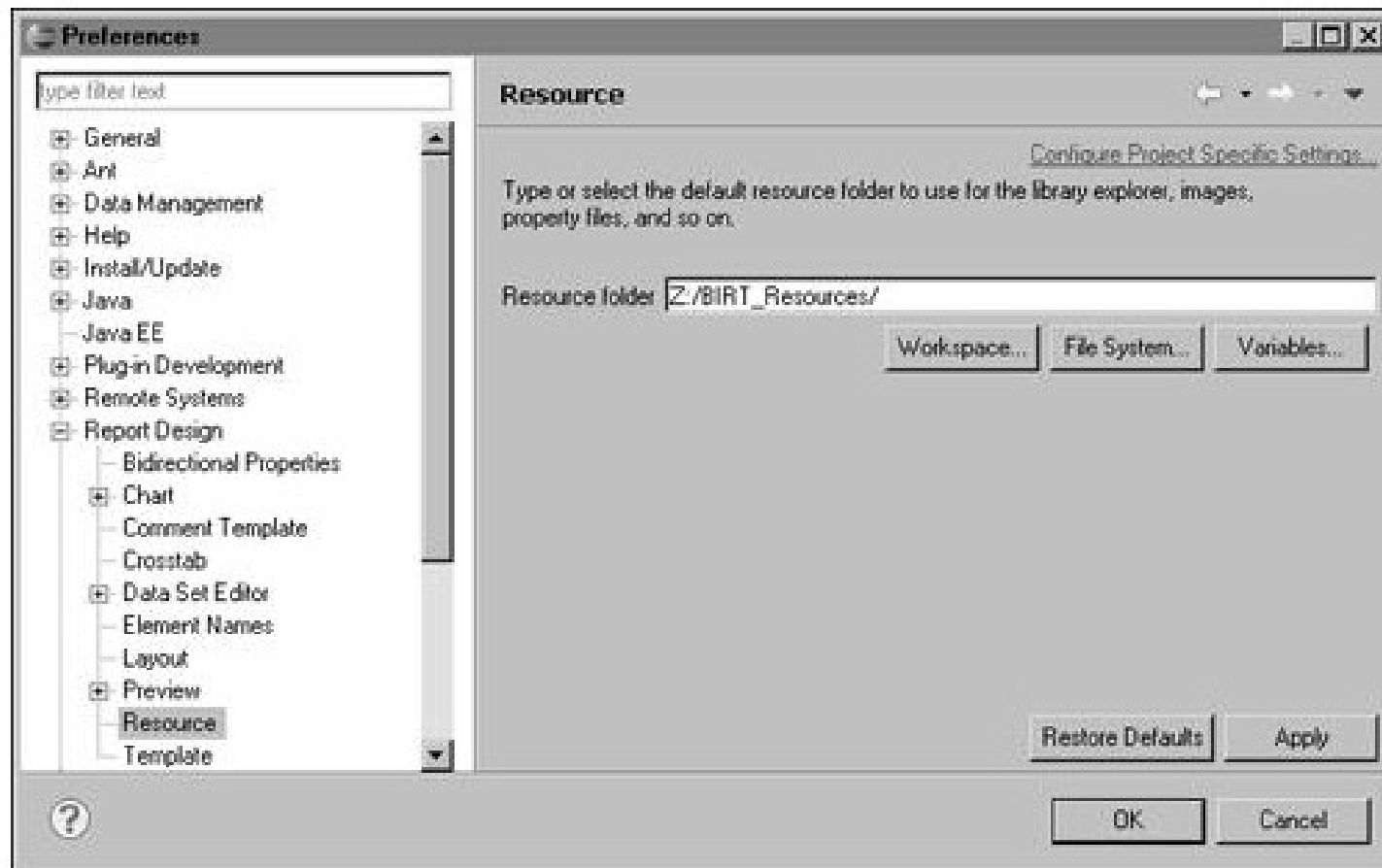
Une fois mise au point, les bibliothèques résidant dans le workspace doivent être publiées dans les ressources partagées

*File → Copy Library to Shared Resource Folder*

Ce dossier doit alors être partagé par l'ensemble des développeurs

*Windows → Preferences → Report Design → Resource*

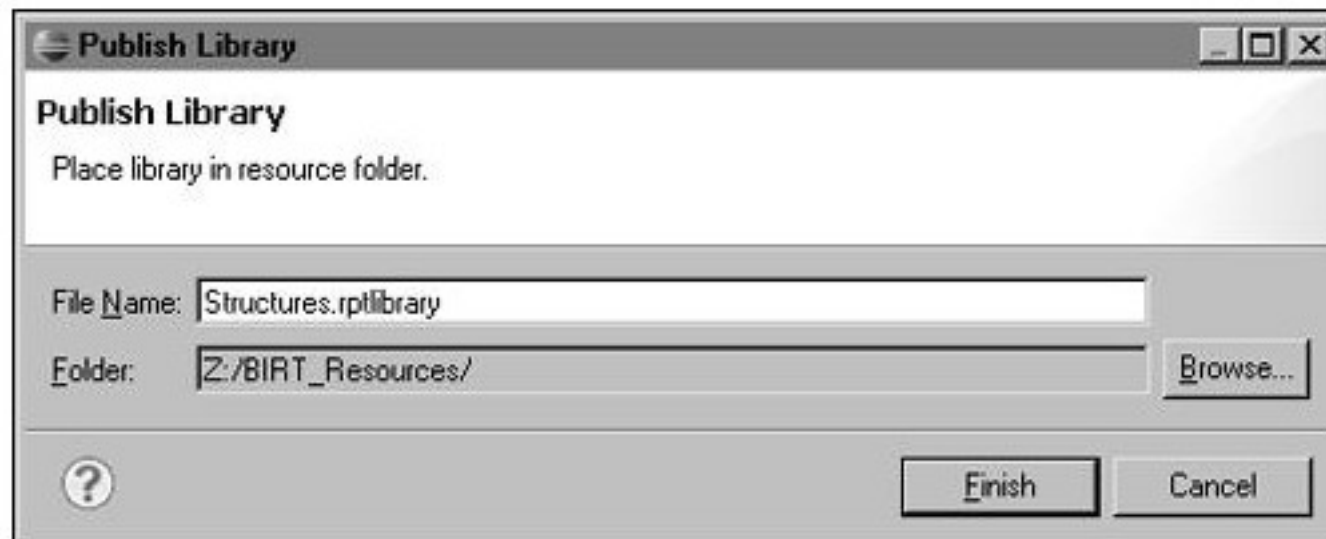
# Emplacement dossier ressources



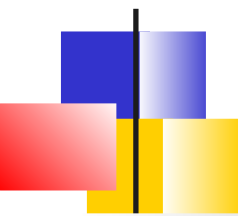


# Publication d'une bibliothèque

Les ressources partagées peuvent être organisées en dossier.  
=> Choisir le bon dossier au moment de la publication








# Utilisation

---

L'utilisation de la bibliothèque se fait via l'onglet ***Resource Explorer*** qui affiche les différentes bibliothèques disponibles et leurs contenus et qui permet d'effectuer des glisser/déposer

- L'onglet doit être rafraîchi manuellement lors de modifications dans le répertoire *ressource*

Il est possible de modifier les propriétés de l'élément provenant de la bibliothèque.  
=> BIRT gère alors les modifications locales effectuées et les propriétés qui restent dynamiques.



# Mises à jour

---

Lorsqu'une bibliothèque est mise à jour, les modifications doivent être publiées dans le répertoire partagé

Les développeurs travaillant avec cette bibliothèque peuvent être obligés de faire un « *Refresh* » pour voir les changements



# Organisation des bibliothèques

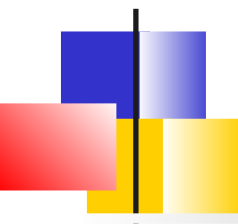
---

Les éléments d'une bibliothèque peuvent également faire référence à une autre bibliothèque

Cela permet d'organiser clairement les bibliothèques des différents projets d'une entreprise.

Par exemple, on pourra fournir :

- Une bibliothèque générale à tous les projets qui contient des thèmes, le logo de la société, des pages maîtres
- Plusieurs bibliothèques dédiés à des projets spécifiques



# Gabarits



# Gabarits

---

Un gabarit est statique.

=> Lors de la création d'un rapport via un gabarit, une copie du gabarit est effectuée

=> Les modifications sur un gabarit n'ont donc pas d'effet sur les rapports les ayant utilisés

Un gabarit fournit une structure pour un rapport et peut contenir tout ce que contient un rapport (source et jeu de données, éléments visuels, page maître, etc.)

Des instructions d'utilisation sont associées au gabarit ainsi qu'à certains de ses éléments qui sont censés être modifiés



# Création de gabarits

---

BIRT propose 2 façons pour créer un gabarit :

- Créer un gabarit vide  
*File → New → Template*
- Créer un gabarit à partir d'un rapport  
*File → Register template with a New Report Wizard*

Un gabarit comporte les propriétés suivantes :

- Un **nom**
- Une **description**
- Une **image**

Ces propriétés sont utilisés par l'assistant BIRT lors de la création d'un nouveau rapport



# Éléments du gabarits

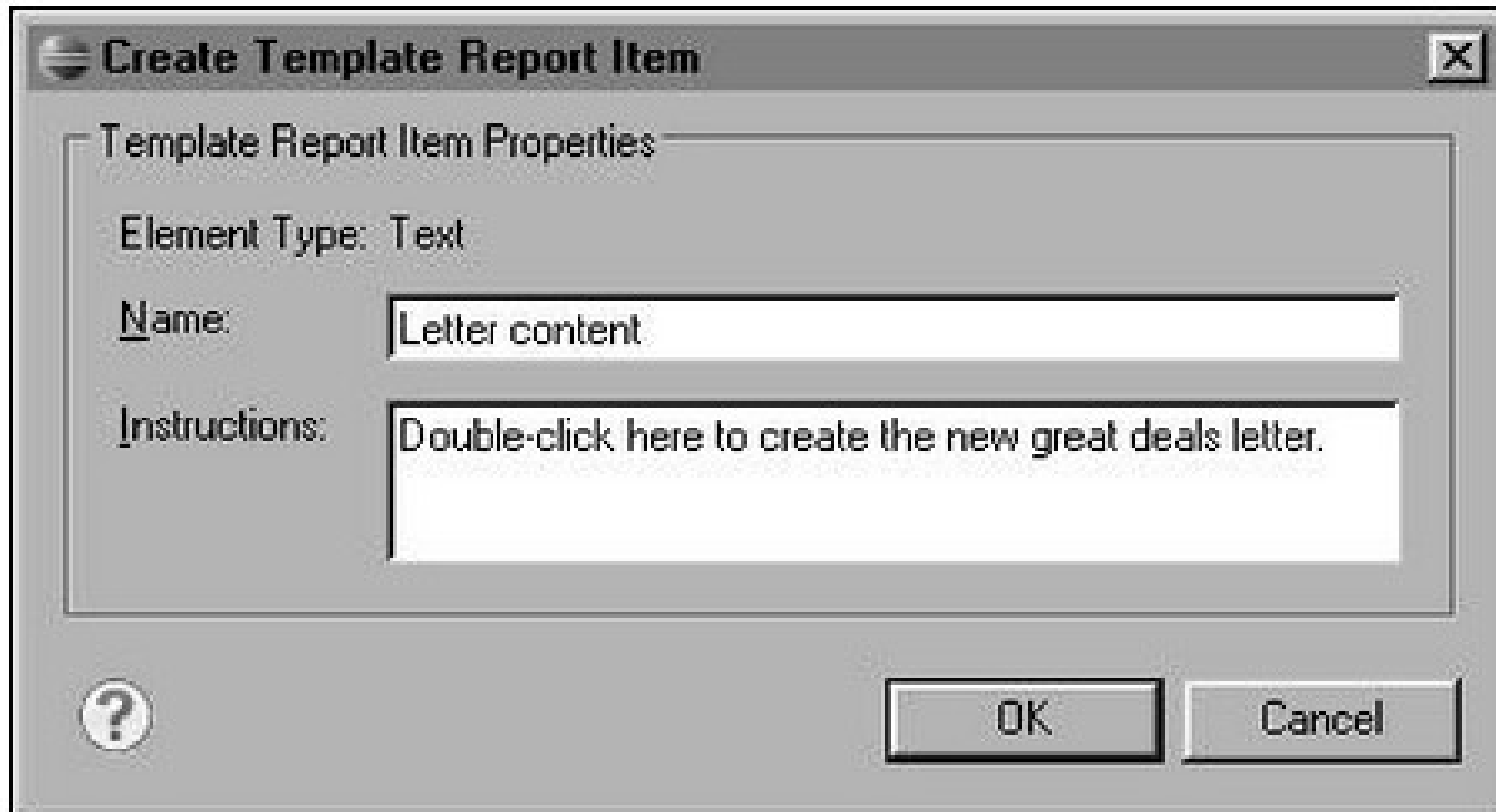
---

Un gabarit contient

- des **éléments standards**
- des **éléments de gabarit** censés être complétés lors de l'utilisation du gabarit  
Les éléments de gabarits ont des instructions associées

Le gabarit dans sa globalité peut également avoir des instructions associées nommées « *Cheat sheet* »

# Instructions associées



**Create Template Report Item**

Template Report Item Properties

Element Type: Text

Name: Letter content

Instructions: Double-click here to create the new great deals letter.

?

OK Cancel





# Partager les gabarits

## *Alternative au SCM*

---

Par défaut, BIRT stocke les gabarits dans le répertoire des gabarits prédéfinis qui n'est pas partagé

L'emplacement des gabarits peut être changé par

*Windows → Preferences → Report Design → Template*

Lorsque le gabarit est prêt, il faut alors le publier via

*File → Register Template with New Report Wizard*



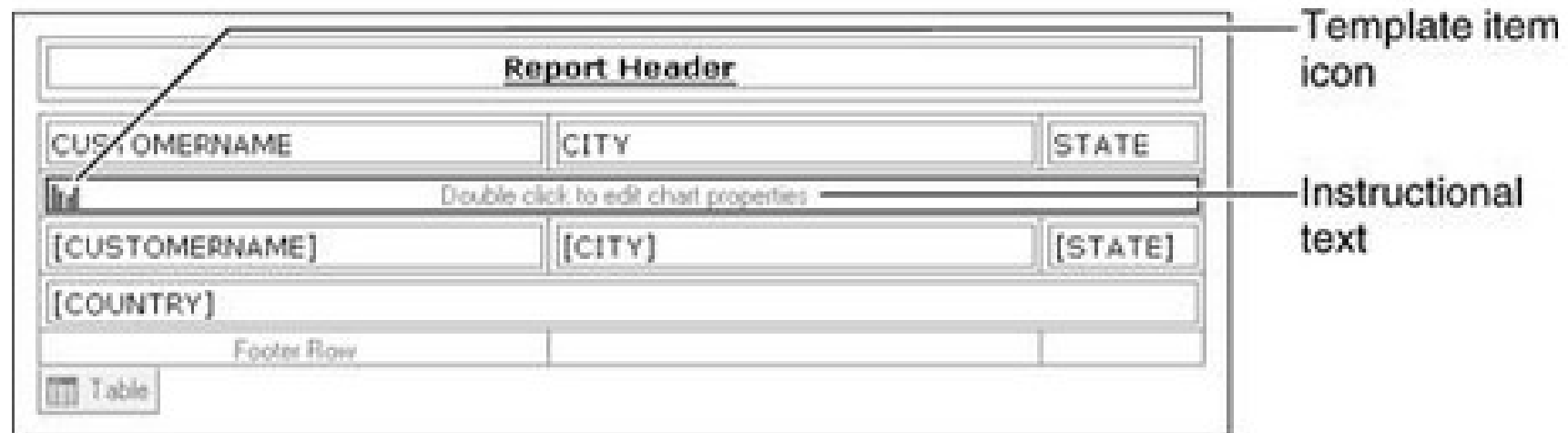
# Utilisation

---

L'utilisation du gabarit est proposée lors de la création d'un nouveau rapport et tous les gabarits du dossier *gabarit* sont proposés

Une fois l'assistant terminé, le rapport hérite de la structure du gabarit et des éléments de gabarits censés être modifiés selon leurs instructions associées

# Élément de gabarit





# Références

---

- « ***BIRT, A field guide*** »,  
Diana Peh, Nola Hague, Jane Tatchell
- Wiki :  
[http://wiki.eclipse.org/index.php/BIRT\\_Project](http://wiki.eclipse.org/index.php/BIRT_Project)
- BIRT Report Object Model (ROM) Documentation :  
<http://www.eclipse.org/birt/phoenix/ref/rom/index.html>
- ROM Spécification :  
[http://www.eclipse.org/birt/phoenix/ref/ROM\\_Overview\\_SPEC.pdf](http://www.eclipse.org/birt/phoenix/ref/ROM_Overview_SPEC.pdf)



# Merci!!!

---

❖ MERCI DE VOTRE ATTENTION