

# ADO.NET / Etats

## Introduction

Crystal Reports est l'outil de création d'états fourni avec Visual Studio .NET. Nous allons voir comment créer très simplement un état alimenté par des données extraites d'une application .NET et comment afficher l'état généré dans une fenêtre Windows.

## 1. Fonctionnement d'un état

Un état peut être considéré comme un "template". Ce modèle va consommer des données brutes et générer un rapport qui contient les données mises en forme. Nous allons extraire des données et nous en servir comme source de données pour notre état Crystal Reports. Le rapport généré sera affiché dans une fenêtre windows via un controle winform spécifique pour les rapports Crystal Reports.

Un état Crystal Reports peut consommer plusieurs types de sources de données .

### I. Utiliser un DataSet fortement typé

#### I-1. Création du dataset fortement typé

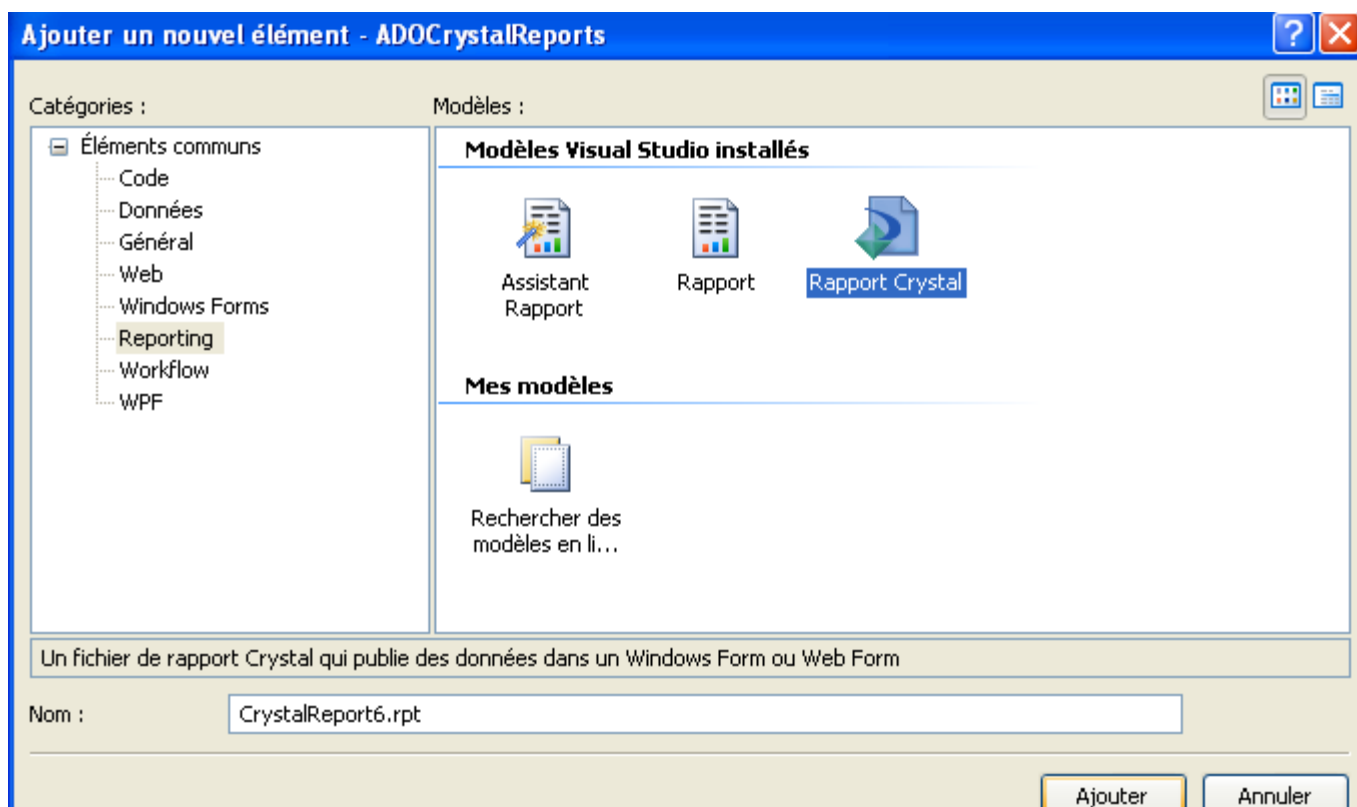
Pour que notre état puisse consommer les données de notre dataset celui-ci doit être fortement typé. En effet, lors de la création de l'état il faut bien que Crystal Reports puisse déterminer le nom et le type des champs pour les données qui lui serviront à générer le rapport. C'est le .xsd d'un dataset fortement typé qui contient toutes ces informations.

Nous allons utiliser la base "TP1" qui contient la table « Employe ». Pour créer rapidement un dataset typé avec Visual Studio .NET :

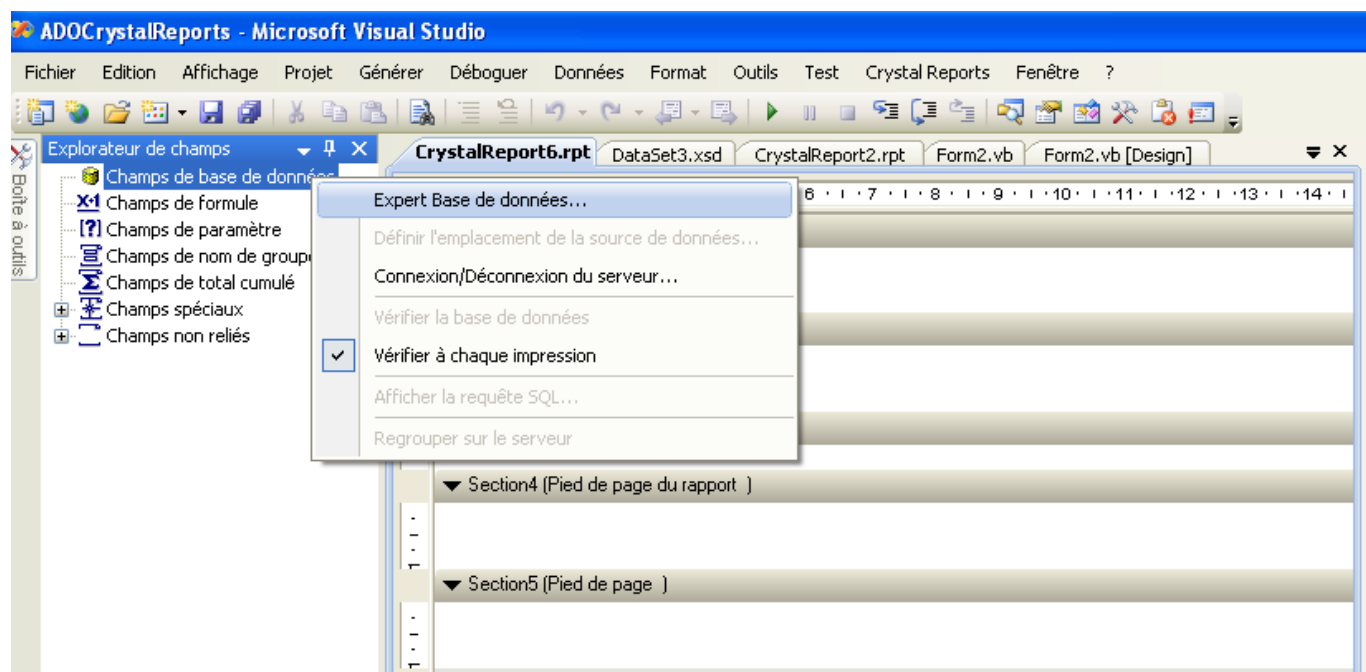
- Ajouter un dataset typé au projet : dans l'explorateur de solution => click droit sur le projet => Ajouter => Ajouter un nouvel élément => Dataset. On l'appellera "dataset1.xsd"
- Créer une nouvelle connexion de données dans la fenêtre "explorateur de serveurs" : clic droit sur connexion de données => ajouter une connexion
- Dans l'explorateur de serveur, développer l'arbre de la connexion à la base TP1 => développer Tables => Glisser/Déposer la table "Employe" dans le concepteur de dataset.

#### I-2. Création de l'état

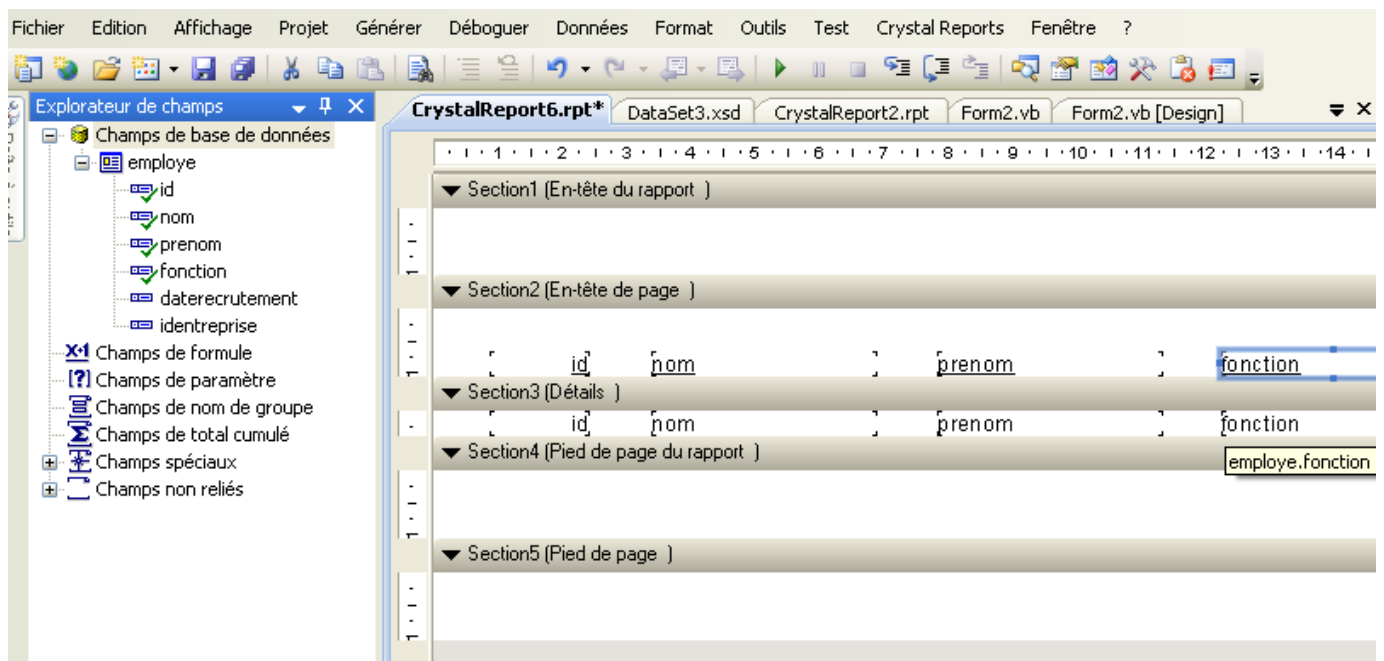
Maintenant que nous avons créé le dataset fortement typé, nous allons construire un état avec Crystal Reports. Tout d'abord il faut ajouter un état Crystal Reports au projet. Dans l'explorateur de solution : click droit sur le projet => Ajouter => Ajouter un nouvel élément.



Sélectionner "Etat Crystal Reports", on appellera cet état : "Etat1.rtp". Cocher « a partir d'un rapport vide ». Dans l'onglet à gauche, click droit sur « les champs de la BD »,



Dans "données du projet", choisissez "ADO.NET Datasets", puis grâce au bouton fléché, basculez notre DataSet dans la partie "Tables sélectionnées". Ensuite pour placer nos données sur l'état, il suffit de réaliser une petite action de glisser-déplacer de nos champs "id" , "nom",... dans la zone détails de notre état, nous obtenons ceci :



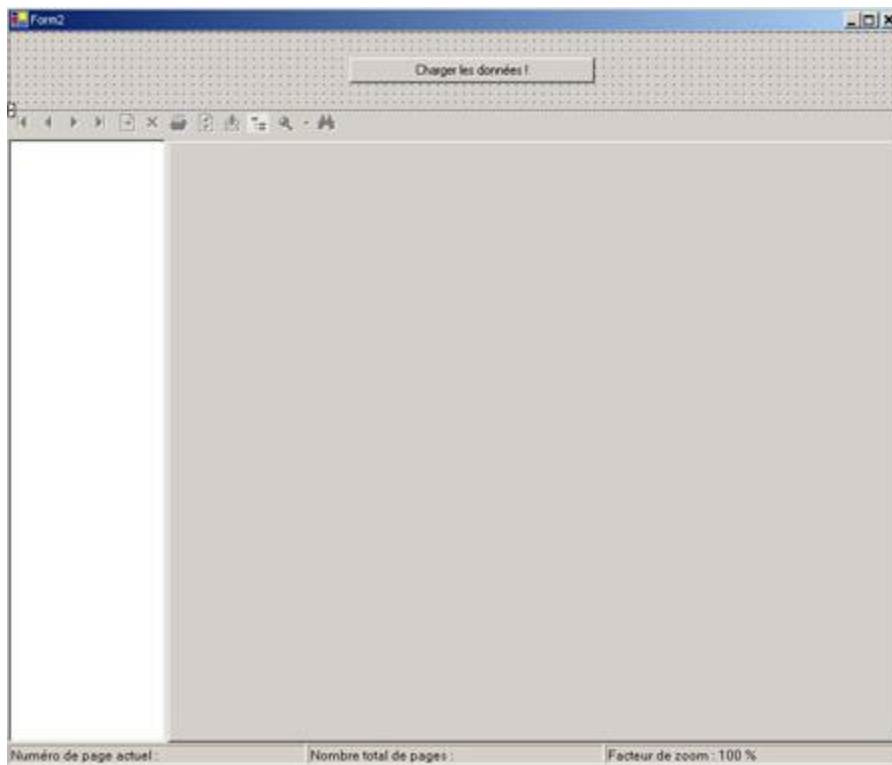
Petite précision sur les différentes zones d'un état :

- En-tête de l'état : tout ce qui sera dans cette zone sera présent sur la première page de l'état.
- En-tête de page : tout ce qui sera dans cette zone sera présent sur chaque page de l'état.
- Détails : tout ce qui sera dans cette zone se reproduira autant de fois que le nombre de "id" que nous avons obtenu grâce à notre requête.
- Pied de page de l'état : tout ce qui sera dans cette zone sera présent que sur la première page de l'état.
- Pied de page : tout ce qui sera dans cette zone sera présent sur chaque page de l'état.

### I-3. Visualisation d'un état dans une application windowsform

Maintenant que nous avons notre dataset fortement typé, et notre état Crystal Reports, voyons comment visualiser et générer un rapport.

Tout d'abord, ajoutez un controle CrystalReportViewer (onglet Windows Form) à votre form1



Le form ci-dessus contient les éléments suivants :

- Un bouton dont l'événement Click nous servira à remplir la source de données de notre état et d'afficher ce dernier.
- Un **CrystalReportViewer** qui contiendra notre état Crystal Report.

Dans l'évènement click du bouton, on ajoute le code qui nous permet de passer les données du dataset à notre crystal report et le visualiser à l'aide du viewer.

```

DataSet1 ds = new DataSet1();
DataSet1TableAdapters.filiereTableAdapter dapFiliere = new
DataSet1TableAdapters.filiereTableAdapter();
dapFiliere.Fill(ds.filiere);
DataSet1TableAdapters.stagiaireTableAdapter dapStagiaire
= new DataSet1TableAdapters.stagiaireTableAdapter();
dapStagiaire.Fill(ds.stagiaire);
CrystalReport1 rapport = new CrystalReport1();
rapport.SetDataSource(ds);
crystalReportViewer1.ReportSource = rapport;

```

Remarques :

- La propriété « SetDataSource » transmet le dataset au rapport
- La propriété « ReportSource » définit le rapport à lier avec le viewer

### TP0 :

Créer, dans votre projet, un rapport qui affiche la liste des stagiaires.

Form1

1 / 1

Apply Delete Main Report

12-03-20

<u>idFiliere</u>	<u>nomF</u>	<u>idstagiaire</u>	<u>nom</u>	<u>prenom</u>	<u>idfiliere</u>
1	TDI	1	n1	p1	1
1	TDI	2	n2	p2	1
1	TDI	5	n5	p5	1
1	TDI	8	oooo	oooo	1
1	TDI	9	n5	p5	1
2	TDM	3	n3	p3	2
2	TDM	4	n4	p4	2
2	TDM	10	Hakimi	ali	2
6	f6	6	n12	p12	6
6	f6	7	llll	hhhh	6

### TP1 :

Créer un rapport qui représente les stagiaires de chaque filière.

ID de la filière :	1
Nom de la filière:	TDI

<u>idstagiaire</u>	<u>nom</u>	<u>prenom</u>	<u>idfiliere</u>
1	n1	p1	1
2	n2	p2	1
5	n5	p5	1
8	oooo	oooo	1
9	n5	p5	1

Mise en forme de l'état.

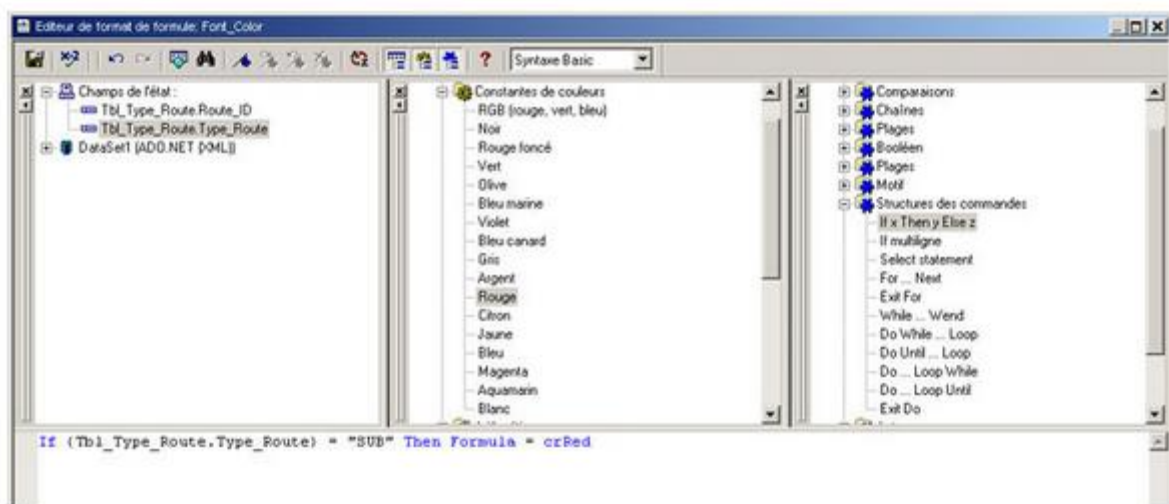
#### A. Exemple : couleur de la police.

Revenons à notre état en mode conception, et faisons un clic droit sur notre champ "Route\_id" dans notre zone détails, puis choisissons "mettre en forme", cette fenêtre apparaît :



Nous remarquons plusieurs onglets ou nous pouvons entre autre changer la couleur de police ou mettre des bordures. Pas besoin de s'étendre ici. Ce qui est intéressant dans cette fenêtre c'est la présence d'un étrange bouton :



C'est là que c'est intéressant, ce bouton va nous permettre de changer la propriété correspondante de notre champ "Route\_id" de manière dynamique. Un petit exemple, imaginons que nous voulions mettre en rouge toutes les routes qui sont de types "SUB". Cliquez donc sur le bouton correspondant à la couleur de la police. Cette fenêtre apparaît :



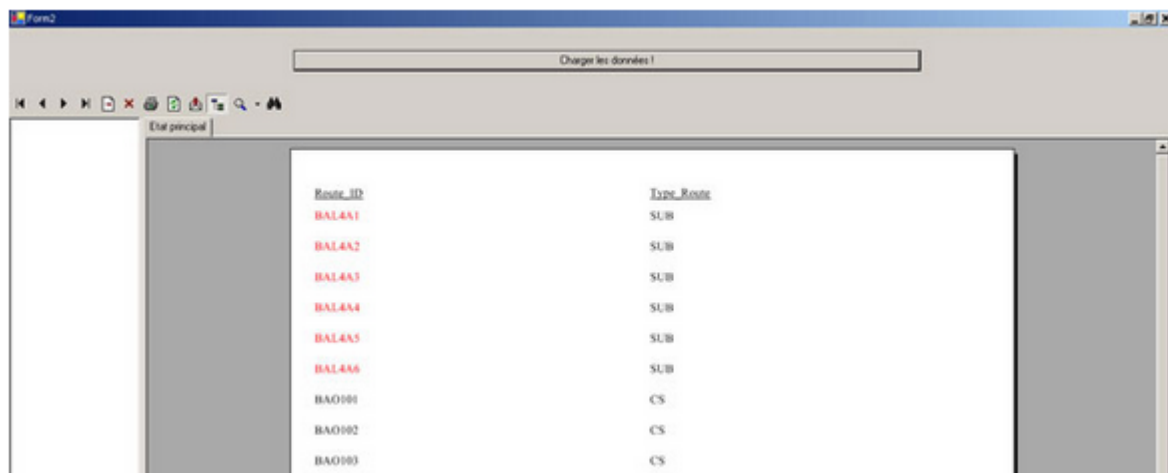
En haut de la fenêtre une zone déroulante nous permet de choisir la syntaxe Basic, choisissez la donc. Les trois fenêtres du haut nous présentent les champs de notre source de données, des constantes, des fonctions, des structures de commandes... afin d'éviter de taper tout le code à la main. Nous allons donc programmer la propriété "Font\_Color" de notre champ "Route\_id" pour le mettre en rouge pour un certain type.

Pour cela double cliqué sur la structure de commande du IF. Une fois que le curseur est dans le IF, double cliqué sur notre champ "Type\_Route". Vous voyez la formule prendre forme dans la fenêtre du bas. Tapez ' = "SUB" ', puis dans le Then, tapez le mot clé "Formula = " et dans la fenêtre du milieu, choisissez la couleur Rouge. Le mot clé "Formula" fait référence à la propriété sur laquelle nous travaillons, ici la Font\_Color de notre champ "Route\_id".

Notre formule est donc prête et veux dire : Si Type\_Route = "SUB" Alors Font\_Color = Rouge !

Vous pouvez vérifier la syntaxe de notre formule en cliquant sur le bouton  et enregistrer puis quitter en cliquant sur le bouton .

Faisons un test et relançons notre application, nous avons bien nos routes de type "SUB" apparaitre en rouge :



Route_ID	Type_Route
BAI4A1	SUB
BAI4A2	SUB
BAI4A3	SUB
BAI4A4	SUB
BAI4A5	SUB
BAI4A6	SUB
BAO101	CS
BAO102	CS
BAO103	CS

## B. Création d'un groupe.

Un groupe dans un état va nous permettre de regrouper et d'ordonner les données selon certains critères. Nous allons dans cet exemple regrouper les routes par type de route. En affichant d'abord les CS routes puis les SUB.

La création d'un groupe se réalise dans la partie design de notre état, dans l'onglet "explorateur de champs", faites un clic droit sur "Champ nom du groupe" puis "nouveau", nous voyons apparaitre cette fenêtre :

Insérer un groupe

Commun

Lorsque l'état est imprimé, les enregistrements sont triés et regroupés par :

Tbl\_Type\_Route.Type\_Route

dans l'ordre croissant.

La section sera imprimée sur tout changement de Tbl\_Type\_Route.Type\_Route

Options de groupe

☐ Personnaliser le champ Nom du groupe

☒ Choisir un champ existant

Tbl\_Type\_Route.Route\_ID

☐ Utiliser une formule comme nom du groupe

☐ Conserver les groupes ensemble

☐ Répéter l'en-tête de groupe sur chaque page

OK Annuler Aide

Choisissez donc le champ "Type\_Route" et "dans l'ordre croissant" pour voir apparaître les "CS" puis les "SUB".

Vous voyez alors la zone de groupe sur notre design et si on relance notre application nous avons ceci :

Form2

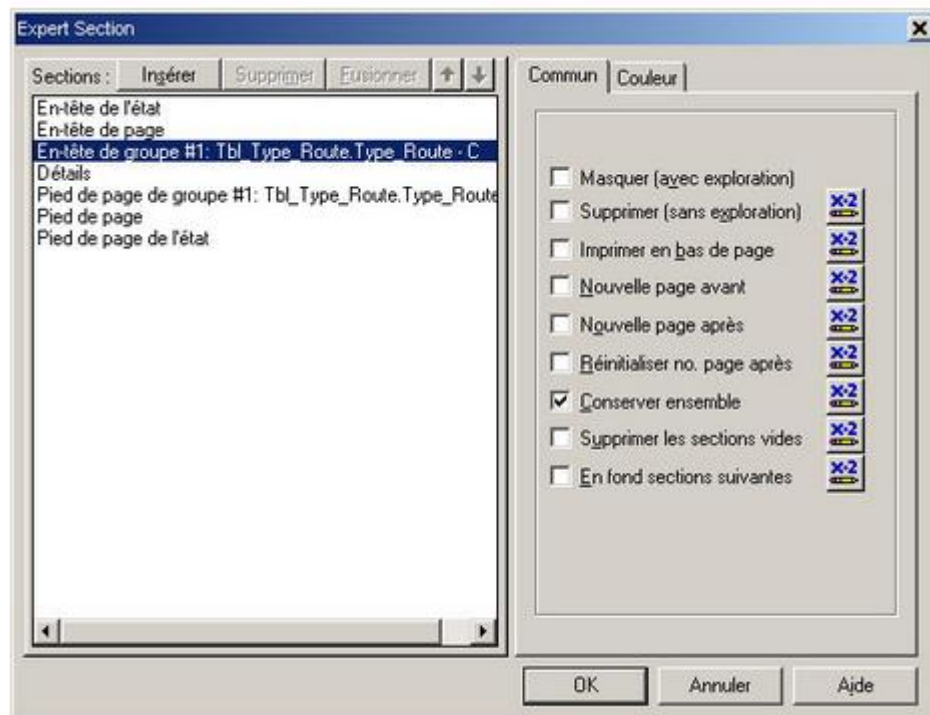
Charger les données !

Etat principal

Route_ID	Type_Route
CS	CS
BAO101	CS
BAO102	CS
BAO103	CS
BAO1A1	CS
BAO1A2	CS
BAO1A3	CS
BAO1A4	CS
BAO1A5	CS

Nous voyons apparaître à gauche nos groupes, ici nos types "CS" et "SUB" dans l'ordre croissant (alphabétique ici). Quand on clique sur l'un des groupes, une zone rouge nous y amène directement. Il est à noter que nos zones d'état (groupe, détails, pied de page...) peuvent également faire l'objet de mise en forme. En faisant un clic droit sur n'importe quelle section et en choisissant "mettre en forme une section", nous voyons cette fenêtre :

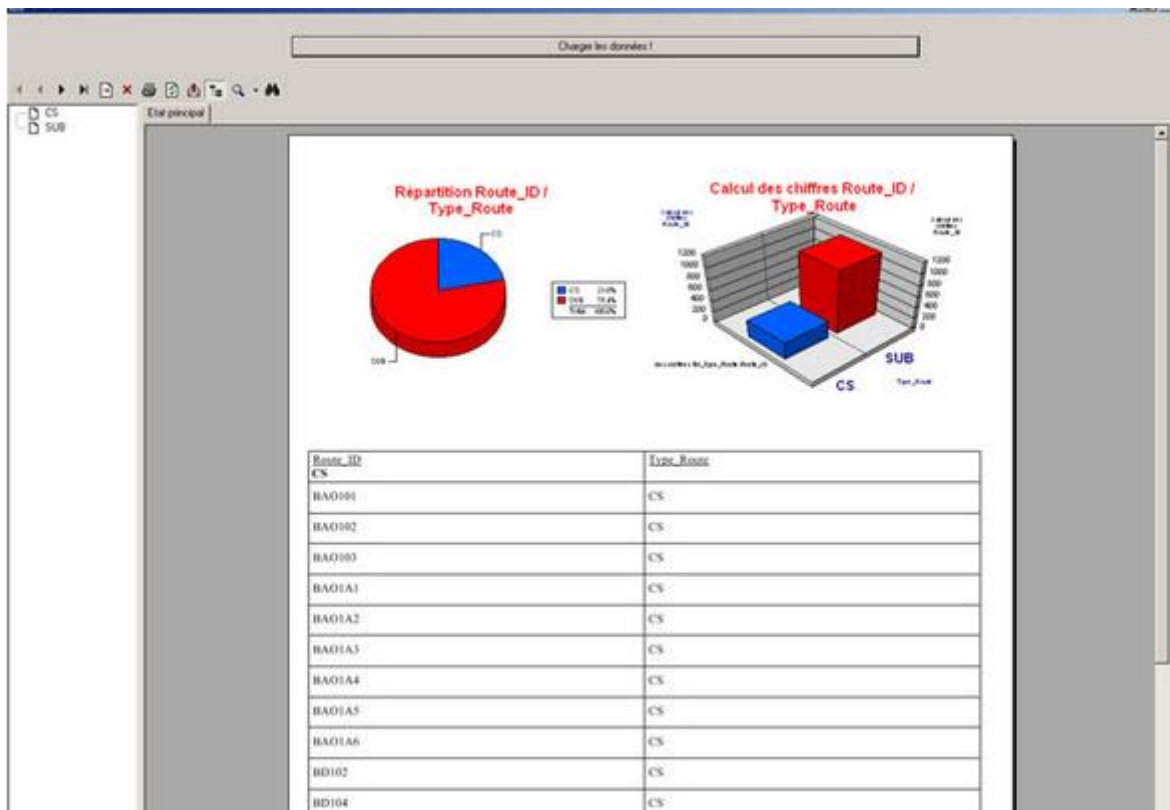




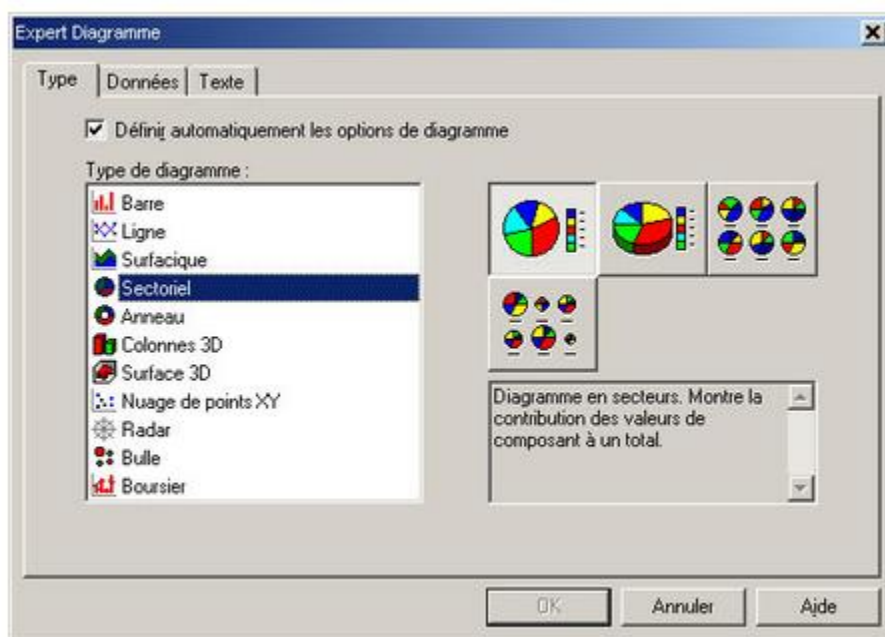
Vous pouvez par exemple en sélectionnant la section détails puis en validant l'option "Nouvelle page après" avoir une nouvelle page après chaque zone de détails. C'est à dire que l'on aura une route par page.

## Insertion de diagramme

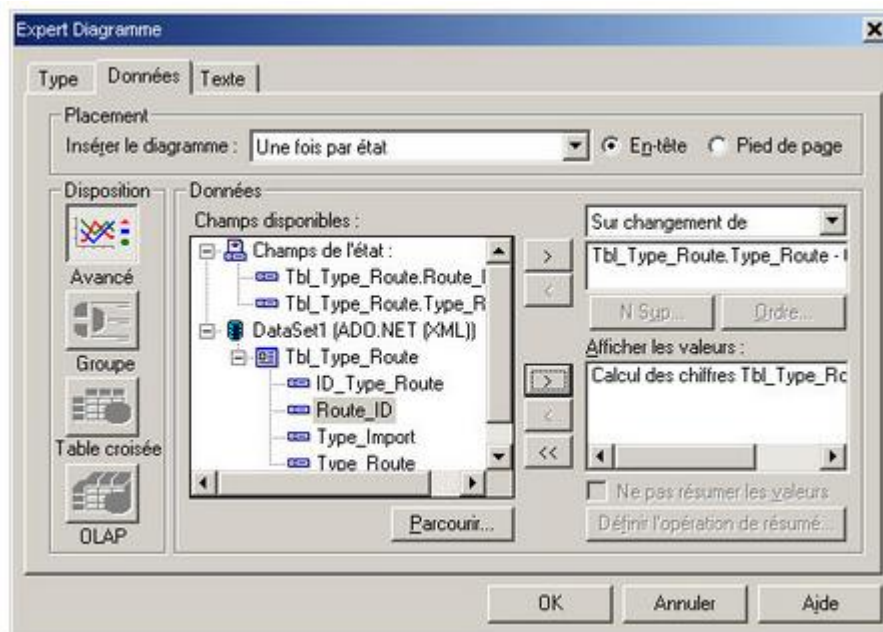
Le but de ce chapitre est d'insérer un diagramme pour représenter nos données de manière plus synthétique. Voici une capture d'écran de ce que nous allons pouvoir obtenir très rapidement :



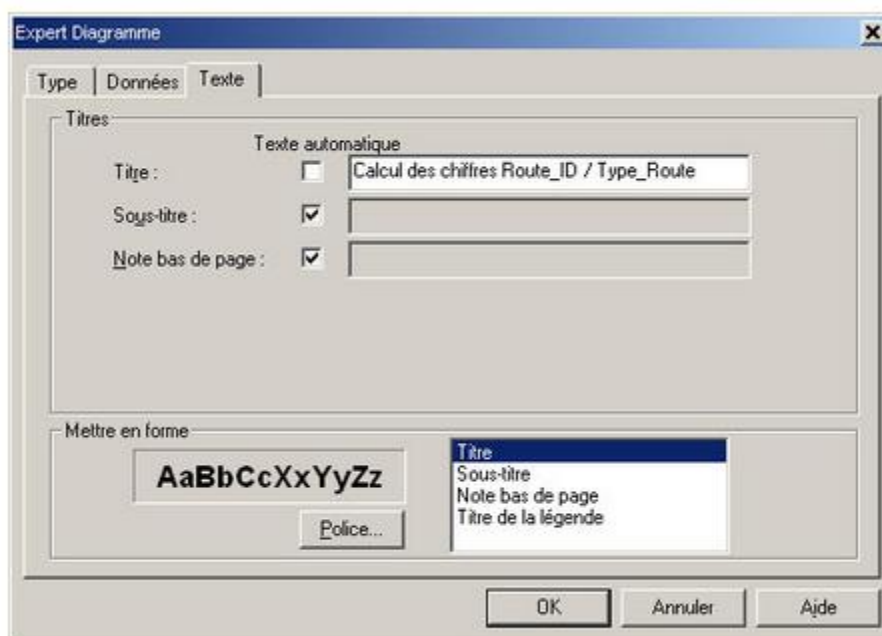
Nous allons insérer notre diagramme dans l'en-tête de l'état. Faites donc un clic droit dans cette partie de l'état puis "insertion" - "diagramme". Voici les étapes successives pour la création de notre diagramme :



La première chose à faire est de choisir le type de notre diagramme, prenons le type sectoriel (ou camember).



C'est ici que nous allons préciser que nous voulons la répartition des routes selon les types de routes. Nous disons donc que sur un changement du type de route, nous calculons le nombre de routes.



Dans l'onglet texte, nous précisons uniquement le texte à mettre pour le titre, sous-titre... de notre diagramme.

Le deuxième diagramme présent sur notre image au début du chapitre a été conçu exactement de la même façon sauf que nous avons choisis un autre type de diagramme !

## Le controle CrystalReportViewer

Nous allons ici parcourir quelques propriétés du controle crystalReportViewer:

*DisplayGroupTree* : permet de faire disparaître la colonne blanche à gauche de l'état. Il est inutile lorsque votre état ne contient pas de groupe. *DisplayToolBar* : permet de faire disparaître totalement la barre d'outils de l'état. L'utilisateur n'aura alors plus accès à l'impression, au zoom ...

- *ShowCloseButton* : permet de faire disparaître le bouton "fermer" de l'état. Il est bien souvent inutile
- *ShowExportButton* : cette propriété est très intéressante. Elle vous permet très facilement d'exporter votre état au format pdf, xls, doc, rtf d'un simple clic !
- *ShowGoToPageButton* : ce bouton permet d'accéder directement à une page de l'état en tapant son numéro
- *ShowGroupTreeButton* : ce bouton n'est plus nécessaire si vous décidez de faire disparaître le group tree
- *ShowPageNavigateButton* : permet d'afficher les boutons flechés pour se déplacer dans l'état, vous pouvez naviguer de pages en pages ou vous rendre
- directement à la première ou à la dernière page
- *ShowPrintButton* : très utile, permet d'imprimer votre état d'un simple clic
- *ShowRefreshButton* : permet de rafraîchir l'état, pour ma part je ne l'ai jamais utilisé
- *ShowTextSearchButton* : permet de rechercher un mot dans l'état
- *ShowZoomButton* : permet de régler le zoom de l'état.
- par défaut le zoom sera de 100%, vous pouvez changer le zoom par défaut avec le code :  
VotreCrystalReportViewer.Zoom(2)
- (ici on affiche toute la page de l'état)