

SharePoint et SQL Server 2008 R2 Report Server

Comment accéder à SharePoint depuis SSRS 2008 R2



Depuis la première version, la solution de rapport proposée par Microsoft "SQL Server Reporting Services" n'a cessé d'évoluer.

Avec la version 2008 R2, Microsoft ajoute l'utilisation de SharePoint comme source de données en natif. Nous verrons donc comment utiliser cette fonctionnalité avec quelques exemples.

Introduction

Depuis sa création, SQL Server Reporting Services permet de générer des rapports à destination des utilisateurs de manière industrielle.

En effet, cette solution fournit des rapports en se basant sur un modèle de page fourni et une source de données. La version 2008 R2 du produit permet donc d'utiliser SharePoint comme source de données sans besoin de complément additionnel.

Présentation

Un précédent article présentait les solutions possibles pour exploiter les données SharePoint depuis les versions 2005 ou 2008 :

- **Reporting Services et les données stockées dans SharePoint**

La version SQL Server 2008 R2 Reporting Services possède un ensemble de nouveautés dont les suivantes :

- Utilisation de listes SharePoint 2007 ou 2010 comme sources de contenu
- Génération de rapports intégrant les données Géo spatiales - Type carte planisphère

Nous verrons dans cet article comment créer un rapport accédant à une liste SharePoint basique, puis des exemples plus complexes.

Rapport simple affichant des données d'une liste basique

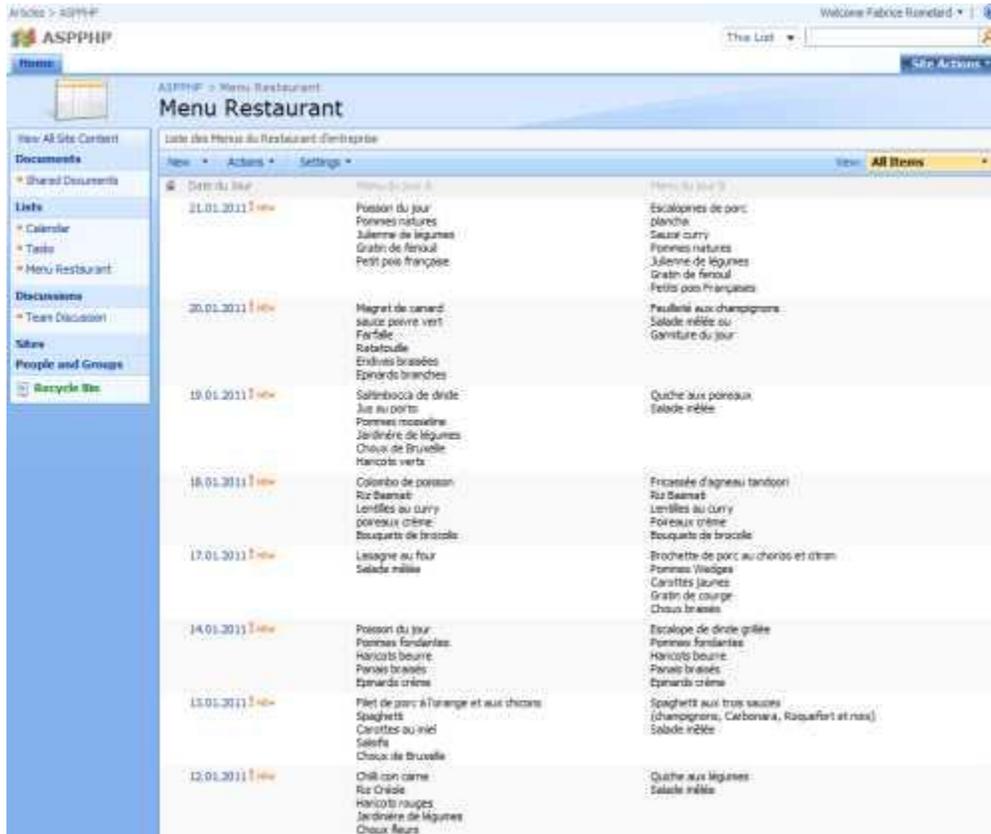
Le principe n'est pas refaire l'article précédent ([Reporting Services et les données stockées dans SharePoint](#)), mais plutôt de créer un rapport basique affichant des données provenant d'une liste SharePoint classique.

Préparation de la liste source

Pour ceci, nous allons travailler avec une liste qui a pour but de stocker les menus du restaurant d'entreprise. Elle est donc de type Custom possède trois colonnes :

- Title: Champ de base Inutilisé dans notre cas, il aura toujours Menu comme valeur et sera masqué dans les vues
- Date du jour: Champ type Date
- Menu du jour A: Champ type RichText
- Menu du jour B: Champ type RichText

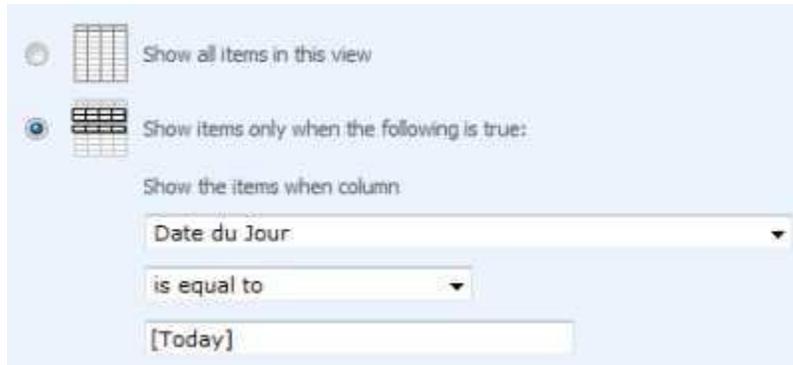
Une fois la liste chargée de son contenu, on modifie la vue par défaut afin de ne pas voir le champ Title et obtient ceci :



Date du jour	Menu du jour A	Menu du jour B
21.01.2011	Poisson du jour Pommes nature Julienne de légumes Gratin de fenouil Petit pois français	Escalopes de porc Panzanella Sauté curry Pommes nature Julienne de légumes Gratin de fenouil Petit pois français
20.01.2011	Magret de canard sauce poivre vert Farfalle Ratatouille Endives braisées Epinards branches	Feuilleté aux champignons Salade mâchée au Garçure du jour
19.01.2011	Salmis de dinde Jus au poivre Pommes mousseline Jardinière de légumes Choux de Bruxelles Haricots verts	Quiche aux poireaux Salade mâchée
18.01.2011	Colombo de poisson Riz Saarnat Lentilles au curry poireaux crème Bouquets de brocolis	Fricassée d'agneau tandoori Riz Saarnat Lentilles au curry Poireaux crème Bouquets de brocolis
17.01.2011	Lasagne au four Salade mâchée	Brochette de porc au chorizo et citron Pommes Wedges Carottes jeunes Gratin de courge Choux braisés
14.01.2011	Poisson du jour Pommes fondantes Haricots beurre Pommes braisées Epinards crème	Escalope de dinde grillée Pommes fondantes Haricots beurre Pommes braisées Epinards crème
13.01.2011	Filet de porc à l'orange et aux champignons Spaghettis Carottes au miel Salade Choux de Bruxelles	Spaghettis aux trois sauces (champignons, Carbonara, Roquefort et noix) Salade mâchée
12.01.2011	Chili con carne Riz Orzo Haricots rouges Jardinière de légumes Choux fleurs	Quiche aux légumes Salade mâchée

Dans le Team Site en lui-même, il est facile de faire une vue pour n'afficher que le menu du jour courant, pour cela il suffit de créer une vue en mettant dans le filtre :

- Date du Jour is equal to [Today]



On obtient alors pour le jour courant le menu prévu :



Ceci est parfait lorsque l'on navigue dans ce TeamSite, mais le gros intérêt de ce type d'information est de pouvoir le diffuser dans les autres sites Intranet, plus particulièrement le Portail Intranet de l'entreprise.

Création du rapport simple

Ainsi, lorsqu'un employé arrive à son poste le matin et se connecte sur l'Intranet, il sait ce qui lui est proposé en menu au restaurant de son entreprise.

Dans le cas présent, il n'y a pas de solution basique permettant de fournir l'affichage de ce genre d'information. Il est obligatoire de faire un minimum de développement (ASP.NET ou SharePoint Designer), ce qui peut être compliqué suivant les entreprises.

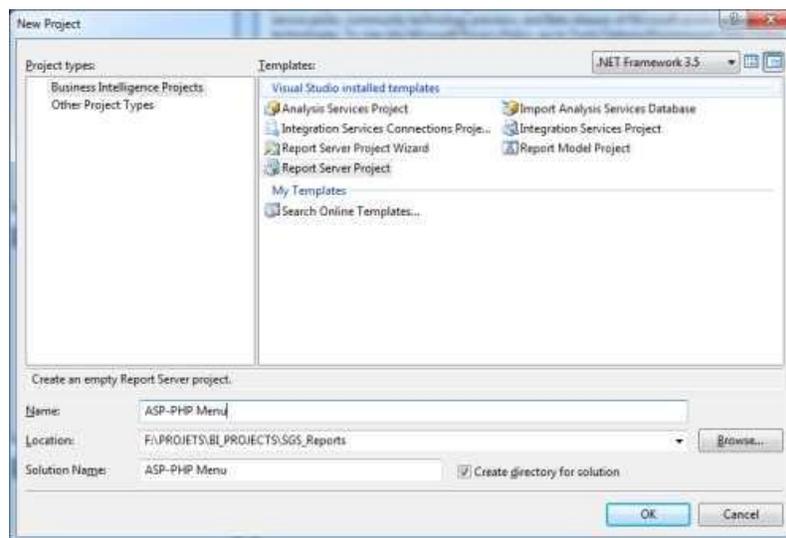
La solution la plus simple est donc de créer un rapport Reporting Services qui va être ensuite simplement intégré dans le portail en utilisant les WebParts de visualisation de SSRS pour SharePoint.

Création du rapport

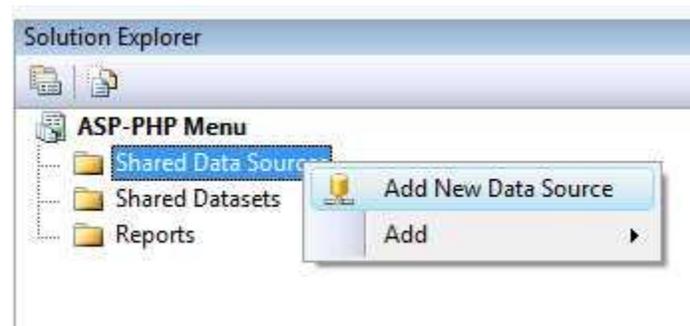
Il faut bien sûr commencer par lancer l'outil de développement qui reste le même :

- SQL Server Business Intelligence Development Studio

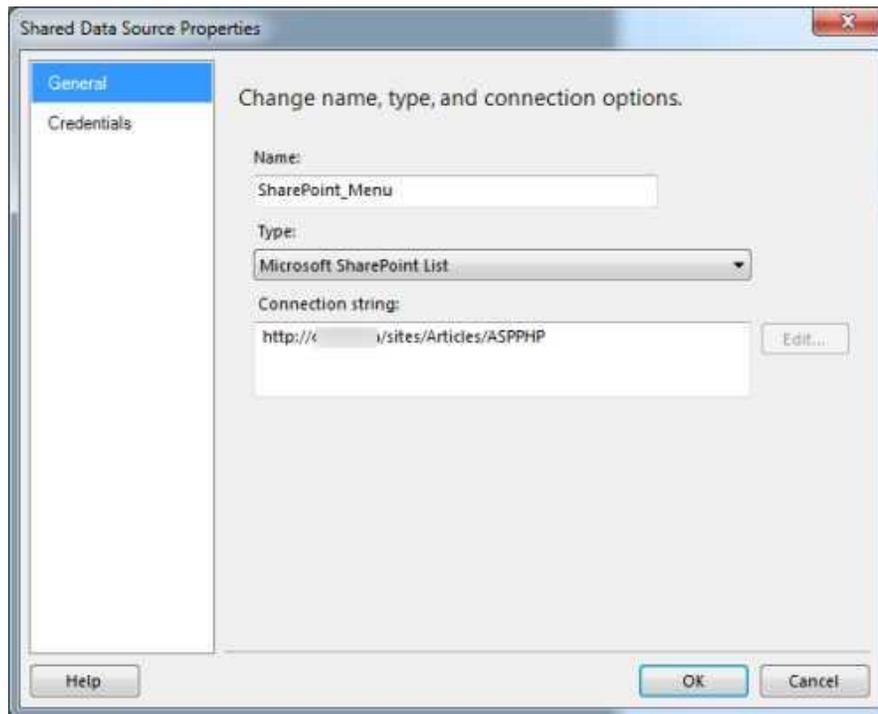
On va créer un nouveau projet (File > New > Project) de type "Report Server Project" que l'on appellera "ASP-PHP Menu"



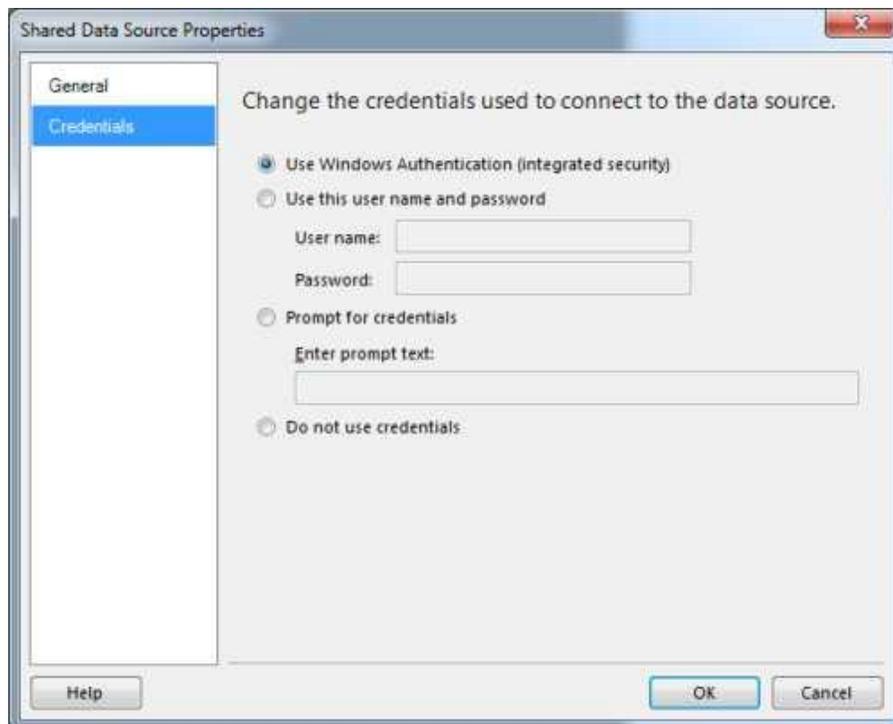
On va créer une source de données partagée en cliquant avec le bouton droit sur l'icône "Shared Data Source" et choisissant "Add New Data Source"



Dans la boîte de dialogue, il faut choisir le Type "Microsoft SharePoint List" et ajouter dans le champ « Connection String » l'URL vers le Team Site ou se trouve notre liste.



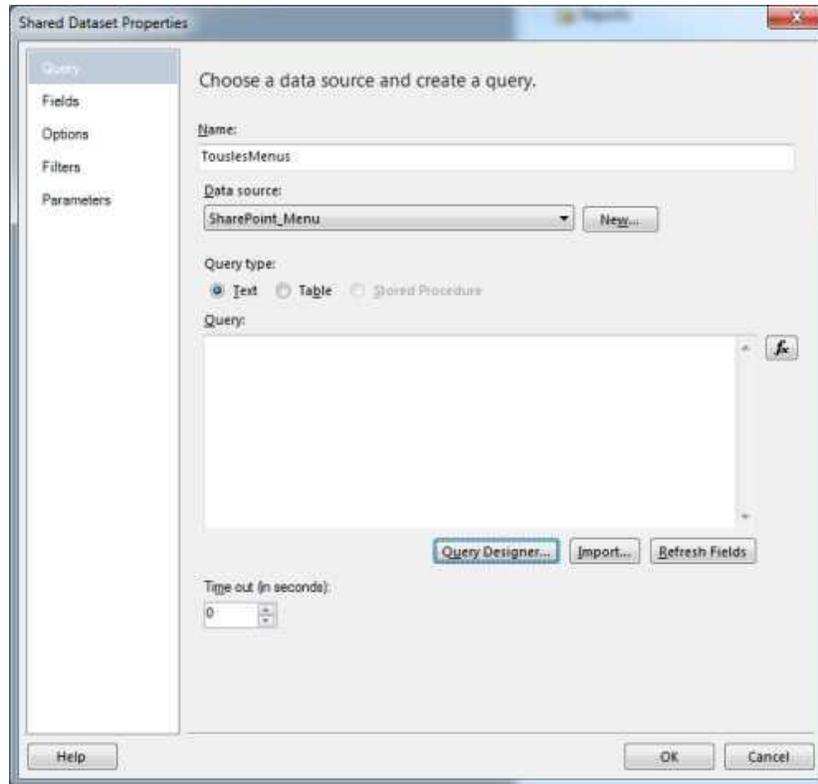
Il faut aussi cocher dans la partie "Credentials" "Use Windows Authentication (integrated security)"



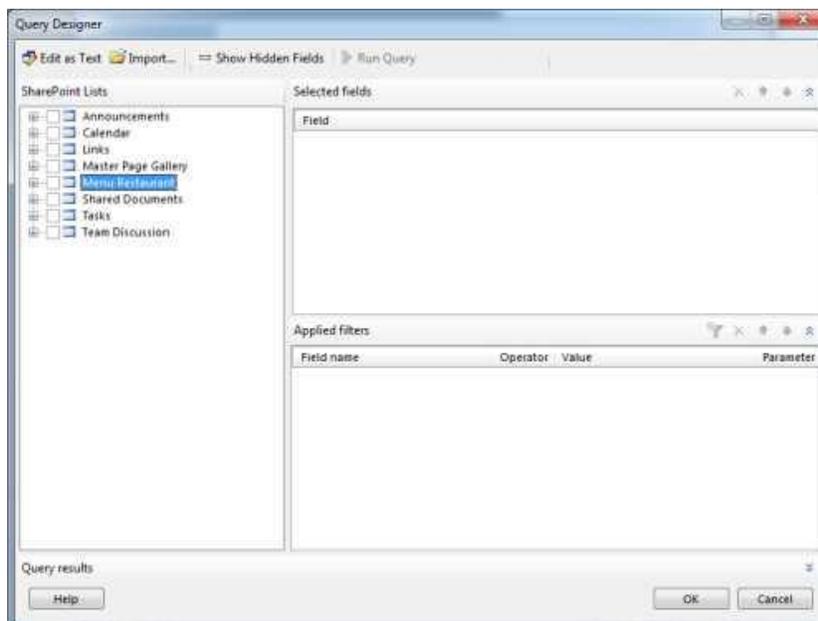
A ce stade, la connexion est créée pour notre projet de rapport. Ce sera une connexion de type partagée afin de pouvoir utiliser la même pour plusieurs rapports.

Création d'un DataSet partagé

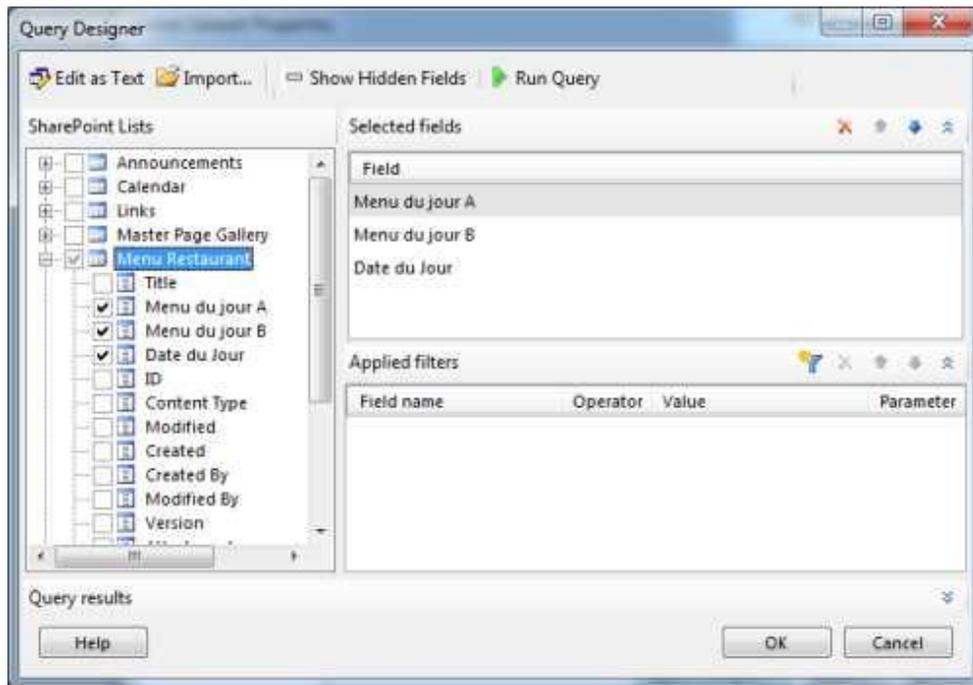
Nous pouvons maintenant créer un DataSet, qui sera donc un jeu de données utilisé dans un ou plusieurs rapports. Pour cela, il faut cliquer avec le bouton droit sur "Shared DataSets" et choisir "Add a new DataSet". On obtient alors un écran nous proposant de renseigner les champs



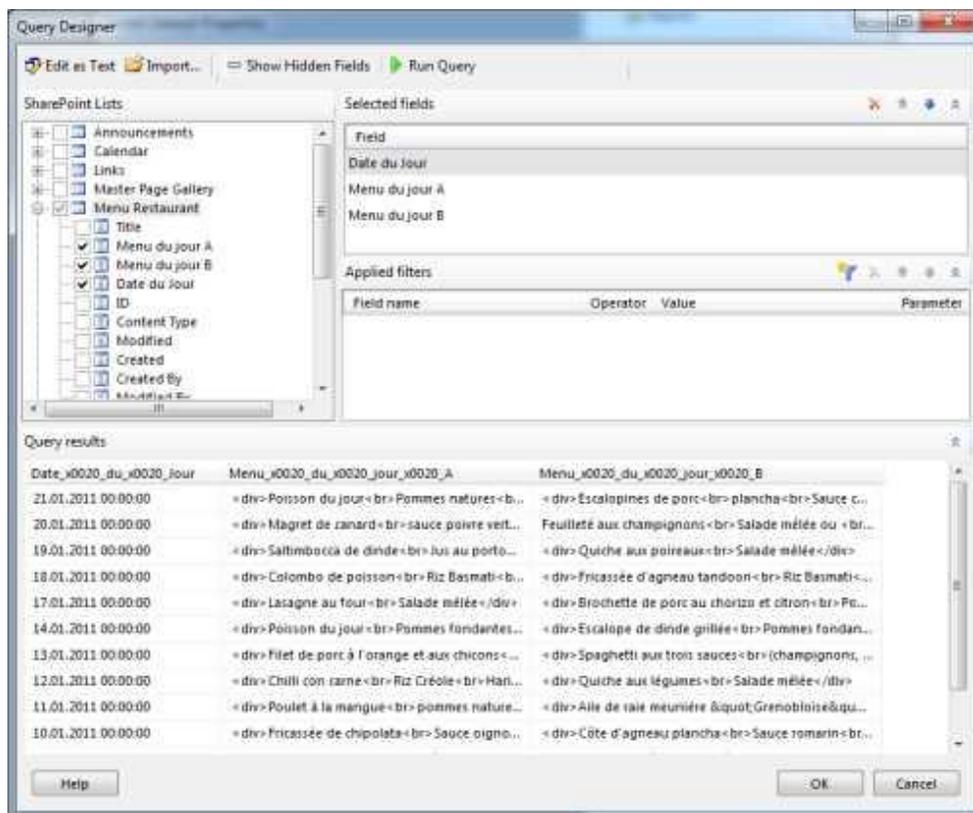
Il faut maintenant cliquer sur "Query Designer..." pour définir le jeu de données que l'on souhaite et surtout faire apparaître le nouvel assistant spécifique à SharePoint



On peut donc sélectionner la liste souhaitée "Menu SharePoint", mais surtout ne cocher que les colonnes qui nous intéressent, donc Menu A, Menu B et Date Menu

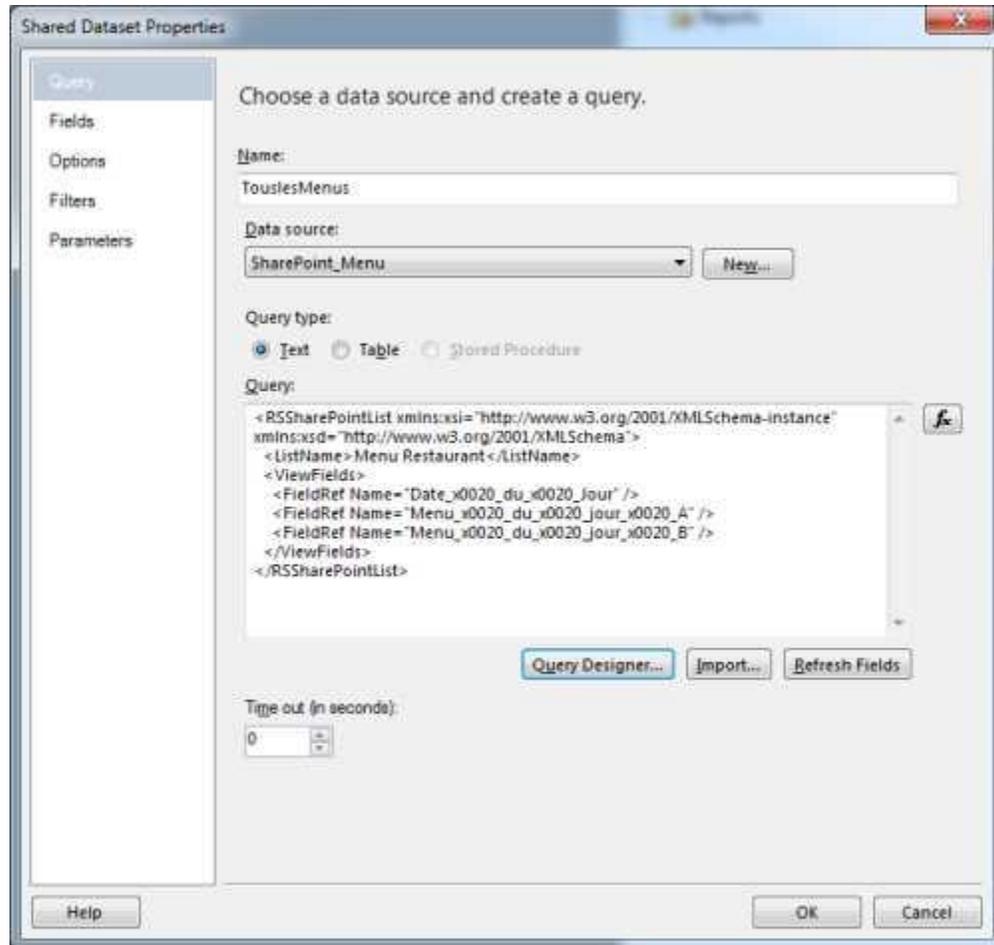


On peut ensuite changer l'ordre des colonnes et cliquer sur "Run Query" pour voir le résultat de la sélection



Notre sélection de données est donc terminée et nous validons par OK. Le résultat de cette sélection est une requête CAML qui est intégrée dans un bloc XML "**RSSharePointList**" :

```
<RSSharePointList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ListName>Menu Restaurant</ListName>
  <ViewFields>
  <FieldRef Name="Date_x0020_du_x0020_Jour" />
  <FieldRef Name="Menu_x0020_du_x0020_jour_x0020_A" />
  <FieldRef Name="Menu_x0020_du_x0020_jour_x0020_B" />
  </ViewFields>
</RSSharePointList>
```

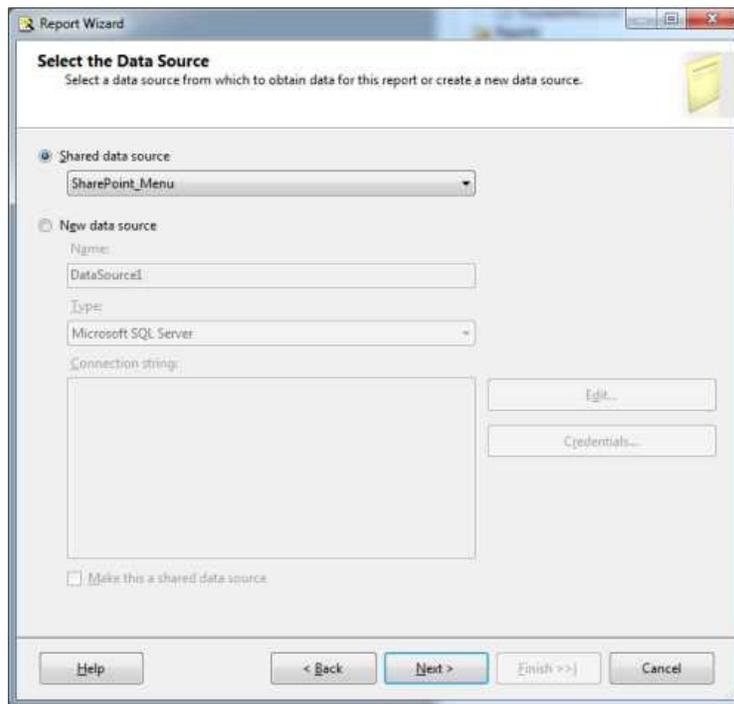


Ce jeu de données peut ensuite être intégré dans un ou plusieurs rapports selon son besoin.

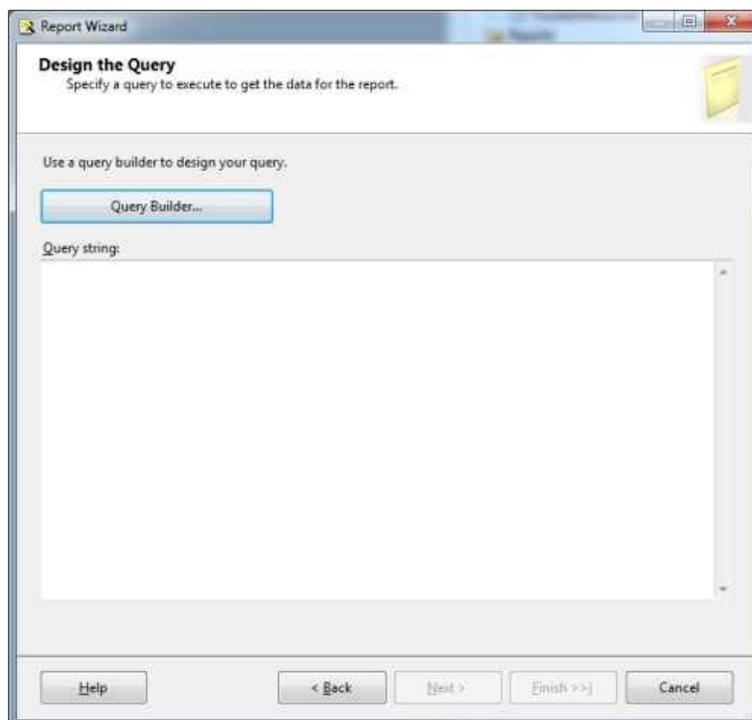
Création du rapport simple avec l'assistant

Pour notre premier exemple, il nous faut simplement afficher les valeurs de notre Liste SharePoint. Nous pouvons créer ce rapport basique en passant par l'assistant disponible dans l'outil.

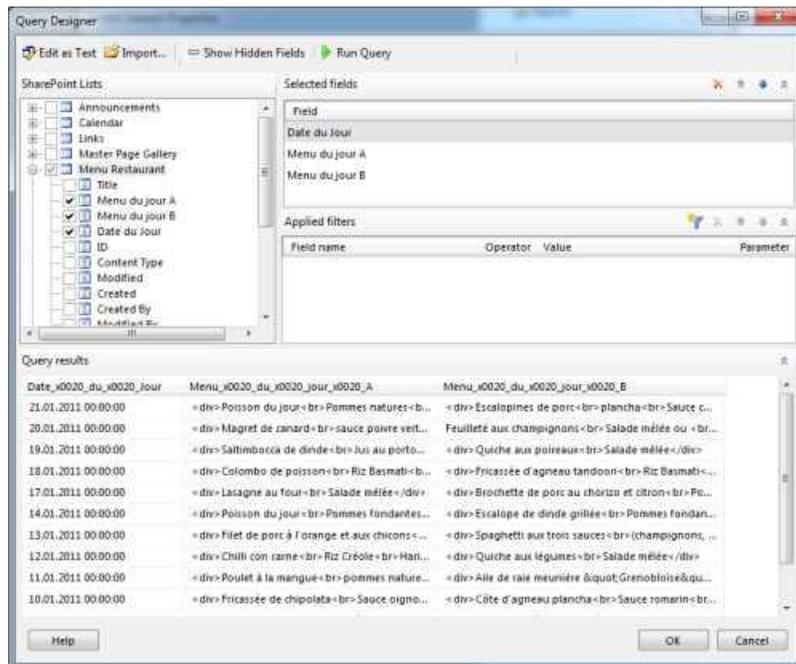
Pour ceci, nous allons cliquer avec le bouton droit sur "Reports", puis choisir "Add New Report". L'assistant s'exécute alors, nous proposant de cliquer sur "Next". L'étape suivante nous demande de spécifier la source de données, nous choisissons notre source précédemment créée "SharePoint_Menu" et cliquons sur "Next"



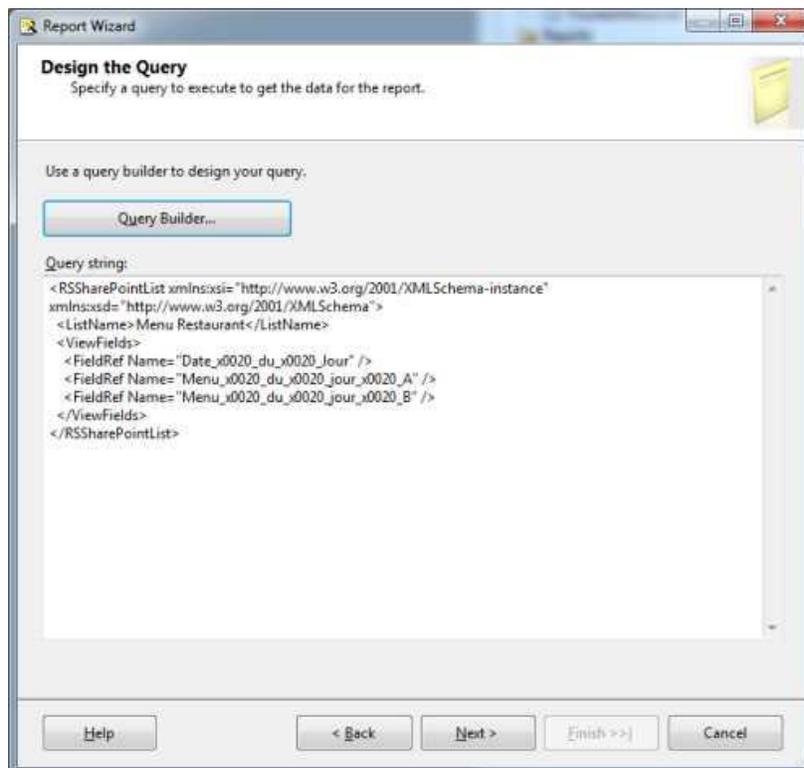
Nous devons maintenant fournir la requête pour le jeu de données, soit manuellement en le tapant dans le champ texte, soit en cliquant sur "Query Builder"



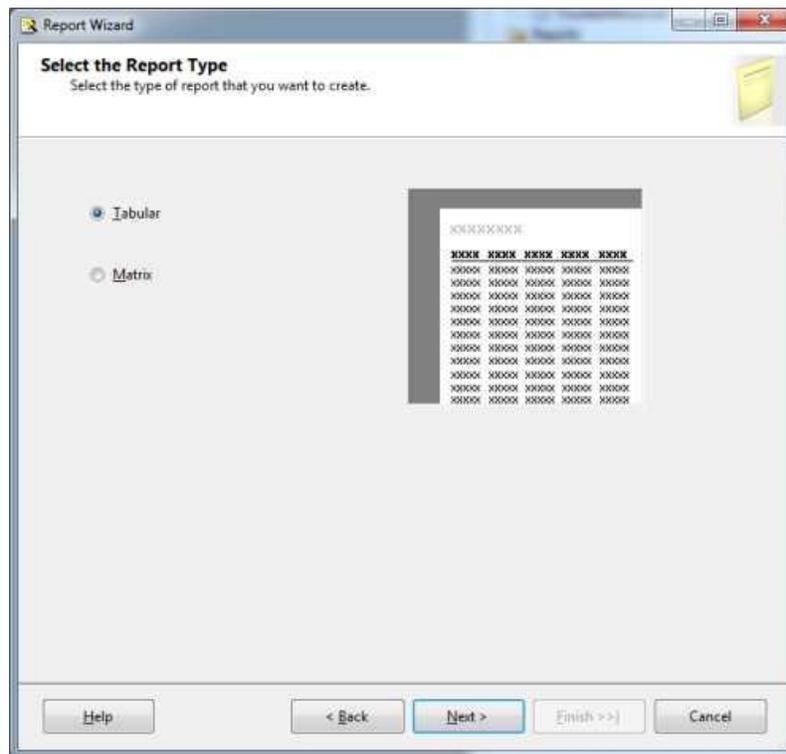
On refait la même sélection que dans l'explication juste avant afin d'avoir les trois colonnes de notre liste et on valide par OK



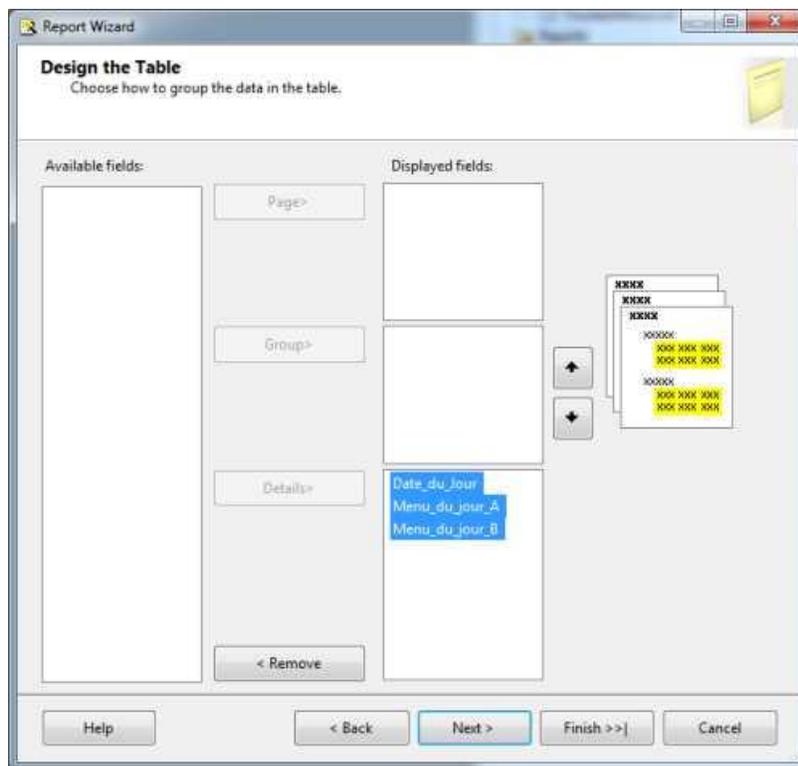
On voit alors la requête CAML dans le champ texte et on clique sur "Next"



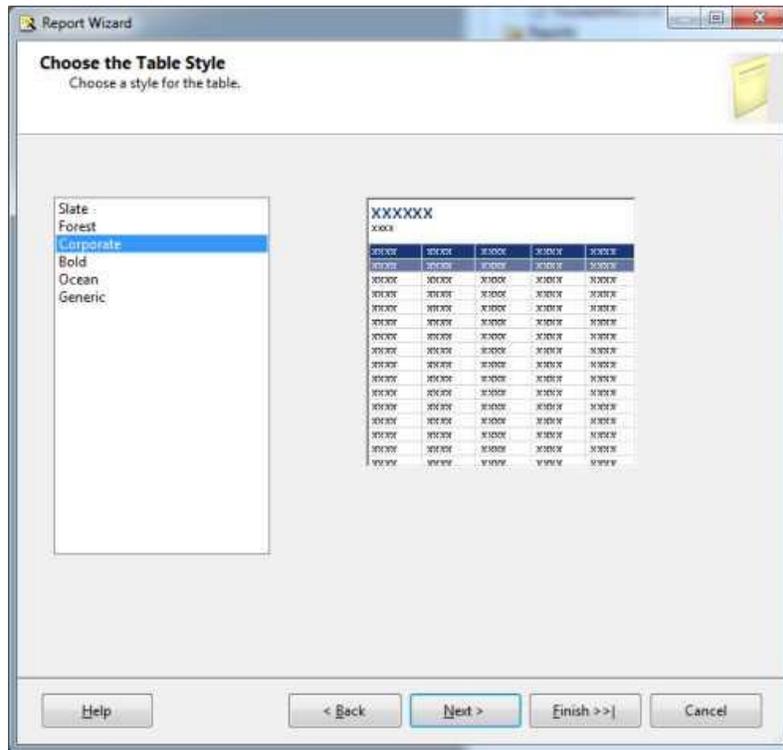
On choisit maintenant le type de rapport "Tabular" et clique sur "Next"



On ajoute toutes les colonnes dans le groupe "Details" et clique sur "Next"



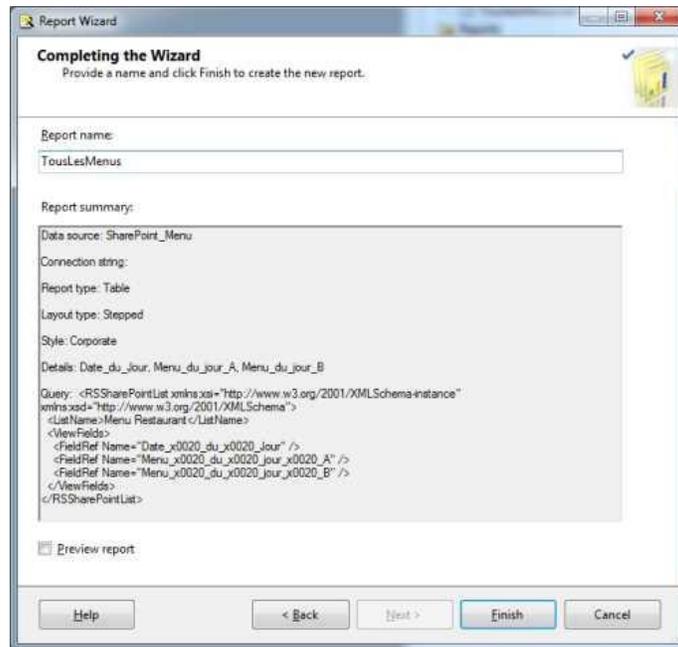
On sélectionne le style de notre rapport parmi les styles prédéfinis et on clique sur "Next"



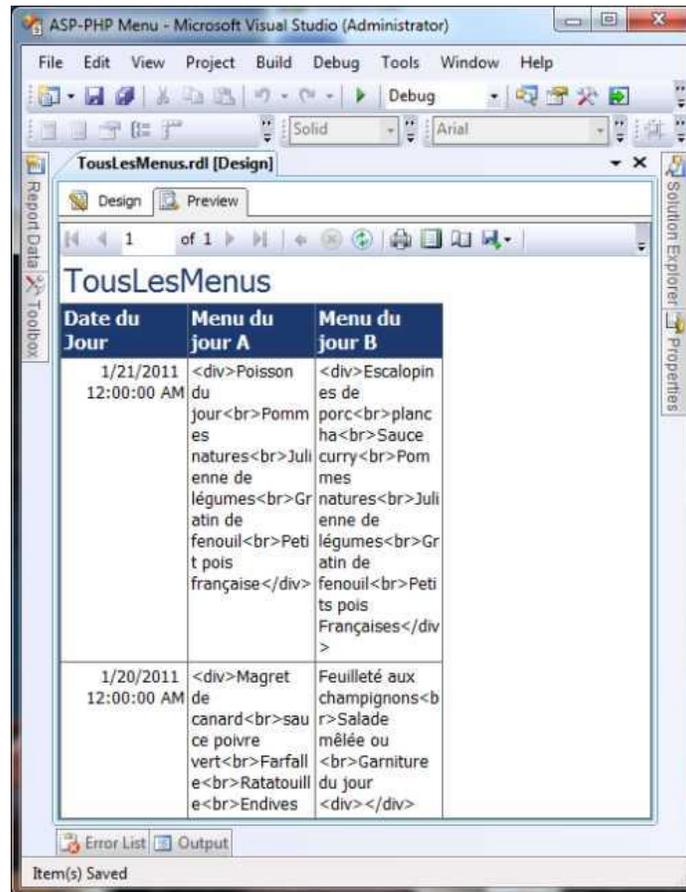
Il est aussi possible de créer son propre style de rapport, vous trouverez un article sur ce sujet :

- **Créer son modèle de style utilisable dans les rapports SSRS**

Il ne nous reste plus qu'à donner un nom à notre rapport et cliquer sur "Finish"

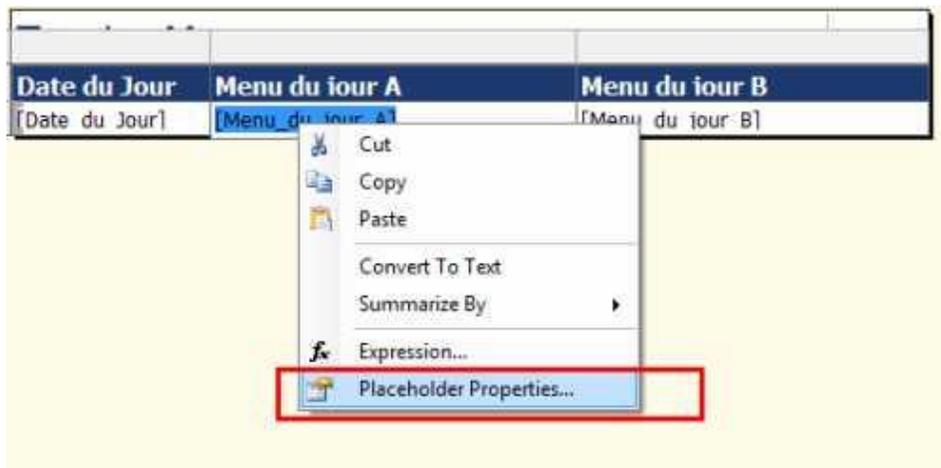


A ce stade, en cliquant sur "Preview", l'éditeur fournit une prévisualisation du rapport

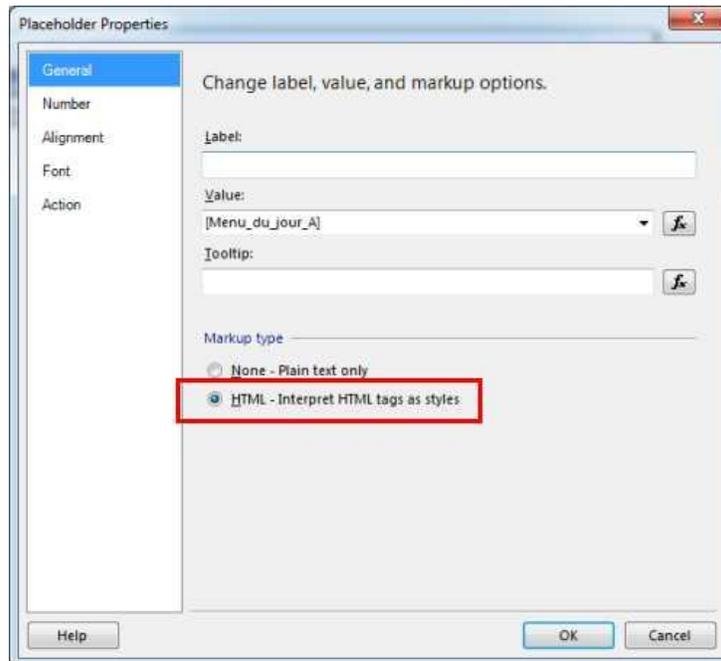


Il nous faut maintenant un peu de personnalisation en agrandissant les colonnes, affichant les dates selon le standard souhaité, et surtout en demandant au serveur de rapport d'interpréter l'HTML fournit par la source de données.

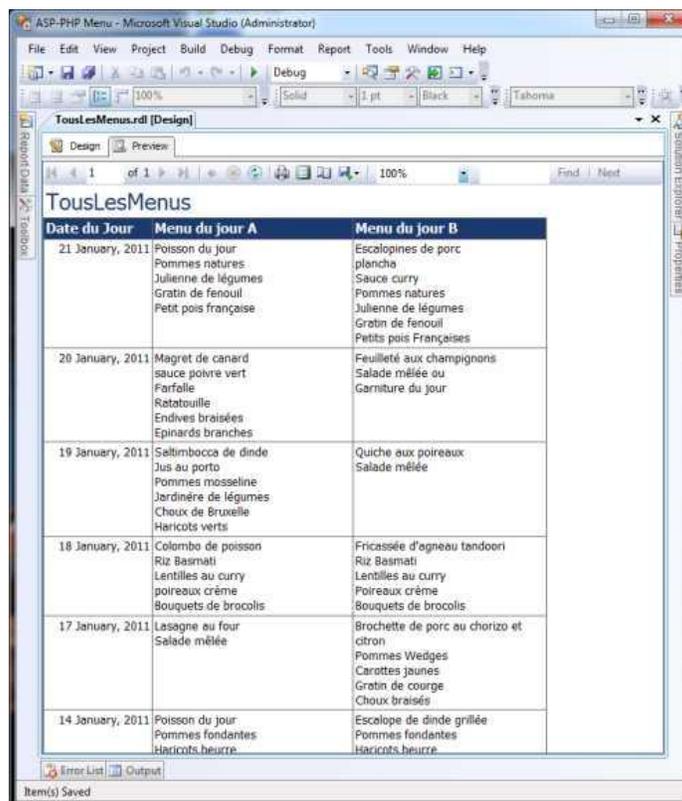
Pour faire cela, il faut cliquer avec le bouton droit sur la case de la donnée et choisir "Placeholder Properties"



Il faut cocher "HTML - Interpret HTML tags as styles" et valider par OK.



Il faut faire cela pour les deux cases de menu et lancer la prévisualisation à nouveau. Notre rapport est alors finalisé.



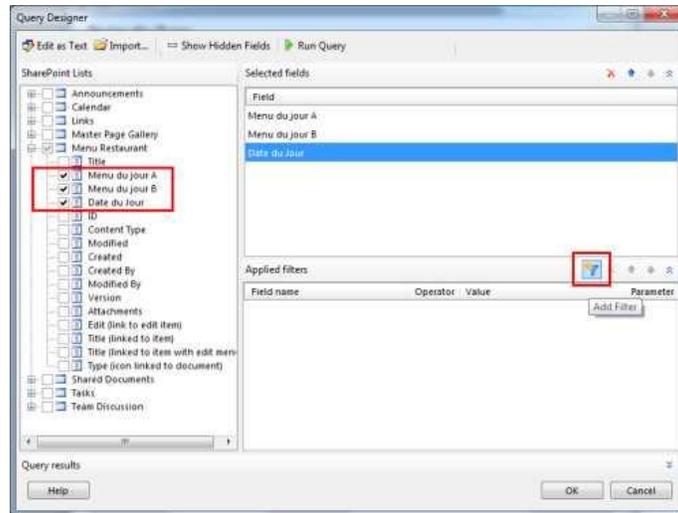
Notre rapport simple est donc terminé. Nous pouvons maintenant ajouter un peu de complexité en cherchant à afficher celui du jour, comme dans notre vue précédente.

Rapport avec filtrage

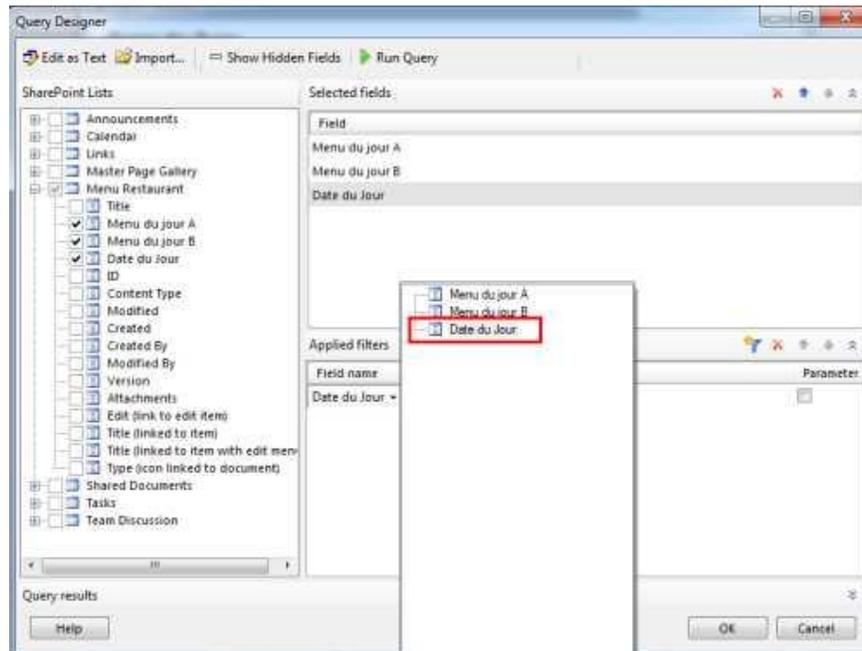
Comme nous l'avons fait dans la vue, nous pouvons effectuer un filtre pour afficher le menu de la date du jour. L'intérêt de ceci est de pouvoir ensuite facilement intégrer ce rapport dans n'importe quelle autre page Intranet de notre site.

Pour cela, nous devons créer un nouveau rapport comme dans l'exemple simple. Nous passerons à nouveau par l'assistant de création du nouveau rapport en suivant les mêmes étapes jusqu'au Query Builder.

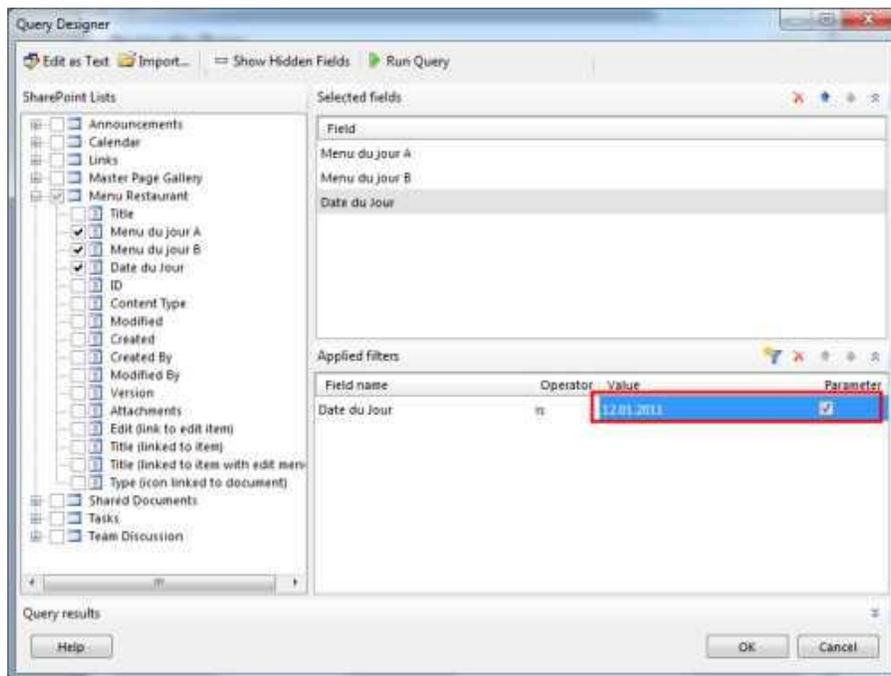
A ce moment, nous devons sélectionner les 3 colonnes de la liste, puis cliquer sur "Add Filter"



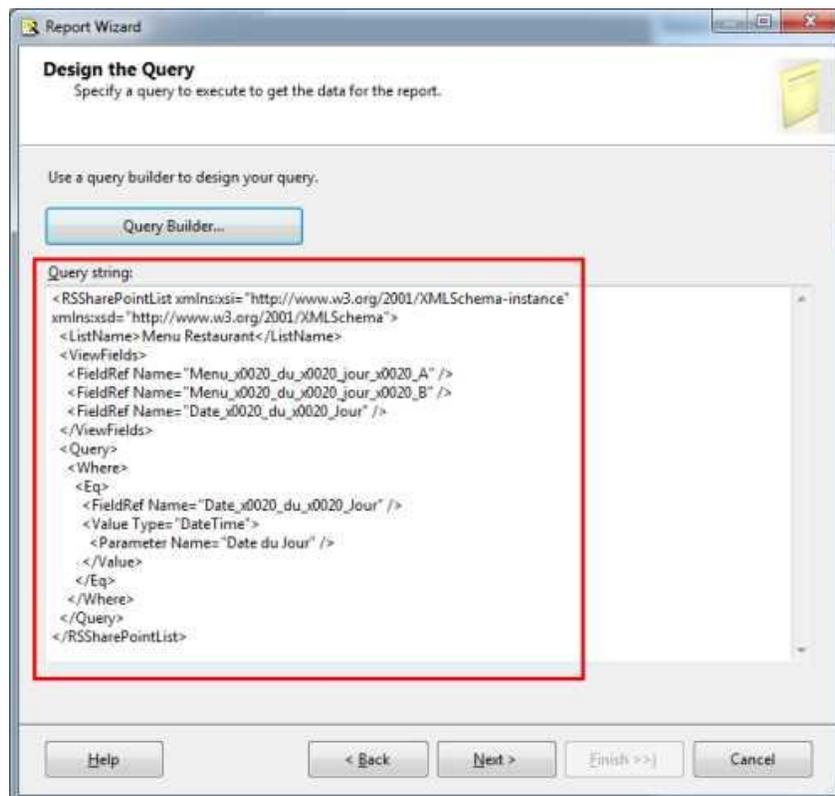
On voit une ligne apparaît dans la zone du bas avec le premier des trois noms. Il faut donc cliquer sur ce nom afin de sélectionner "Date du Jour"



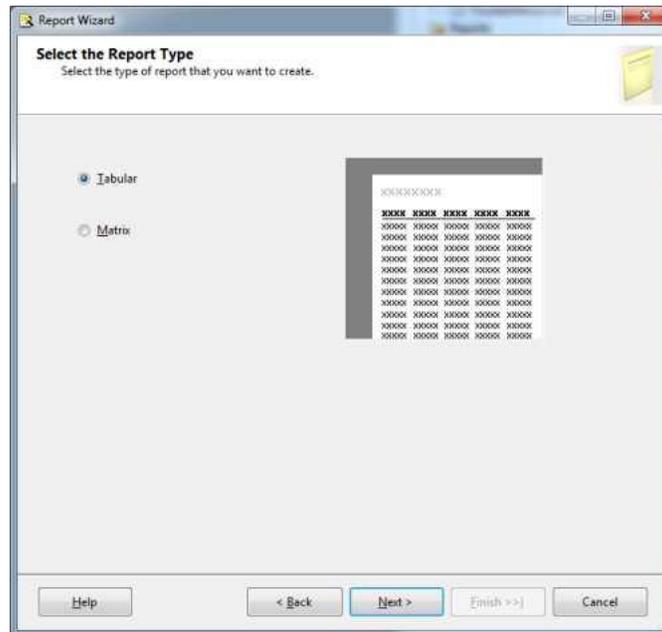
On ajoute alors la date du jour dans la valeur et coche la case "Parameter"



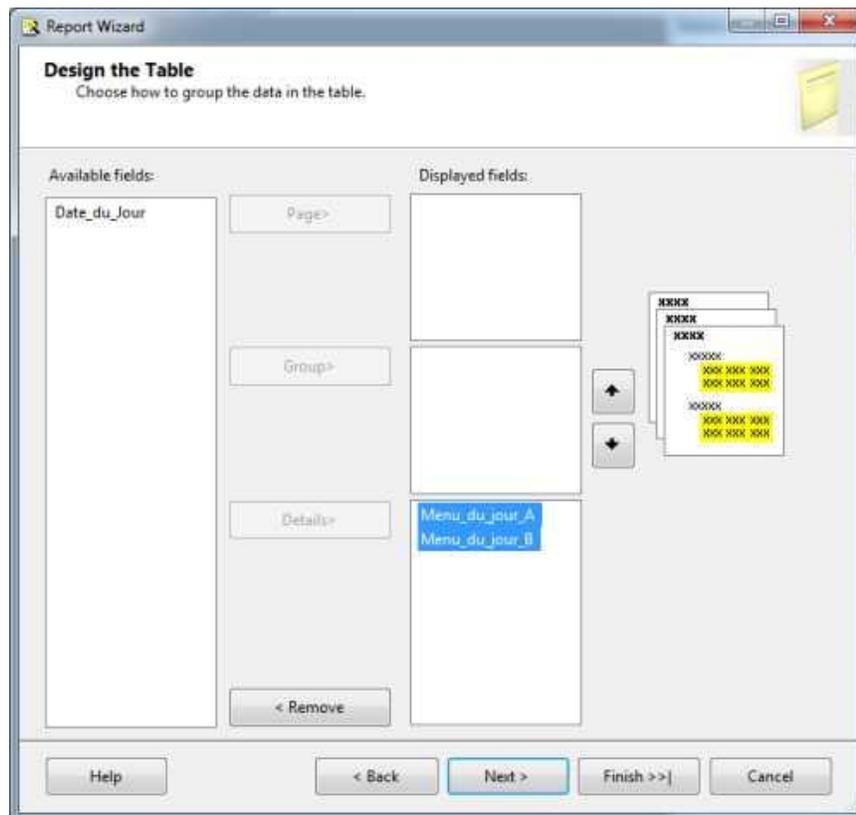
On peut alors valider par OK et voir apparaître la requête CAML



On peut alors cliquer sur "Next", pour ensuite choisir l'affichage en mode Tabulaire



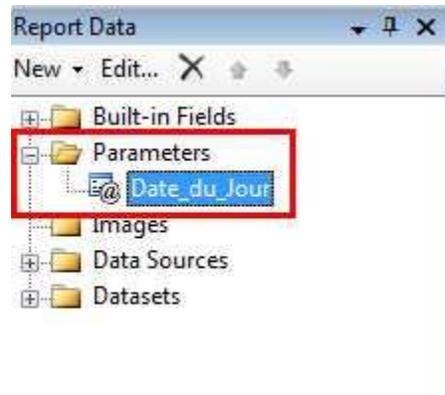
On ajoute maintenant les deux menus dans l'affichage des détails et on clique sur suivant



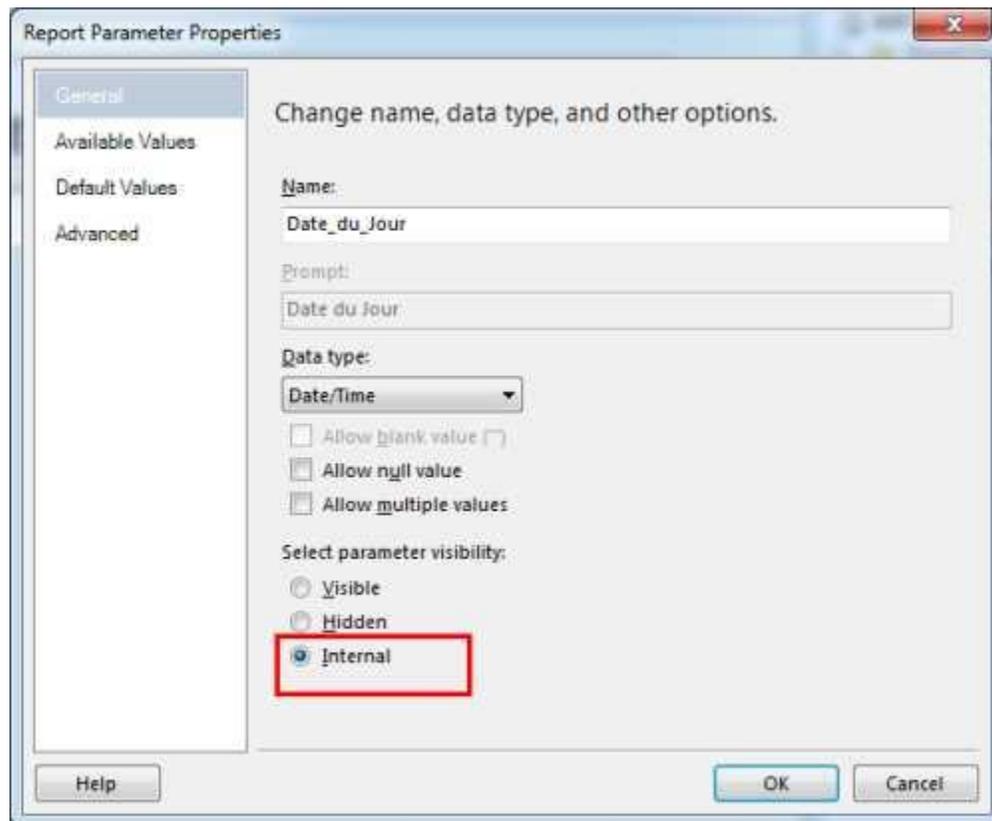
On choisit le même style que précédemment "Corporate" et on donne un nom à notre rapport "Menu_Today" pour enfin cliquer sur OK

Notre rapport est maintenant créé, mais affiche le menu d'une date fixe, il nous faut donc rendre ce paramètre invisible et surtout automatique avec la valeur du jour.

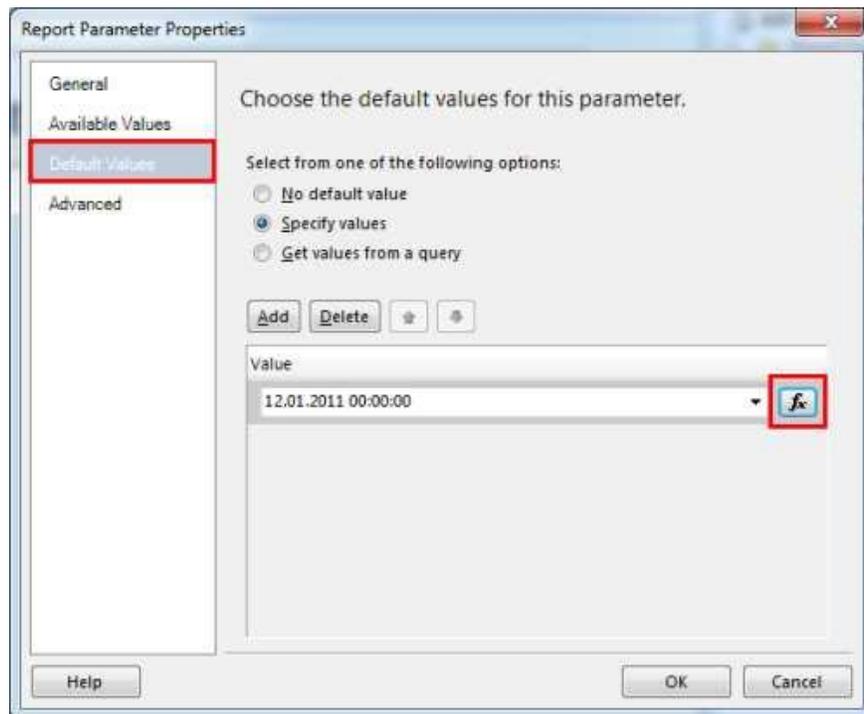
Pour cela, on va ouvrir les propriétés de notre paramètre en double cliquant sur "@Date_du_Jour"



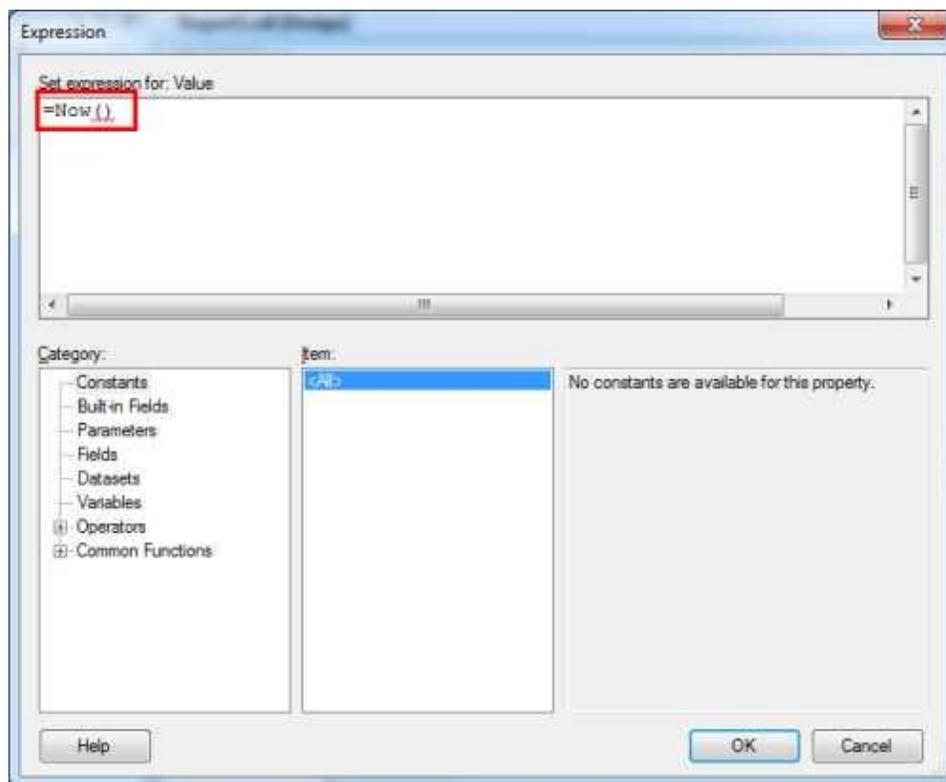
Dans la partie générale, on définit la visibilité comme "Internal" pour qu'il ne soit plus visible



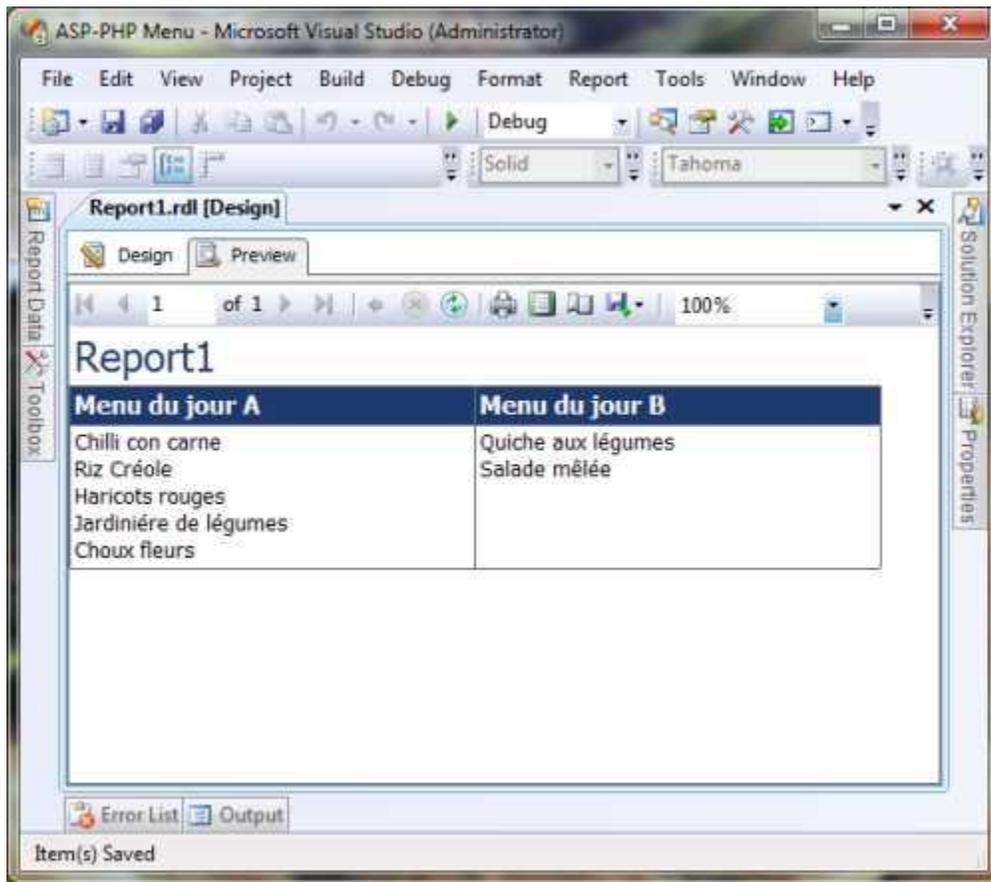
Il faut ensuite aller dans la définition de la valeur par défaut qui doit donc être dynamique avec la valeur du jour courant. Pour cela on clique sur la rubrique "Default Values" et dans la case où on voit apparaître la date tapée manuellement "12.01.2011 00:00:00", on clique sur le bouton "fx" à droite



On ouvre alors l'éditeur de fonction de Reporting Services, et dans la partie expression, on doit taper "=now()". On valide ensuite par OK



On peut donc modifier les paramètres d'affichage pour les valeurs de menu (Exécuter le code HTML) comme expliqué précédemment, adapter le design comme souhaité et enfin visualiser le résultat



Le rapport est donc terminé et peut être publié sur le serveur de reporting.

Rapport avancé Géospatial

Le rapport montré jusqu'ici est très basique et même si on utilise SharePoint comme source de données, il n'exploite en rien une des nouveautés de SQL Server 2008 R2 Reporting Services.

Voyons maintenant comment on peut coupler les deux solutions en affichant sur une Map Monde, la répartition des données provenant d'une liste de contacts.

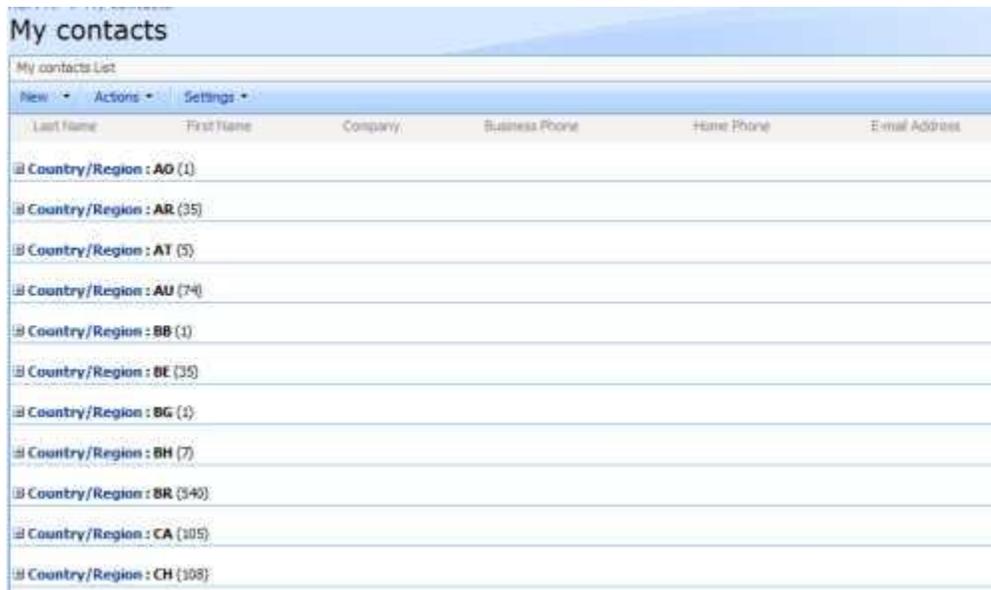
Pour cela, nous allons utiliser une liste SharePoint de type contact qui contient un ensemble de contacts avec le nom et l'adresse de chacun.

Dans une liste SharePoint de type Contact, une colonne de base existe "Country/Region", cette colonne est de type libre et il suffit que la donnée chargée dans cette colonne soit définie avec soit le nom du pays soit son code ISO à 2 lettres (bien plus fiable avec le code ISO).

Il existe d'ailleurs un script PowerShell permettant de charger dans une liste de ce type les comptes Active Directory :

- **[CodePlex - SharePoint Management PowerShell scripts : Set-WSS-Contact-List-From-ActiveDirectory](#)**

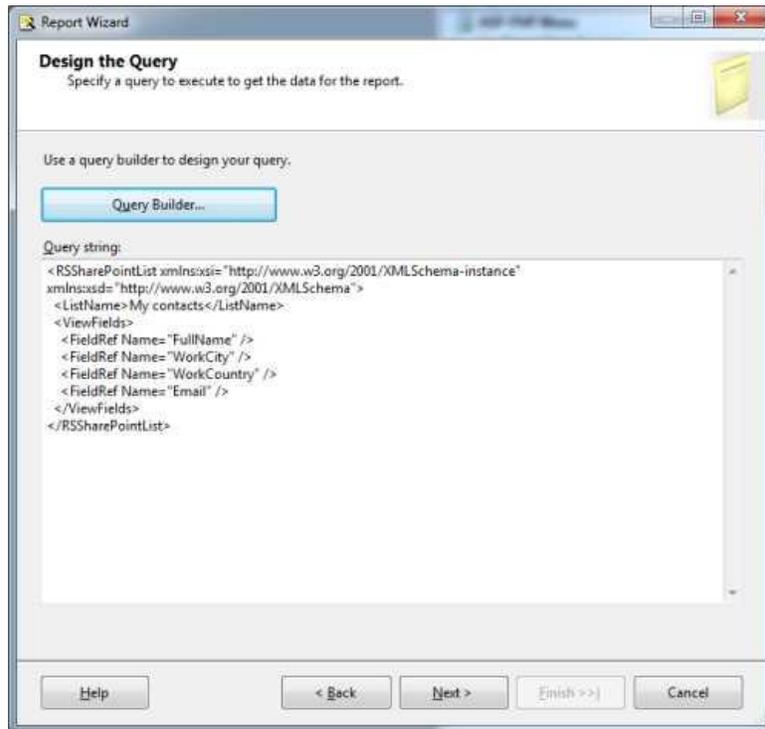
Ainsi, lorsque la liste est chargée avec les contacts, nous pouvons créer une vue groupant les contacts par Code Pays



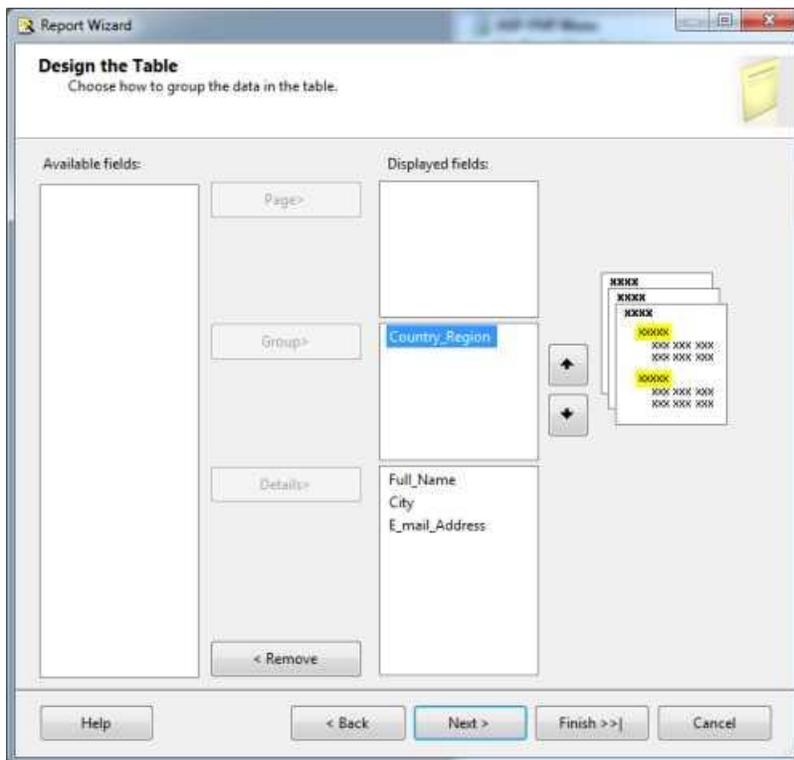
The screenshot shows a SharePoint list view titled "My contacts". The list is grouped by "Country/Region". The columns visible are Last Name, First Name, Company, Business Phone, Home Phone, and E-mail Address. The groups are as follows:

Country/Region	Count
AO	1
AR	35
AT	5
AU	74
BB	1
BE	35
BG	1
BH	7
BR	340
CA	105
CH	108

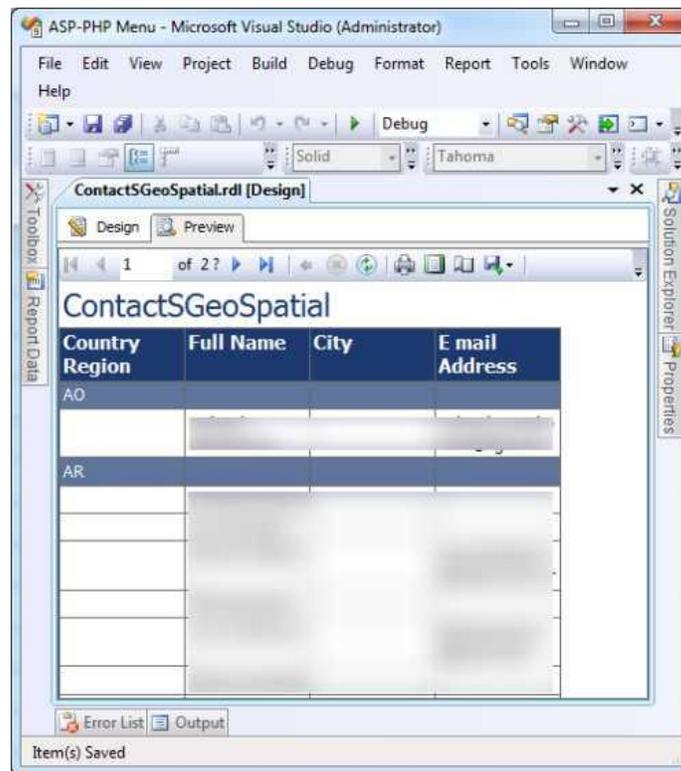
Il faut donc aller dans Visual Studio pour créer un nouveau rapport basé sur cette liste de contacts avec un affichage simple des contacts dans un tableau. Ainsi, nous sélectionnons les champs FullName, City, Country/Region et Email



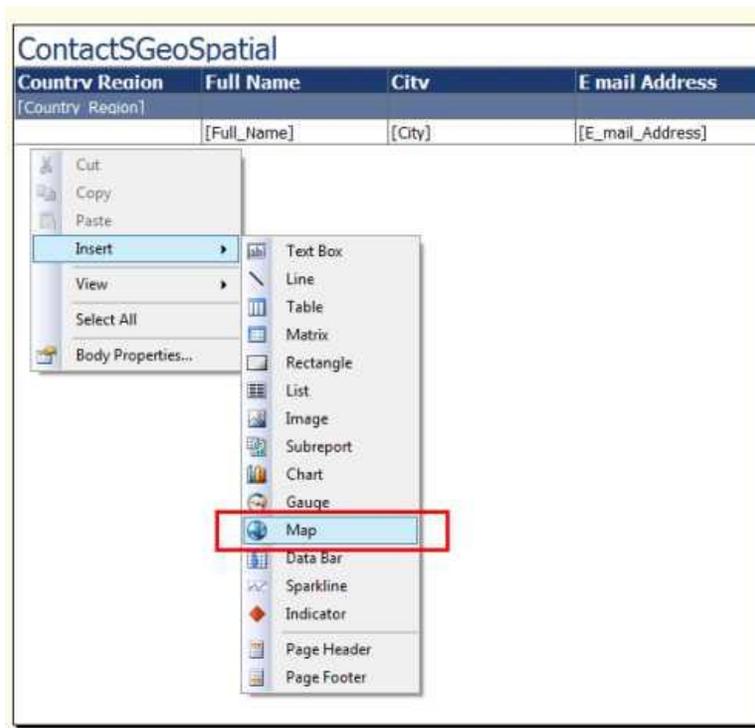
Puis prendre un rapport en mode tabulaire dont l'affichage va être groupé par région et on ajoute les sous-totaux



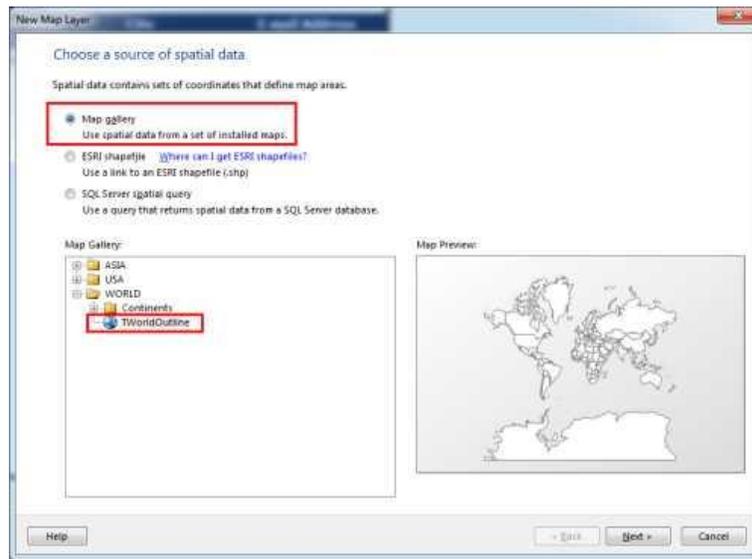
Le rapport est maintenant terminé et un rapide test permet de valider son bon fonctionnement



A ce niveau rien de très différent aux exemples précédents. On va donc cacher le tableau (propriété "Hidden" de l'objet "Table1"), puis on ajoute un objet de type "Map" (Bouton droit dans le rapport, "Insert > Map")



Un assistant se lance afin de nous guider dans le processus de configuration de la carte, Nous devons pour notre exemple, choisir "Map gallery" et sélectionner dans la liste du bas la carte "TWorldOutline".



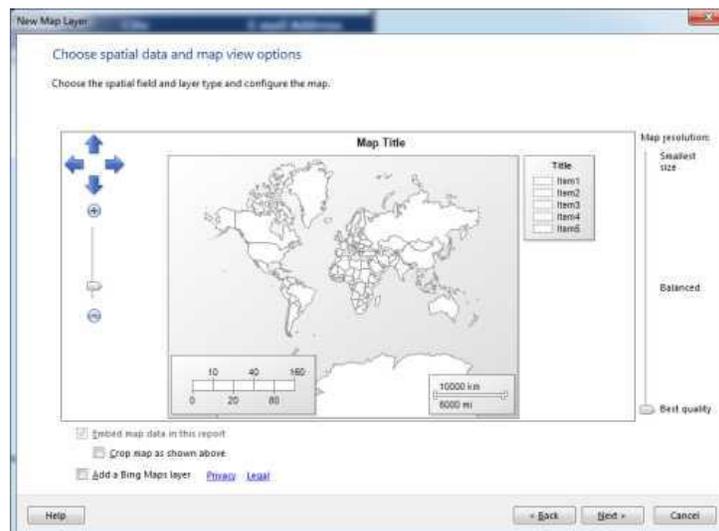
ATTENTION

De base, Visual Studio contient peu de cartes, une carte pour le monde et quelques-unes pour les Etats-Unis. Si vous souhaitez en ajouter de nouvelles, vous pouvez trouver un ensemble de cartes en ligne sur CodePlex :

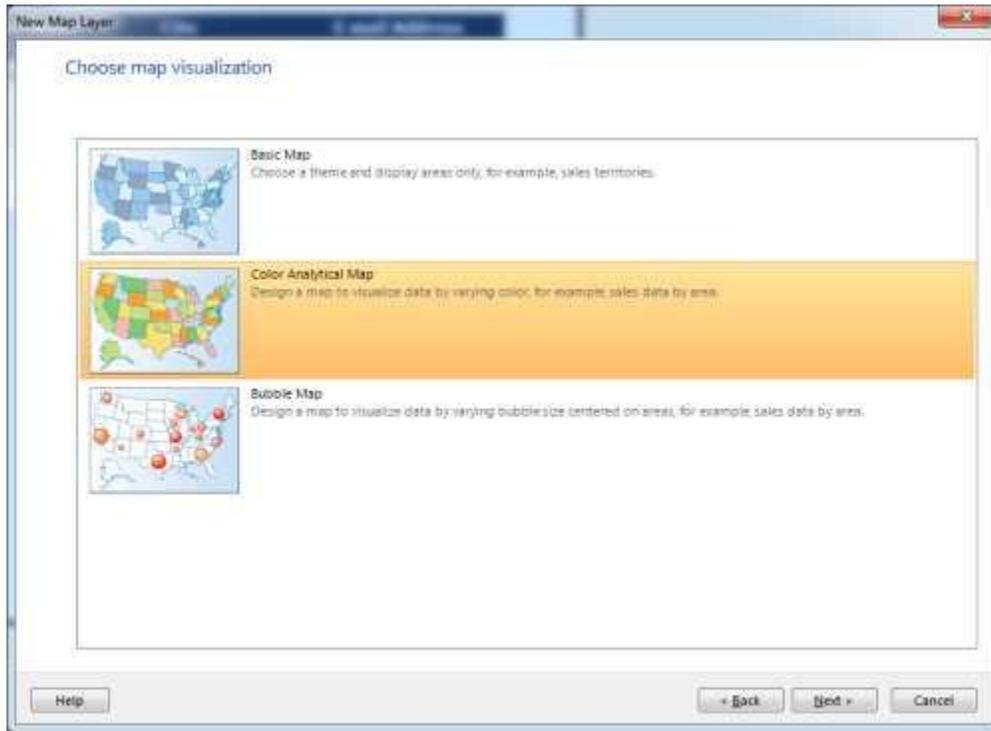
- [MapGallery of Reporting Services in SQL Server 2008 R2](#)

On créer vos propres cartes en utilisant des données provenant de différentes sources (Google Earth par exemple). Ceci n'est pas dans le cadre de cet article.

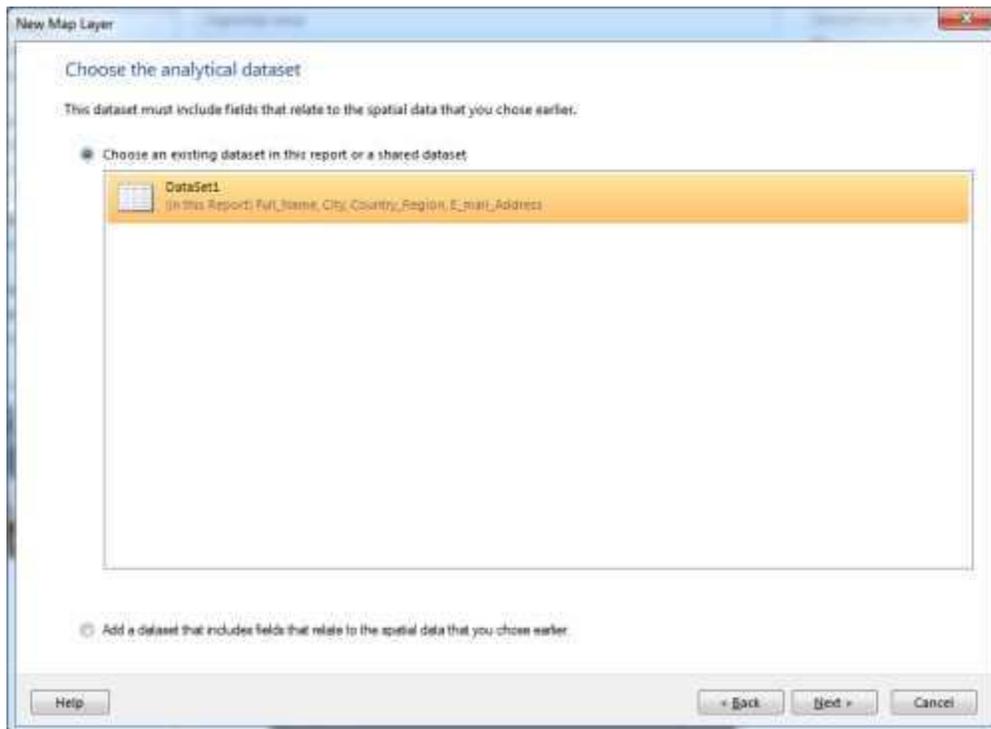
L'assistant vous propose alors de positionner au mieux la carte selon ce que vous souhaitez montrer (par exemple, uniquement une zone ou un pays). Vous pouvez aussi faire un couplage avec Bing Maps si vous le souhaitez, mais cela nécessitera une connexion internet à tous les postes exécutant le rapport. Pour cet exemple, nous laissons les valeurs par défaut, et cliquons sur "Next"



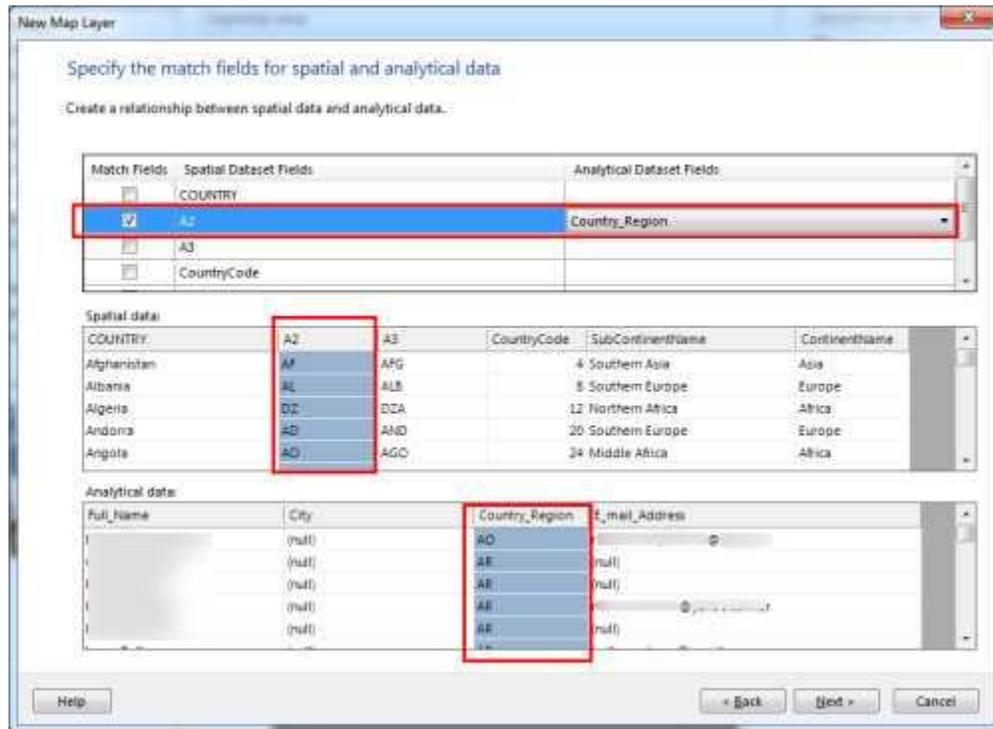
L'assistant nous demande le type de visualisation que l'on souhaite avoir dans notre rapport (dégradé de couleurs, taille de points ou jeu de couleurs spécifique), nous prendrons l'option "Color Analytical Map"



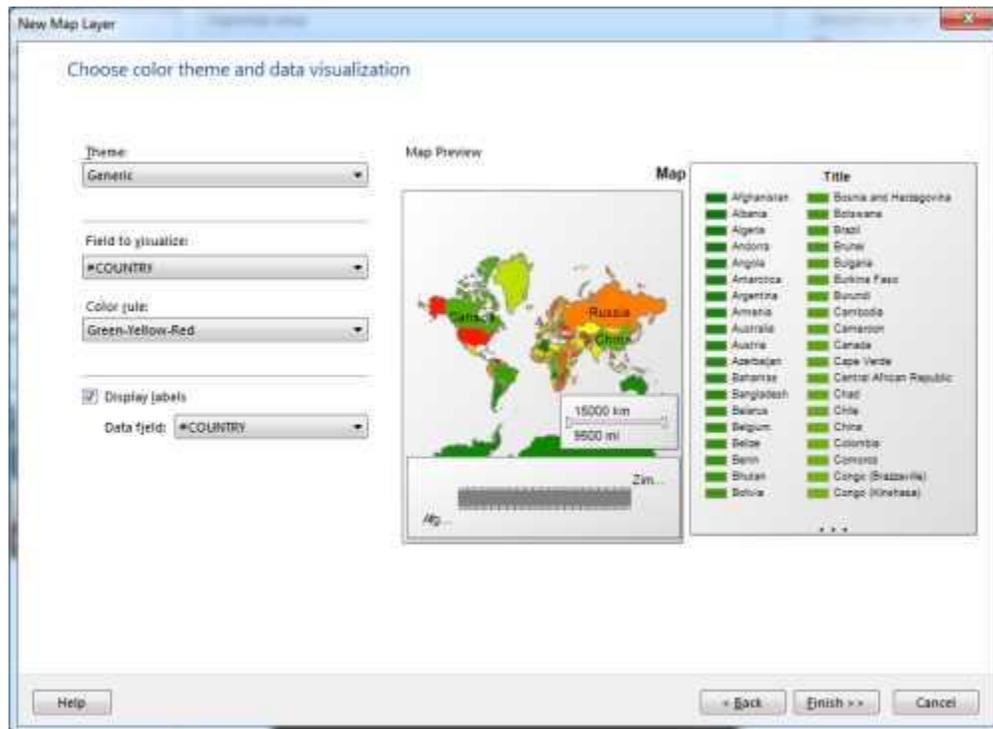
Le but de ce type de graphique est de faire une jointure entre la carte choisie et la source de données. Il faut donc choisir la source de données adaptée provenant de notre rapport (DataSet1 dans notre exemple), puis cliquer sur Next



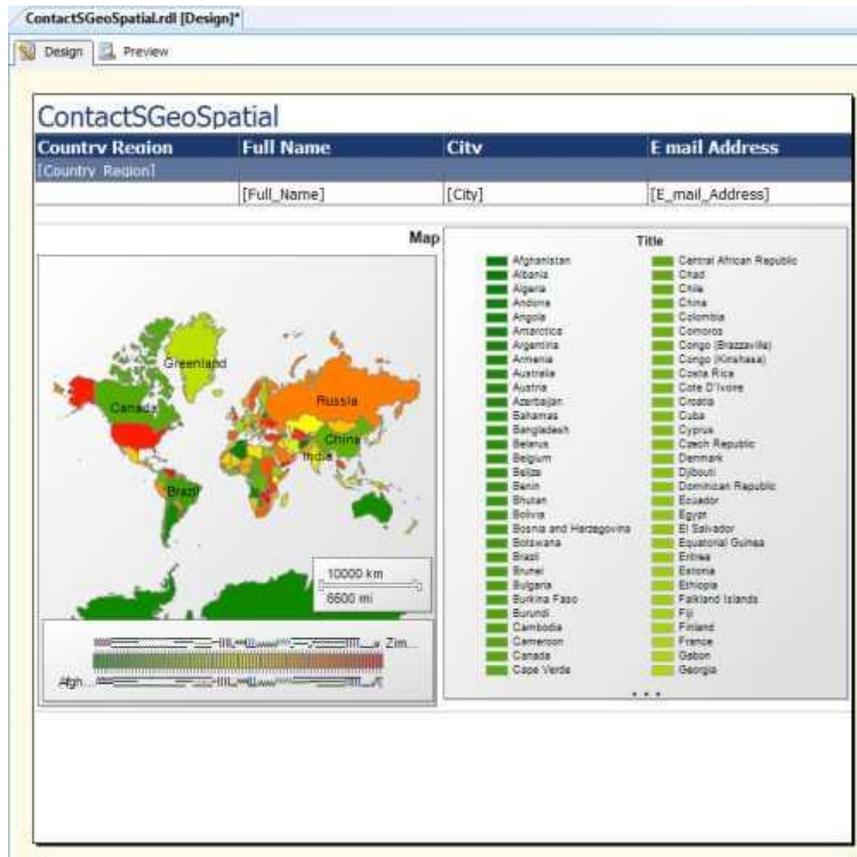
A ce stade, l'assistant nous demande de spécifier quelle va être la jointure à appliquer. Ce qui correspond à choisir les champs qui concordent entre la source de données et la carte. Pour notre exemple, il faut choisir les deux colonnes qui proposent le code pays ISO deux lettres et cliquer sur Next



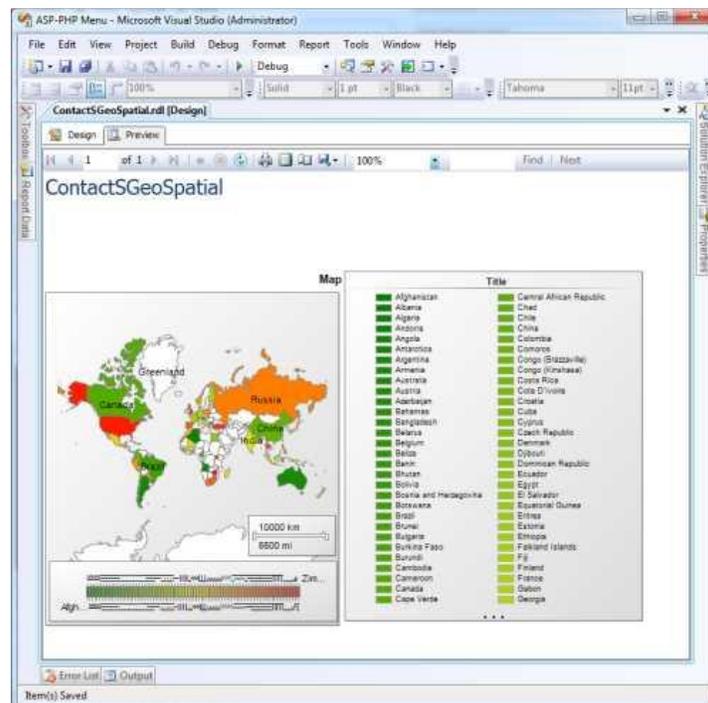
L'assistant propose maintenant le style graphique à appliquer autour de cette carte, le type de dégradé à appliquer, mais aussi l'affichage ou non des labels (code ou nom de pays) et on clique enfin sur "Finish"



Notre carte est alors ajoutée dans le fond de page de notre rapport

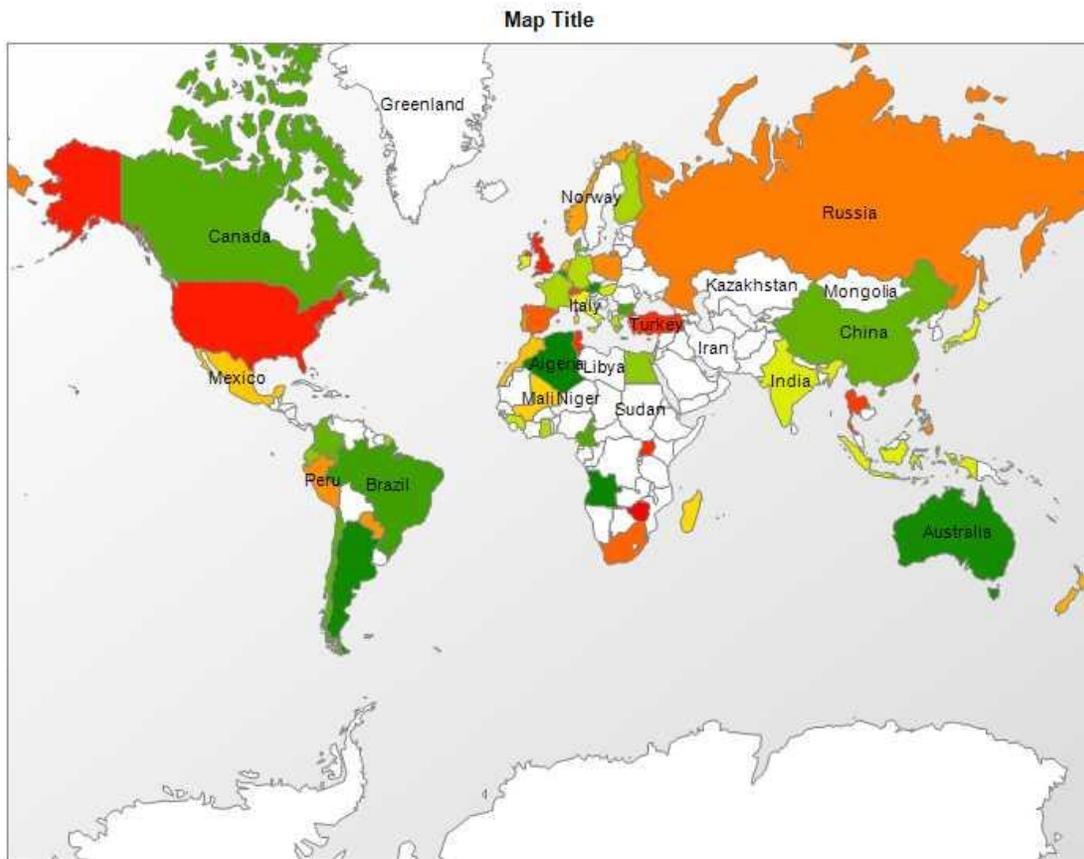


On peut visualiser le rapport dans le mode preview



On peut finir par un peu de personnalisation du rapport en supprimant l'affichage des titres de pays sur la droite et supprimant les références de couleur et de distance sur le bas et on obtient le résultat suivant

ContactSGeoSpatial



Je vous laisse découvrir toutes les options de personnalisation de cette carte afin de créer votre propre rapport. Il ne vous reste plus qu'à trouver des cas pratiques afin d'exploiter cette solution.

Conclusion

Nous avons abordé dans cet article une des grandes nouveautés fournies avec SQL Server 2008 R2. Le support natif de SharePoint comme source de données pour les rapports.

Ceci doit tout de même respecter quelques règles simples :

- Support uniquement de l'authentification Windows
- Support uniquement de SharePoint 2007 et 2010

La première règle implique donc que l'utilisateur qui souhaite afficher le rapport doit avoir au minimum les droits d'accès en lecture à la liste.

Ceci exclut immédiatement cette solution pour fournir des rapports de consolidation sans que l'utilisateur ne puisse avoir les droits sur le détail des données, comme par exemple les consolidations financières.

De la même façon, le support uniquement des versions supérieures à 2007 est une contrainte à valider lors de l'implémentation de cette solution.

A cela, il faut ajouter encore la possibilité de jointure entre sources de données. Comme nous l'avons remarqué dans les exemples, le choix de la liste et la règle de sélection des données est très basique, il est donc difficile d'ajouter une notion de listes jointes dans la sélection des données (choses très simples dans une base de données).

Si vous tombez dans ces exigences, il est préférable de choisir une solution tierce qui propose encore plus d'options dans la création de rapports donc les données proviennent de SharePoint :

- **Enesys RS Data Extension**

Ce produit est d'ailleurs en évolution constante avec de très nombreux ajouts dans chaque version. On peut d'ailleurs noter l'utilisation de SQL Lite dans la sélection des données ce qui est très utile pour les jointures justement.

Voici quelques liens utiles si cet article vous a intéressé :

- [Reporting Services et les données stockées dans SharePoint](#)
- [Get Started with Chart Development in Reporting Services 2008](#)
- [TechEd Online : SSRS 2008R2 Integration With SharePoint 2010](#)
- [Using a SharePoint List as a Data Source in SQL Server Reporting Services 2008 R2](#)
- [Collaboration : Intégration de SQL Server 2008 R2 Reporting Services dans SharePoint 2010](#)
- [SQL Server 2008 R2 Reporting Services, SharePoint List Data Source](#)
- [SQL Server 2008 R2 Reporting Services – The World is But a Stage \(T-SQL Tuesday #006\)](#)
- [Reporting Services 2008 R2: Geospatial Visualization – Part I](#)
- [Country Files \(GNS\)](#)
- [Map Data Links](#)
- [MapGallery of Reporting Services in SQL Server 2008 R2](#)
- [KML to Shapefile File Conversion \(Zonum\)](#)
- [Google Earth to ShapeFile - KML2SHP Online](#)
- [CodePlex - SharePoint Management PowerShell scripts : Set-WSS-Contact-List-From-ActiveDirectory](#)
- [Enesys RS Data Extension](#)
- [Office Online](#)
- [Club SPS MOSS FRANCE\(FR\)](#)

En vous souhaitant de bons projets de développement.

Romelard Fabrice (alias F___)

Intranet/Extranet CTO - **SGS**