

Microsoft®

# InfoPath®



## Notes de cours InfoPath 2003 à 2010

---

Vincent ISOZ, 2014-02-20 (V3.0 Revision 9)  
{oUUID 1.681}

**Microsoft®**



**Remarques:**

Pour qu'il soit utilisable d'une manière rationnelle et sans danger, ce support qui constitue un "super condensé" d'un exposé qui tiendrait très facilement sur plusieurs milliers de pages (voir les ouvrages de cette taille disponible sur le commerce) et qui constitue une suite logique de mes livres sur MS Access, le XML et SharePoint doit absolument être complété par de nombreuses notes et exposés oraux, au cours desquels les notions nouvelles sont présentées au moyen de situations concrètes et illustrées par de nombreux exemples dont le choix dépend essentiellement du déroulement de la formation afin d'exciter l'esprit critique des apprenants. Ce support correspond à une formation d'environ 15 jours à 6.5 heures par jour pour un groupe de 6 personnes.

Il y a de nombreuses marques déposées qui sont nommées dans le présent support. Plutôt que d'utiliser le symbole du trademark sur chaque occurrence de marque nommée, j'ai choisi d'utiliser le nom seul uniquement dans un souci d'esthétique éditoriale (ce qui devrait aussi bénéficier au propriétaire de la marque), sans aucune intention de violer une quelconque réglementation ou législation.

Pour terminer, je voudrais remercier ici les quelques collègues et clients qui ont bien voulu me faire part de leurs remarques pour améliorer le contenu de ce livre électronique. Il est cependant certain qu'il est encore perfectible sur de nombreux points.

## TABLE DES MATIÈRES

1. Liens Internet.....	5
2. Add-Ins.....	6
1. À propos de l'auteur.....	7
3. Références .....	8
4. Introduction .....	9
4.1 Formulaire prêts à l'emploi .....	15
4.2 Les affichages.....	16
4.3 Workflow.....	18
4.4 Développement.....	19
5. Dangers.....	22
6. Limites.....	23
7. InfoPath vs Webforms.....	25
8. Compatibilité InfoPath/InfoPath Services.....	26
8.1 Version 2007 .....	26
8.2 Version 2010 .....	30
9. Applications pratiques (sans dév.) .....	31
9.1 XML, XSL, XSD, XSF .....	31
9.2 Formulaire simple (notes de frais).....	37
9.2.1 Types de contrôles.....	38
9.2.1.1 Zone de texte .....	40
9.2.1.2 Contrôles de Section et Section optionnelle.....	43
9.2.1.3 Contrôles Tableau extensible et Section extensible .....	46
9.2.1.4 Contrôle Section récursive extensible .....	46
9.2.2 Utiliser les Regular Expression .....	47
9.2.3 Liste LookUp.....	49
9.2.4 Publier versus Enregistrer .....	53
9.2.4.1 Protection contre la conception .....	54
9.2.4.2 Publication.....	56
9.2.4.3 Versions des formulaires .....	58
9.2.5 Vider le cache.....	62
9.2.6 Signature.....	63
9.2.7 Export d'un formulaire vers MS Word.....	70
9.2.8 Sélecteur de contact (Custom control) .....	71
9.2.9 Media Player (Custom control) .....	83
9.3 Formulaire XSD .....	87
9.4 Ajout d'une liste avec les utilisateurs SharePoint.....	93
9.5 Suppression des doublons d'une liste déroulante .....	99
9.6 Filtrage d'une section répétitive sur des sélections antérieures .....	102
9.7 Formulaire de saisie connecté (MS Access).....	105
9.8 Formulaire de saisie connecté (MS Access/SharePoint).....	115
9.9 Formulaire de requête connecté (MS Access).....	131
9.10 Désactivation conditionnelle de boutons.....	134
9.11 Assignation d'une numérotation automatique à un champ.....	136
9.12 Interdire la saisie de doublons sur un champ .....	137
9.13 Utilisation des rôles .....	139
9.14 Protection contre la modification .....	146

9.15	Utilisation d'InfoPath avec Outlook .....	148
9.16	Formulaires multi-langues.....	161
10.	Applications pratiques (avec dév.) .....	171
10.1	Utilisation de JS comme calculatrice .....	171
10.2	Utilisation de JS pour la validation des données.....	171
10.3	Utilisation de ADO avec JS.....	172
10.4	Création d'une boîte de dialogue de sauvegarde avec C# .....	177
10.5	Peupler une liste avec les utilisateurs SharePoint et C#.....	184
10.6	Utilisation d'un webservice simple en C# .....	191
10.7	Utilisation d'un webservice en C#.....	197
11.	Applications pratiques (pour SharePoint) .....	214
11.1	Récupérer le nom de l'opérateur de saisie du formulaire à la prochaine ouverture du formulaire (v2010) .....	214
11.2	Personnalisation du Data Information Panel (v2007) .....	223
11.3	Création de formulaires InfoPath avec flux SharePoint Designer (v2007).....	232
11.4	Publier sur InfoPath Services (v2007).....	257
11.5	Boîte de message à la soumission dans InfoPath Services (v2007).....	266
11.6	Mise à jour liste SharePoint suite à modification de structure d'un formulaire InfoPath .....	273
11.6.1	InfoPath/SharePoint 2007 .....	273
11.6.2	InfoPath/SharePoint 2010 .....	276
11.7	Publier un formulaire en tant que Content Type (v2007) .....	278
11.8	Création d'une liste déroulante en cascade (v2007) .....	284
11.9	Ajout d'un élément à une liste SharePoint (v2003) .....	302
11.10	Incrémenter automatiquement un nom de fichier formulaire InfoPath avec l'ID de la librairie SharePoint (v2010) .....	309

## 1. Liens Internet

Comme InfoPath va probablement disparaître, nous revoyons nos lecteurs à la solution de remplacement que je considère actuellement comme étant la meilleure sur le marché à ce jour:  
<http://www.nintex.com/en-US/Products/Pages/NintexForms.aspx>

Sinon voici d'autres liens:

<http://www.infopathdev.com>

Super site de référence sur le logiciel avec des tutoriaux, des plug-ins à télécharger, des trucs et astuces et j'en passe...

<http://www.bizsupportonline.net>

Excellent site pour apprendre InfoPath avec des vidéos ou encore en achetant les livres qui les accompagnent (livres qui au passage sont considérablement plus complets que le présent support de notes de cours et que tout ce que l'on peu trouver en français sur le sujet).

<http://www.developpez.net/forums/f574/hardware-systemes-logiciels/microsoft-office/infopath>

Forum InfoPath francophone de Developpez.com

<http://msdn.microsoft.com/office/understanding/infopath/training/default.aspx>

Webcasts et Labs de Microsoft sur leur produit InfoPath

<http://www.learnthat.com/computers/learn.asp?id=1878&index=0>

Très bon tutorial InfoPath

<http://www.microsoft.com/downloads/details.aspx?FamilyId=7E9EBC57-E115-4CAC-9986-A712E22879BB&displaylang=en>

Lien vers le téléchargement du Toolkit de développement VS.Net 2003 pour InfoPath

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odc\\_2003\\_ta/html/odc\\_landinfo03\\_tr.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odc_2003_ta/html/odc_landinfo03_tr.asp)

Lien vers les labos InfoPath de Microsoft

<http://www.infoscope.co/kr>

Un très bon viewer de formulaires InfoPath (pour ceux n'ayant pas la version 2010)

Vincent ISOZ

## **2. Add-Ins**

<http://www.qdabra.com>

Add-In pour avoir les métadonnées gérées relativement facilement dans InfoPath

## 1. À propos de l'auteur



Nom Prénom: ISOZ Vincent  
Domicilié à ce jour à Lausanne (Suisse)

Formation: Ingénieur Physicien HES (B.Sc.)  
Équivalence Internationale: Bachelor of Science

Année de naissance: 1978

Actuellement, je suis consultant trilingue (français, anglais et allemand) en mathématiques appliquées dans le tutorat d'analystes quantitatifs (niveau Bac+5 à Bac+7) et auteur de plusieurs livres électroniques dans les domaines suivants:

- maîtrise statistique des processus/procédés (méthodes paramétriques et non paramétriques)
- modélisation prévisionnelle/décisionnelle avancée (arbres de décisions, chaînes de Markov)
- recherche opérationnelle (simplexe, algorithmes génétiques, algorithme GRG)
- data mining (réseaux de neurones, ACP, AFC, régressions, scoring, clustering, etc.)
- modélisation du risque en gestion de projets et finance d'entreprise (monte-carlo, etc.)
- gestion de projets (modèles et best practices théoriques EFQM+Six Sigma, MS Project)
- ISO 9001:2008, 5807:1985, 10015:1999, 31000+31010:2009, 8258:1991, 10017:2003, etc.
- Adobe Photoshop et Illustrator
- 12 applications de la suite Microsoft Office System (Project, Visio, SharePoint, Access, etc.)

À ce jour interventions dans plus de ~200 entreprises dont 10 du *Fortune 500* selon listing 2009 et 3 universités et écoles d'ingénieurs suisses dans des cours de modélisation de bases de données et simulations stochastiques du risque. Formation de plusieurs dirigeants de multinationales en one to one.

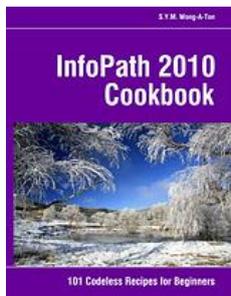
Accessoirement j'interviens pour des formations sur des logiciels comme MS Project, MS Visio, MS Access et une vingtaine d'autres dont je délègue l'organisation à des entreprises spécialisées dans la formation continue en bureautique (niveau licence et en-dessous).

Il est très fortement conseillé de planifier rigoureusement mon arrivée et le cahier des charges si vous souhaitez faire appel à mes services. Je suis effectivement très exigeant sur le respect des standards de la gestion de projets et des normes minimales du travail entreprise (ISO 9001, ISO 690, ISO 9660, ISO 5807, ISO 10015, etc.) et je n'hésiterai pas à vous dire franchement ce qui ne va pas dans votre organisation (je ne suis pas payé pour vendre un produit ou une méthode mais pour dire la vérité!). Je suis également très regardant sur les compétences des employés invités aux réunions que je dois piloter et les conditions d'accueil. Vous voilà prévenus si jamais!

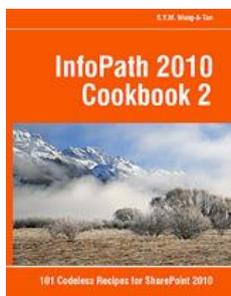
### 3. Références

Depuis le temps que MS InfoPath 2003 a été sorti, de plus en plus de personnes ce sont mises à écrire dessus et en particulier un fan du logiciel qui a écrit la (les) bible(s) d'InfoPath.

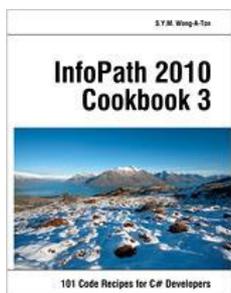
Comme ce logiciel est majoritairement dans son usage quotidien l'application de recettes de cuisine tellement il a été mal pensé... il a écrit des Cookbook sur le logiciel (ce qui rend les présent e-book totalement inutile tellement le travail de compilation qu'il a fait est de qualité):



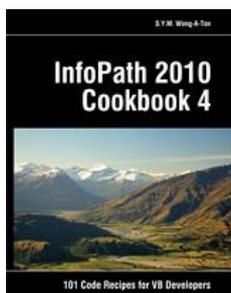
ISBN: 978-1456542764  
Publication Date: Feb 2011  
Author: S.Y.M. Wong-A-Ton  
Page Count: 284  
Language: English



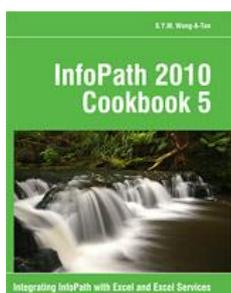
ISBN: 978-1463533243  
Publication Date: Oct 2011  
Author: S.Y.M. Wong-A-Ton  
Page Count: 478  
Language: English



ISBN: 978-1469988016  
Publication Date: Mar 2012  
Author: S.Y.M. Wong-A-Ton  
Page Count: 478  
Language: English



ISBN: 978-1478277392  
Publication Date: Aug 2012  
Author: S.Y.M. Wong-A-Ton  
Page Count: 478  
Language: English



ISBN: 978-1481962063  
Publication Date: Jan 2013  
Author: S.Y.M. Wong-A-Ton  
Page Count: 362  
Language: English

## 4. Introduction

Microsoft Office InfoPath est un programme (qui date de 2003) qui permet de créer et de remplir des formulaires commerciaux relativement puissants. Ce document offre principalement un aperçu d'InfoPath 2003/2007 en relation avec Outlook/Excel/Word/Access/SharePoint 2007 et 2010 (sachant que la version 2010 est encore beaucoup plus flexible).

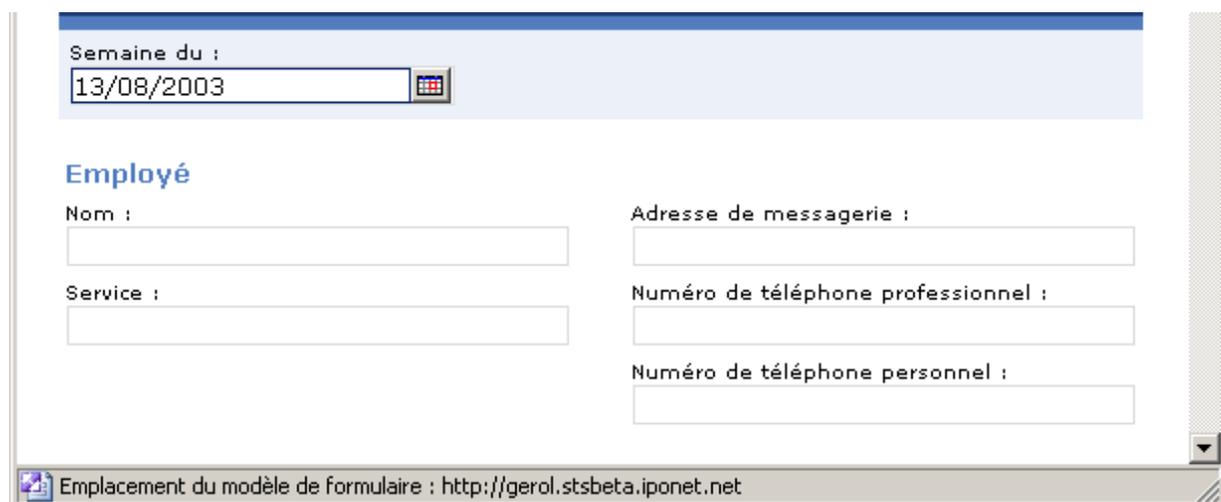
Comme la plupart des gens, vous remplissez probablement des formulaires professionnels, tels que des notes de frais ou des bons de commande, des fiches de présences, de temps, presque tous les jours. Vos responsabilités vous amènent peut-être également à créer des formulaires qui sont utilisés par d'autres personnes dans votre entreprise. Quelle que soit votre situation, vous pouvez utiliser Microsoft Office InfoPath afin d'améliorer les processus de gestion existants et accroître la productivité dans votre entreprise.

InfoPath permet, entre autres, d'éviter les erreurs d'entrée de données, d'assurer la cohérence de la mise en forme, de combiner (fusionner) ou d'exporter des données à partir de plusieurs formulaires et de générer différents affichages des données dans votre formulaire. Suivant votre niveau d'expérience, vous pouvez également utiliser InfoPath pour automatiser les processus de gestion, notamment concernant la soumission et le stockage des données de formulaire dans une base de données. Ceci en utilisant SharePoint comme portail d'accueil de vos formulaires.

Mais comment se présente un formulaire InfoPath ?

Ci-dessous un formulaire InfoPath. Il est présenté en trois écrans:

- Figure 1 Formulaire InfoPath Feuille de Présence A
- Figure 2 Formulaire InfoPath Feuille de Présence B
- Figure 3 Formulaire InfoPath Feuille de Présence C



The screenshot shows a web-based InfoPath form. At the top, there is a header bar with the text 'Semaine du : 13/08/2003' and a small calendar icon. Below this, the form is titled 'Employé' in blue. The form contains several input fields: 'Nom :', 'Adresse de messagerie :', 'Service :', 'Numéro de téléphone professionnel :', and 'Numéro de téléphone personnel :'. At the bottom of the form, there is a status bar with the text 'Emplacement du modèle de formulaire : http://gerol.stsbeta.iponet.net'.

Vincent ISOZ

Un formulaire est principalement un document qui contient des champs à remplir, comme illustrés ici. Les formulaires sont généralement utilisés pour recueillir, compiler et distribuer des informations.

InfoPath offre des avantages distincts par rapport aux formulaires Web ou à d'autres solutions pour recueillir des informations. Dans ce document, nous allons étudier ces avantages et déterminer comment ils peuvent vous permettre de gagner du temps et d'accroître votre productivité.

**Formulaire1 - Microsoft Office InfoPath 2003 Beta**

Fichier Edition Affichage Insertion Format Outils Tableau ? Tapez une question

Verdana 10 G I S

### Rapport de présence

**Remarque :** toutes les heures supplémentaires doivent être approuvées au préalable par votre responsable.

Semaine du : 13/08/2003	Lun.	Mar.	Mer.	Jeu.	Ven.	Sam.	Dim.
Heure d'arrivée							
Heure de départ							
Heure d'arrivée							
Heure de départ							
Équipe							
Sous-total							

### Rapport de présence

Congés maladie							
Congés							
Jours fériés							
Congés sans solde							
Congés de décès							
Sous-total							

**Total heures consignées**

Emplacement du modèle de formulaire : <http://gerol.stsbeta.iponet.net>

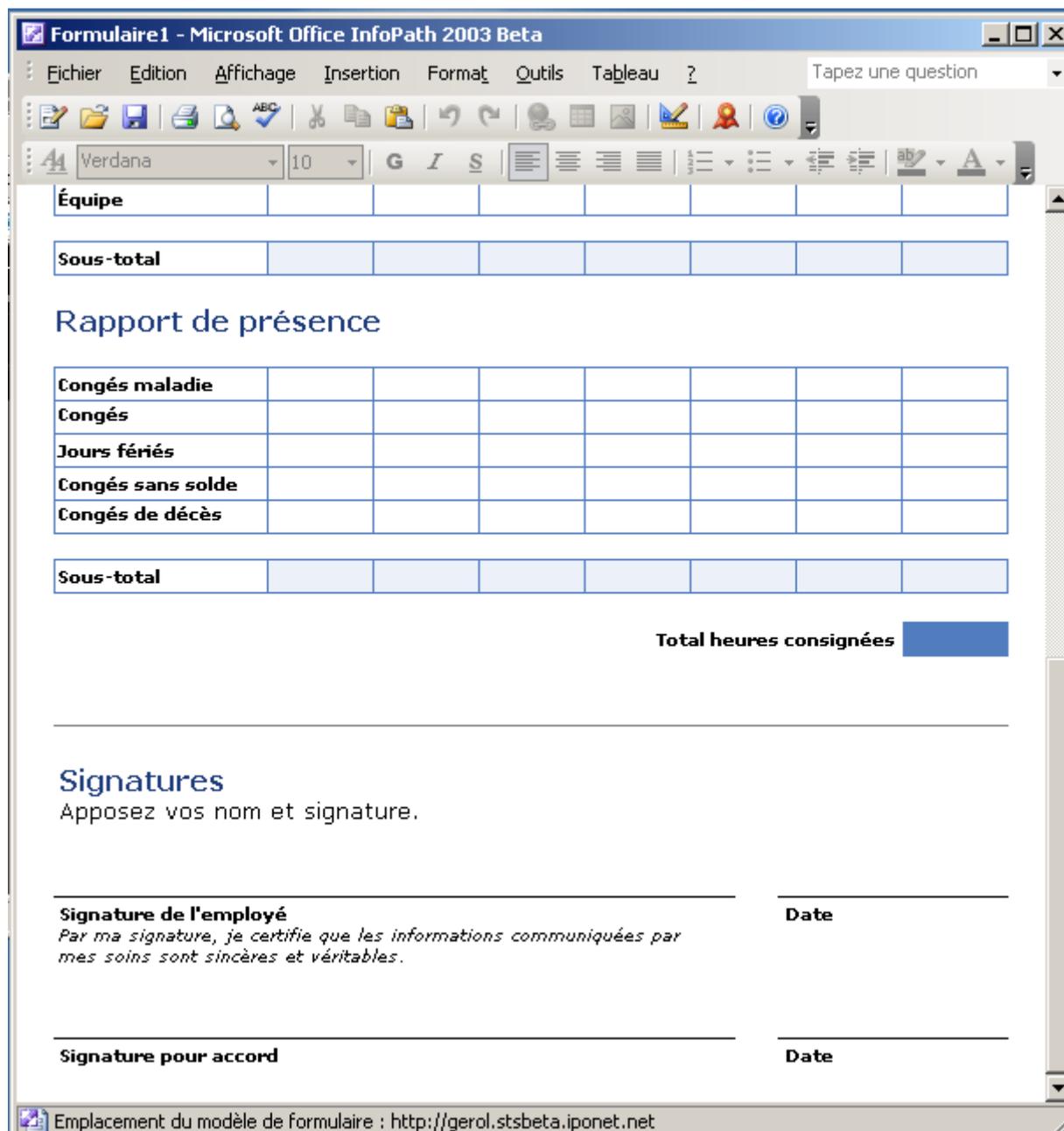
Tout comme les formulaires papier traditionnels, les formulaires InfoPath servent à collecter et à partager des données avec d'autres personnes. Et comme tout autre formulaire, un formulaire InfoPath contient généralement des espaces blancs à compléter (ou champs de formulaire) dans lesquels vous entrez des informations.

Vincent ISOZ

La différence est que les données stockées dans un formulaire InfoPath ne restent pas bloquées dans le formulaire pour toujours, mais peuvent être facilement séparées du formulaire et remises en forme ou réutilisées de différentes manières, et ce, parce que les données sont enregistrées dans un format de fichier hautement flexible, tel qu'un fichier XML (Extensible Markup Language).

Pour illustrer cette idée, imaginez que vos vendeurs vous soumettent chaque mois des rapports de ventes. Bien qu'il soit utile d'examiner les données de chaque rapport de ventes individuellement, vous pouvez également fusionner les rapports en un seul rapport de ventes à envoyer à votre responsable. Vous pourriez également exporter (sous certaines limites et contraintes) les données de ventes des rapports dans une feuille de calcul Microsoft Excel en vue d'établir des prévisions de ventes.

De plus dans la troisième partie vous pouvez voir la signature pour accord:



À moins de travailler dans un domaine lié à l'informatique, les trois lettres XML ne vous disent certainement pas grand chose. Bien qu'InfoPath soit basé sur la norme XML, il est important de savoir que vous pouvez créer et remplir des formulaires sans la moindre connaissance en XML. Dès lors, au lieu de s'attacher aux transformations des fichiers XSL et des fichiers de schéma XML, vous pouvez vous concentrer sur la réalisation de votre travail.

Le domaine du XML étant un sujet relativement complexe, il ne sera pas abordé plus en détail dans le cadre de ce cours préliminaire (voir mon livre électronique sur sujet). Si vous souhaitez en savoir plus sur la manière dont InfoPath utilise le format XML, reportez-vous à l'aide d'InfoPath.



Les personnes d'un groupe de travail utilisent un modèle de formulaire unique pour créer des formulaires commerciaux structurés. Cette structure sous-jacente, telle que définie par le modèle de formulaire, garantit que les données du formulaire sont entrées, mises en forme et collectées de manière cohérente.

Chaque formulaire InfoPath est basé sur un *modèle de formulaire*. Vous utilisez un modèle de formulaire pour créer plusieurs formulaires du même type, comme illustré ici.

Dans une petite entreprise, une personne est généralement chargée de créer, gérer et distribuer les modèles de formulaires (car c'est un métier!). Il peut s'agir de votre administrateur de groupe, chef de bureau ou de la personne qui gère l'intranet de votre société. Si vous travaillez pour une grande entreprise, votre équipe ou groupe de travail immédiat peut développer des modèles de formulaires, ou votre département informatique peut créer des modèles de formulaires à intégrer dans les processus de gestion existants de votre entreprise.

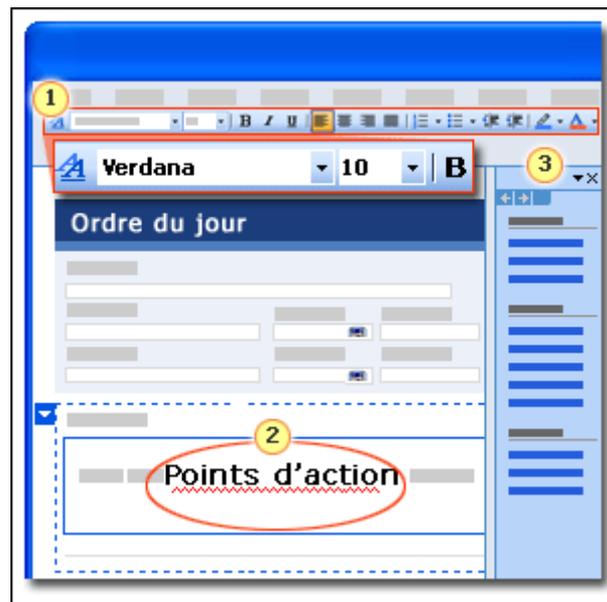
Un modèle de formulaire définit l'apparence et le comportement des éléments du formulaire, tels que les zones de texte et les cases à cocher. Il détermine également ce qui suit:

- *La structure du formulaire:* La structure fait référence à la disposition et à l'ordre des différentes sections du formulaire et à la manière dont elles sont liées. Enfin, la structure est importante car elle assure que les données du formulaire sont entrées, mises en forme et collectées de manière cohérente, et ce, pour une réutilisation ultérieure des données plus aisée.
- *Les sections du formulaire:* Un formulaire InfoPath contient généralement plusieurs sections. Le modèle de formulaire détermine à la fois l'ordre dans lequel ces sections apparaissent et si elles sont obligatoires ou facultatives. Vous en apprendrez davantage sur les sections plus loin dans ce cours.
- *Le type de données que vous pouvez entrer:* Un modèle de formulaire définit les types de données (tels que nombres, dates ou texte) pouvant être entrés dans des champs de formulaire. Pour réduire les erreurs d'entrée de données, l'auteur du formulaire peut

définir le formulaire pour avertir l'utilisateur lorsqu'il tape un type de données non valide dans un champ ou qu'il omet de remplir un champ obligatoire.

- *La disponibilité des affichages:* Un modèle de formulaire peut comprendre plusieurs affichages. Un affichage est simplement une autre façon de consulter, filtrer ou afficher des données dans un formulaire. Par exemple, si vous remplissez un formulaire connecté à une base de données, le modèle de formulaire peut proposer deux affichages, l'un pour entrer des données et l'autre pour extraire des données de la base de données.

Utilisation connue, similaire à celle d'un document



- 1 Options de mise en forme courantes.
- 2 Outils d'édition auxquels vous êtes familiarisé, tels que le correcteur orthographique.
- 3 Volets Office pratiques.

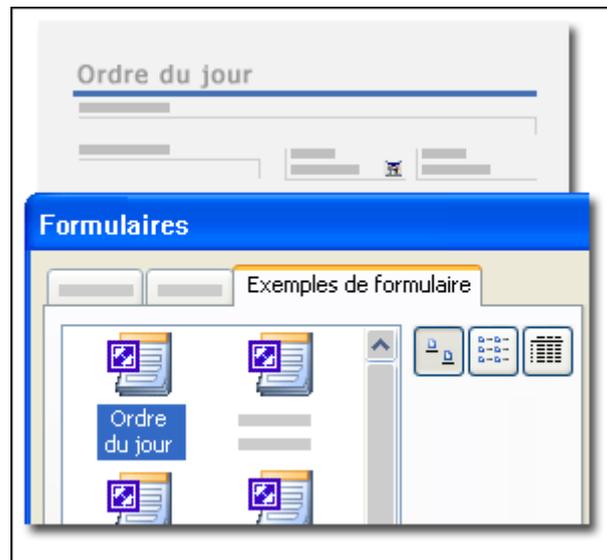
Dès que vous commencerez à utiliser InfoPath, vous retrouverez un grand nombre de barres d'outils, menus et fonctions similaires à d'autres programmes Microsoft Office. Ces points communs vous aident à éviter de perdre un temps précieux à découvrir et apprendre une nouvelle interface.

Par exemple, imaginons qu'un mot soit mal orthographié dans le formulaire. InfoPath indique votre erreur sous forme du traditionnel trait rouge ondulé, comme ici. De même, supposons que vous vouliez mettre en évidence du texte dans un champ de commentaires. Suivant la structure du formulaire, vous pouvez afficher le texte en gras ou en souligné, modifier la couleur du texte, voire même insérer des images et des tableaux. Les traditionnels formulaires visuels (et à fortiori les formulaires papier) ne permettent pas toujours ce type de personnalisation.

En outre, avec les traditionnels formulaires visuels, les utilisateurs sont souvent obligés de remplir et de soumettre le formulaire en une seule session. Dans InfoPath, en revanche, vous pouvez enregistrer le formulaire sur votre ordinateur et le compléter ultérieurement, tout

comme avec n'importe quel autre fichier. Prenons un exemple pour illustrer ce point. Si vous voyagez fréquemment dans le cadre de votre travail, vous devez utiliser le temps passé dans un aéroport ou dans un avion pour remplir vos notes de frais sur votre ordinateur portable. Plus tard, lorsque vous êtes de retour à votre bureau et connecté, vous pouvez ouvrir les formulaires et les soumettre à votre département de comptabilité en vue de leur traitement.

#### ***4.1 Formulaires prêts à l'emploi***



Les exemples de formulaires figurent dans la boîte de dialogue *Formulaires*. Pour ouvrir cette boîte de dialogue, cliquez sur *Plus de formulaires* dans le volet *Office Remplir un formulaire*.

InfoPath est fourni avec 25 exemples de formulaires commerciaux. Vous pouvez utiliser ces exemples de formulaires tels quels ou les personnaliser en fonction des besoins de votre groupe de travail ou de votre organisation.

La liste suivante représente quelques-uns des formulaires fournis avec InfoPath:

- Formulaire Demande de congé
- Formulaire Modification de commande
- Formulaire Note de frais
- Formulaire Facture
- Formulaire Ordre du jour
- Formulaire Analyse des performances
- Formulaire Rapport d'état

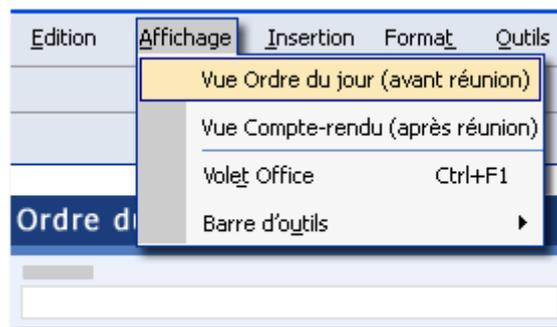
Les exemples de formulaires constituent un merveilleux moyen d'apprendre à l'aide d'exemples. Si vous voulez voir comment un formulaire donné a été créé, vous pouvez le décortiquer dans InfoPath et examiner les décisions prises pour obtenir un certain résultat. Vous pouvez également modifier les exemples de formulaires en fonction de vos besoins, par exemple en basant le formulaire de note de frais de votre entreprise sur celui fourni avec InfoPath.

Comme le dit l'expression populaire, nul n'est à l'abri d'une erreur. Ceci peut s'avérer particulièrement vrai lorsqu'il s'agit de remplir des formulaires. Ainsi, vous pouvez, par inadvertance, taper votre nom dans un champ qui accepte uniquement des nombres ou des dates.

Heureusement, InfoPath peut vérifier automatiquement votre formulaire en cas d'erreur et propose souvent des conseils pour les corriger. Ces conseils, ou *alertes d'erreur*, peuvent se présenter sous deux formes:

1. *Alertes en ligne*: Si une bordure rouge pointillée apparaît autour d'un champ dans votre formulaire, soyez attentif. Cette modification visuelle vous indique qu'il y a un problème avec les données que vous avez entrées. De même, si une ligne rouge apparaît sous un champ, cela signifie que le champ est obligatoire et que vous ne pouvez pas le laisser vierge. Pour en savoir plus sur une alerte en ligne spécifique, placez le pointeur de la souris sur le champ ou cliquez dessus avec le bouton droit.
2. *Alertes de boîte de dialogue*: Ces zones de message apparaissent dès que vous retirez le curseur du champ où vous avez effectué une entrée erronée. Si vous ne corrigez pas l'erreur immédiatement, le champ est marqué par une bordure rouge pointillée.

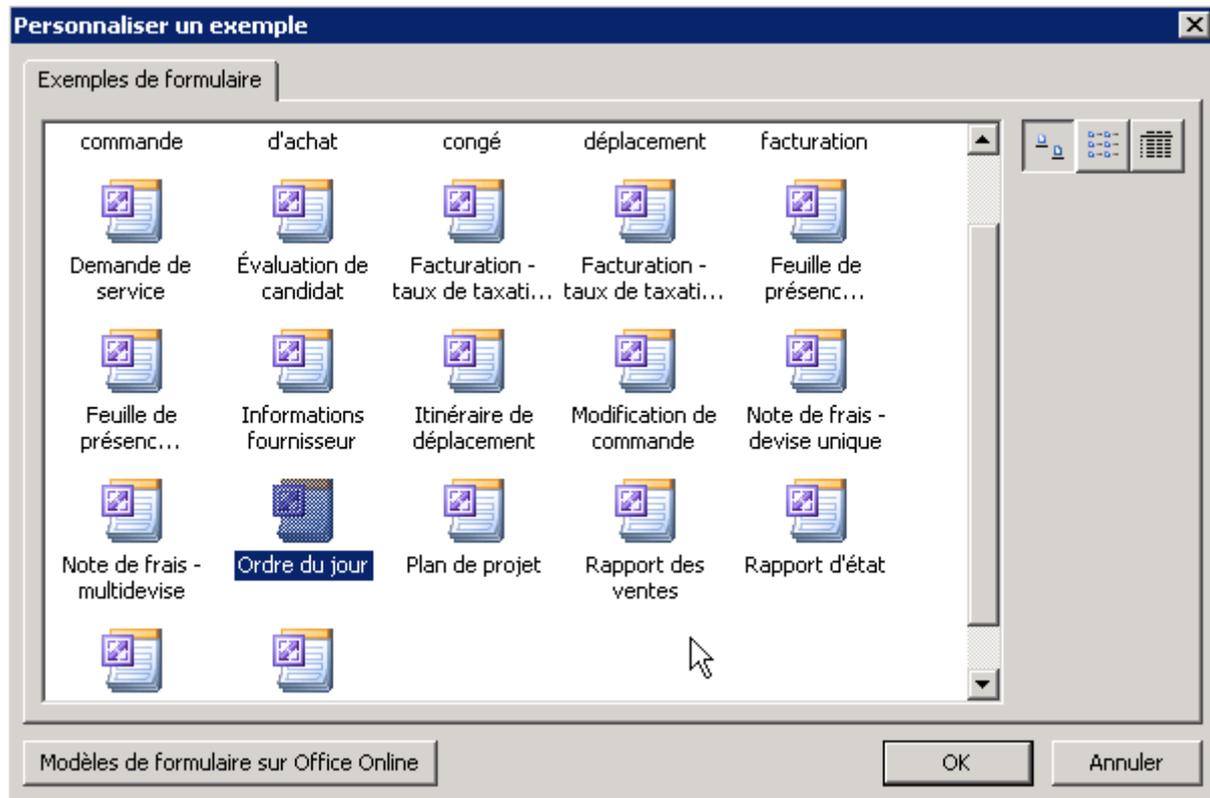
## 4.2 Les affichages



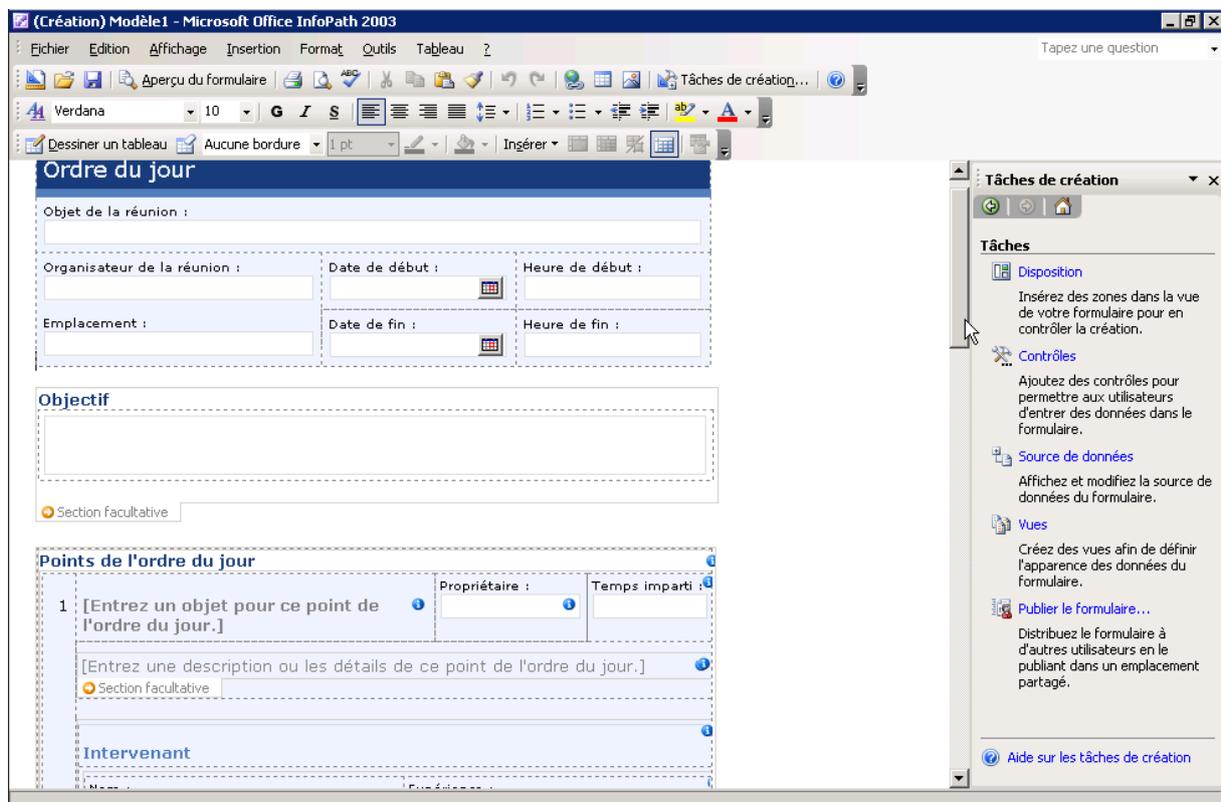
Si le formulaire que vous remplissez propose plusieurs affichages, vous pouvez basculer entre ces affichages en cliquant sur les commandes du menu Affichage.

Les formulaires InfoPath peuvent comprendre plusieurs affichages. Si vous vous en souvenez, dans la première leçon, nous avons vu que les affichages offrent une autre manière de présenter les données dans un formulaire.

Reprenons notre exemple de formulaire *Ordre du jour* fourni avec InfoPath:



Ce formulaire est conçu pour vous aider à organiser des réunions efficaces:

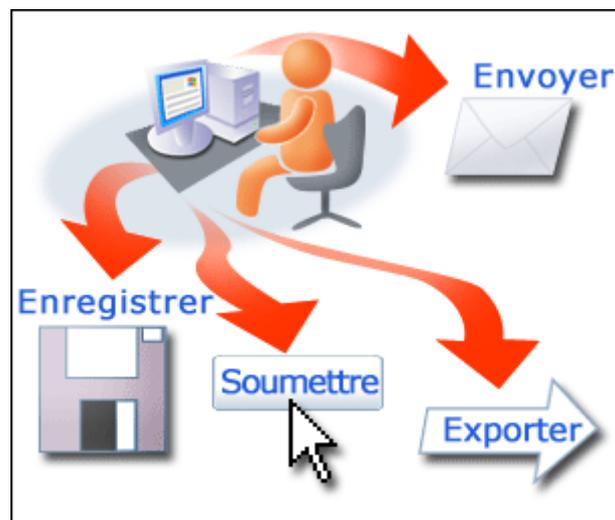


Ce formulaire (du moins dans sa version 2003...) contient deux affichages: un affichage *Ordre du jour* (avant réunion) et un affichage *Vue Compte-rendu* (après-réunion). Bien que ces deux affichages soient très similaires, ils ont chacun une fonction bien spécifique:

- Avant de rencontrer vos collègues, vous pouvez utiliser l'affichage *Ordre du jour* pour définir l'objectif de la réunion, pour spécifier les éléments de l'ordre du jour et leurs auteurs et pour énumérer les informations et la documentation fournies aux participants.
- Après la réunion, utilisez l'affichage *Compte rendu* pour synthétiser les décisions prises lors de la réunion et pour suivre les éléments d'action.

Tous les affichages partagent des informations communes, telles que les noms et les adresses de messagerie des participants. Vous pouvez donc éviter de retaper ou de copier et coller des données d'un affichage à un autre.

### 4.3 Workflow



Une fois le formulaire complété, vous pouvez enregistrer, soumettre, envoyer ou exporter les données. Une fois le formulaire rempli, quelques options sont à votre disposition. Vous pouvez notamment:

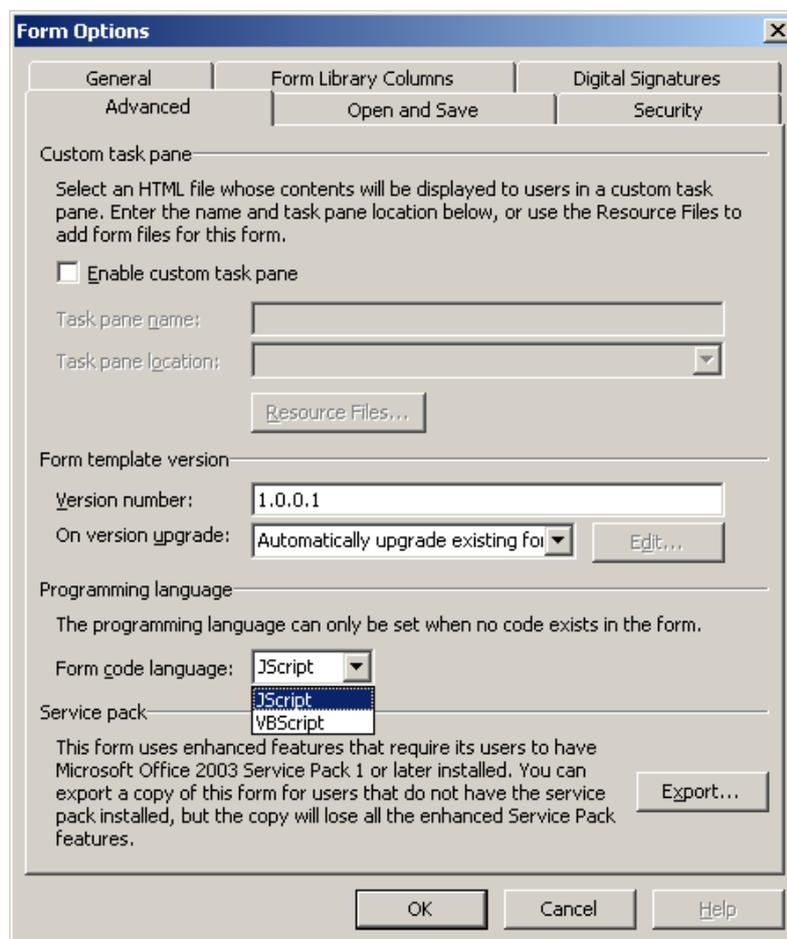
1. *Enregistrer votre formulaire*: Vous ne parvenez pas à remplir votre formulaire pour une raison quelconque ? Pas de problème. Enregistrez votre formulaire et terminez-le à un autre moment.
2. *Soumettre le formulaire terminé*: Si vous avez terminé de remplir un formulaire, vous pouvez l'envoyer à une base de données ou à un autre emplacement. Les formulaires InfoPath comprennent souvent des commandes spéciales permettant d'envoyer un formulaire (par exemple, un formulaire peut inclure une commande *Envoyer* dans le menu *Fichier*). Cette commande est spécifiée par la personne qui crée le formulaire.
3. *Exporter des données vers une feuille de calcul ou un navigateur Web*: Si vous utilisez Microsoft Excel, vous pouvez exporter des champs spécifiques de chaque formulaire (par exemple, noms d'employés et objectifs de performance à partir d'un formulaire d'analyse des performances) vers une feuille de calcul Excel pour analyse détaillée. Vous pouvez également exporter un formulaire pour que les autres personnes puissent l'afficher dans un navigateur Web. Cette fonctionnalité est pratique lorsque vous voulez partager votre formulaire avec quelqu'un qui ne possède pas InfoPath.

4. *Envoyer votre formulaire par courrier électronique*: Vous voulez que d'autres personnes vérifient ou examinent votre formulaire ? Dans ce cas, vous pouvez utiliser la commande *Envoyer au destinataire du message* pour envoyer le formulaire dans un message électronique.

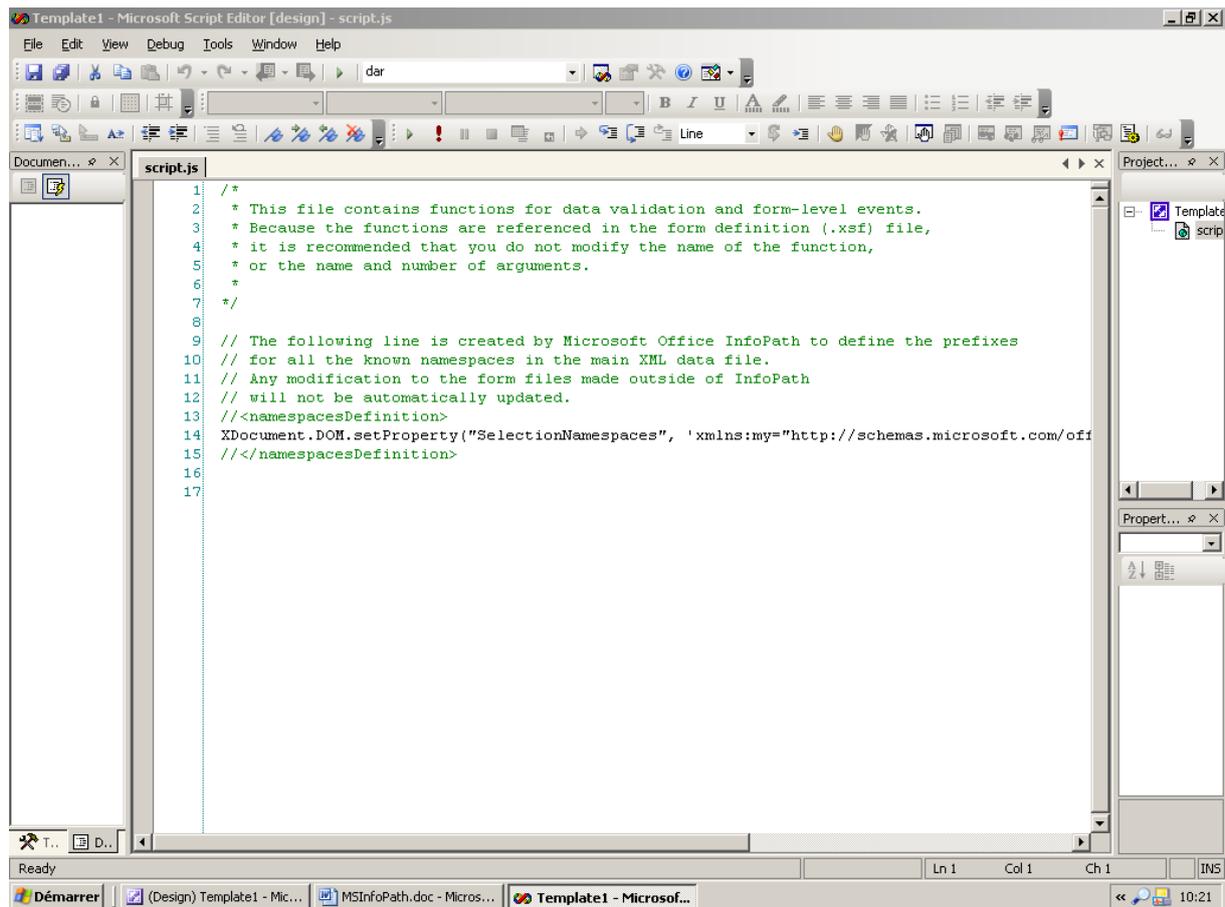
## 4.4 Développement

InfoPath (2003) supporte deux types de langages: le VBScript et le JScript (la version 2007 supporte le C# en plus et la 2010 ne supporte plus que du VB.Net ou du C#.Net). Un formulaire spécifique ne peut utiliser qu'un de ces deux langages et vous devez sélectionner le langage avant d'activer l'affichage du client de scripting.

Pour ce faire, allez dans *Tools/Form Options* et dans l'onglet *Advanced* pour InfoPath 2003 (sinon dans 2007 c'est dans *Tools/Form Options* et dans la section *Programming* et dans 2010 dans *File/Info/Form Options*):



Pour éditer du code dans *InfoPath* la méthode la plus rapide consiste à faire la combinaison de touches *Alt+Shift+F11*. Après il apparaît à l'écran:



**Remarque:** Il est important de se souvenir que le langage JScript est sensible à la casse.

Il peut être utile dans un formulaire d'avoir la date de création insérée automatiquement dans un champ. **InfoPath a déjà une fonction intégrée pour cela** mais faisons le en code pour le fun...

Créez ainsi un formulaire vierge et insérez y un contrôle (champ) de type texte que vous nommerez *date*. Après dans le menu *Tools* choisissez *Script/On Load Event*. Insérez-y le code suivant (**pour InfoPath 2003 seulement car depuis 2007 cela a complètement changé**):

```

function XDocument::OnLoad(eventObj)
{
var dateField = XDocument.DOM.selectSingleNode("//my:date");
dateField.text = todaysDate();
}
function todaysDate()
{
var d = new Date();
var s = (d.getMonth() + 1) + "/";
s += d.getDate() + "/";
s += d.getFullYear();
return(s);
}

```

Vincent ISOZ

Remarquez l'expression XPath //my:date pour identifier l'élément que nous souhaitons manipuler et la fonction Javascript connu pour formater un date au format voulu (très connue par les webdesigners).

L'autre considération à prendre en compte est simplement lorsque le formulaire sera sauvé et ouvert à nouveau. Effectivement, la plupart des personnes souhaitent que la date reflète celle de création du document. A ce moment là, il va falloir adapter un peu le code pour qu'il ne mette pas à jour le champ *date* lorsque une date s'y trouve déjà. On changera alors la procédure en ajoutant à l'endroit ad hoc:

```
if (dateField.text == "") dateField.text = todaysDate();
```

Remarque: par ailleurs ne pas ajouter cette ligne peut créer des surprises lorsque l'on ouvre un formulaire déjà existant.

Exercice: À l'aide de ce qui a été vu, créez une petite calculatrice qui fait l'addition de deux champs.

## 5. Dangers

Il est extrêmement rare que je parle des dangers d'un produit. Au même titre que pour MS Office SharePoint (voir mon livre électronique sur le sujet) où il est extrêmement dangereux de le déployer dans une entreprise sans un architecte et un spécialiste de la gouvernance SharePoint, InfoPath a **un piège très dangereux** qui est le suivant:

Si vous modifiez un modèle de formulaire de façon conséquente (perso je dirai même si vous le faites de manière minimale...), n'écrasez **jamais** l'ancien modèle car il se peut dès lors en fonction des modifications que vous avez faites que plus aucun des formulaires remplis avant le changement ne puissent s'ouvrir dans le nouveau modèle!!!! Il vaut donc mieux garder les anciens modèles pour retraiter les anciennes saisies et publier à côté la nouvelle version.

Ainsi, vous pourrez nommer et publier vos modèles respectivement sous le nom:

- NotesDeFraisV1\_Or8
- NotesDeFraisV2\_Or17
- etc.

Malheureusement, même si cette astuce est facile à mettre en place avec InfoPath utilisé sur des disques réseaux, il en va tout autrement lorsque vous utiliser InfoPath Services. Cette dernière configuration et l'erreur qui s'en suivra (et les centaines d'heures de correction) vous apprendra pourquoi avant de créer un formulaire on passe plusieurs jours à le modéliser (analyse fonctionnelle et structurelle).

## 6. Limites

Le produit (comme tout produit) a des limites qu'il est bon de savoir. Voici les limites courantes les plus ennuyeuses:

1. Le code JScript ou VBScript écrit avec InfoPath 2003 ou 2007 ne fonctionne plus dans InfoPath 2010 (qui ne gère que le code managé comme le .Net).
2. Si vous voulez faire un formulaire pour une base MS Access ou SQL Server, il vous faudra obligatoirement commencer par le modèle vierge de base prévu par Microsoft à cet effet (sinon il vous faudra faire du code).
3. Les *Template Part* (nouveau depuis InfoPath 2007) n'ont aucun intérêt puisque si elles sont utilisées dans des centaines de formulaires, elles ne se mettent pas à jour automatiquement et il faut faire quasiment les mêmes manipulations que si on les insérait pour les mettre à jour.
4. Si vous voulez créer un formulaire InfoPath d'une liste SharePoint existante, la manipulation est relativement longue et complexe avec InfoPath 2003 et 2007. Par contre, avec la version 2010, tout a changé et c'est devenu beaucoup plus simple.
5. La conversion de formulaires Word ou Excel en formulaires InfoPath est inutile car elle ne fonctionne que pour des formulaires triviaux qui sont très éloignés de la réalité des entreprises.
6. La publication de formulaires InfoPath dans SharePoint contenant des sections répétitives ne peut pas être utilisée sous forme de liste SharePoint (vous ne retrouverez pas les données de la section répétitive dans SharePoint).
7. Les rôles ne peuvent pas être gérés dans les formulaires InfoPath Services 2007 ou 2010.
8. Le contrôle *Contact selector* ne peut pas proposer le carnet d'adresse Active Directory de l'entreprise mais uniquement la liste des utilisateurs d'un site SharePoint (que ce soit sous la version 2003, 2007 ou 2010).
9. Le contrôle des styles de texte dans les formulaires est médiocre depuis la version 2003 jusqu'à la version 2010 (inutilisable si l'on ne va pas dans le code du fichier).
10. Il n'est pas possible de déclencher des événements particuliers avant l'ouverture d'un formulaire sur InfoPath Services ni après la fermeture sans faire de la programmation en .Net.
11. Vous ne pouvez pas envoyer de formulaires via Outlook 2007 contenant des champs de texte riche (RTF) ou possédant des connexions de données ou du code (ce dernier point étant très embêtant car beaucoup de designer InfoPath souhaiteraient que leurs employés puissent soumettre directement la réponse dans SharePoint).
12. La fusion de formulaires ne marche que pour des cas simplistes qui n'ont aucun intérêt dans la réalité des entreprises. Cette fonctionnalité est donc inutile dans son état actuel que ce soit dans InfoPath 2003, 2007 ou 2010.

13. Si vous éditez le formulaire d'une liste SharePoint 2010 avec InfoPath 2010, SharePoint utilisera le même formulaire pour la création et la modification. Il n'existe malheureusement pas de méthode simple autre que du formatage conditionnel pour avoir deux visuels différents.
14. On ne peut pas lier une colonne calculée SharePoint avec InfoPath 2003 ou 2007 (très utile pour concaténer deux colonnes SharePoint en une seule) et on ne peut pas concaténer les champs provenant de SharePoint dans InfoPath même sans passer par le flux XML de la liste correspondante (nous verrons dans les exemples comment utiliser ce flux).
15. Lorsque vous renommez ou créez une colonne dans SharePoint, celle-ci ne sera pas disponible automatiquement ou mise à jour automatiquement dans les formulaires InfoPath. Il faudra refaire l'assistant de connexion.
16. Compatibilité descendante de 2007 à 2003 catastrophique pour les formulaires complexes.
17. Vous ne pouvez pas lancer de workflows (flux) ou de requêtes sur les serveurs de base de données depuis InfoPath

## 7. InfoPath vs Webforms

Bien que InfoPath soit infiniment plus puissant que MS Word pour faire des formulaires, il n'en reste pas moins à ce jour un outil bureautique s'adressant principalement à des non informaticiens qui ne souhaitent ou ne savent pas faire du code et qui sont donc limités par rapport à leurs besoins réels.

Voici les points les plus demandés par des utilisateurs que l'on ne peut faire que via de la programmation informatique pure et pour lesquels il devient dès lors plus avantageux de laisser tomber InfoPath en ce qui concerne les formulaires Internet:

1. Limitations des contrôles non compatibles sur le navigateur (InfoPath Services)
2. Impossible de créer sur le navigateur des contrôles de son choix (limités aux contrôles proposés par Microsoft).
3. Cryptage des données sur le canal non possible
4. Gestion des cookies ou des variables de session non possible
5. L'accès à un formulaire sans avoir accès à la liste SharePoint n'est pas possible (problème avec l'accès anonyme donc).
6. Pas possible de faire des formulaires structurés en plusieurs étapes (du type assistant avec des boutons *Suivant*) qui se passent des paramètres.
7. Pas possible de centraliser le design et les fonctionnalités (mise à jour en cascade)
8. Ne marche bien que dans un environnement Microsoft avec des utilisateurs ayant un compte Active Directory et InfoPath Designer ou InfoPath Filler installé sur leur poste.
9. La mise à jour de la structure d'un formulaire avec SharePoint 2007 est cauchemardesque
10. Impossible de faire des éléments graphiques (diagrammes du type Excel) ou formules dynamiques complexes (genre formule de Black & Sholes) en fonction de la saisie en temps réel.
11. Il n'est pas possible de gérer les écritures concurrentes dans une base de données

## 8. Compatibilité InfoPath/InfoPath Services

### 8.1 Version 2007

Source:

<http://office.microsoft.com/fr-fr/infopath-help/compatibilite-des-navigateurs-web-dans-infopath-forms-services-HA010204085.aspx?CTT=5&origin=HA010233891>

● = Support total   ◐ = Support partiel   ○ = Pas de support

Contrôle InfoPath 2007	Navigateurs de poste de travail	Navigateurs d'appareil mobile
Zone de texte	●	●
Case à cocher	●	●
Bouton	●	●
Zone d'expression	●	●
Zone de liste	●	●
Zone de liste déroulante	●	●
Section extensible	●	●
Lien hypertexte	●	○
Case d'option	●	○
Section facultative	●	○
Tableau extensible	●	○
Contrôle de pièce jointe	●	○
Sélecteur de dates <sup>1</sup>	◐	◐
Zone de texte enrichi <sup>2</sup>	◐	○
Contrôle ActiveX	○	○
Contrôle de liste à puces	○	○
Groupe de choix	○	○
Section de choix	○	○
Zone de liste modifiable	○	○
Tableau extensible horizontal	○	○
Zone horizontale	○	○
Contrôle d'image manuscrite	○	○
Contrôle maître/détail	○	○
Zone de liste à sélection multiple	○	○
Contrôle de liste numérotée	○	○
Contrôle d'image	○	○
Contrôle de liste simple	○	○
Groupe de choix extensible	○	○
Section récursive extensible	○	○
Zone de défilement	○	○
Étiquette verticale	○	○

## Vincent ISOZ

<sup>1</sup>Bien que vous puissiez utiliser des sélecteurs de date dans un modèle de formulaire compatible avec le navigateur, l'icône du calendrier et le calendrier contextuel s'affichent uniquement dans les versions prises en charge d'Internet Explorer. Dans d'autres navigateurs, le sélecteur de dates s'affiche sous la forme d'une zone de texte, mais les options de mise en forme de dates sont toujours prises en charge.

<sup>2</sup>Vous ne pouvez pas activer des images incorporées ou spécifier des options de mise en forme de texte enrichi partielles pour les contrôles de zone de texte dans un modèle de formulaire compatible avec le navigateur. Toutes les options de mise en forme de texte enrichi ne sont disponibles que dans les versions prises en charge d'Internet Explorer. Les utilisateurs peuvent modifier le texte dans une zone de texte enrichi dans n'importe quel navigateur, mais les options de mise en forme ne sont pas disponibles, notamment la modification de la couleur du texte et le choix d'une mise en forme des caractères.

Le tableau suivant indique le niveau de prise en charge des fonctionnalités de logique métier InfoPath spécifiques par les navigateurs compatibles.

● = Support total   ◐ = Support partiel   ○ = Pas de support

Fonctionnalité InfoPath 2007	Navigateurs de poste de travail	Navigateurs d'appareil mobile
Formules <sup>1</sup>	●	●
Validation des données	●	●
Règles	●	●
Mise en forme conditionnelle <sup>2</sup>	●	◐
Rôles d'utilisateur	○	○

<sup>1</sup> Sauf fonctions `xdXDocument:get-Role`, `Position` et `Dernier`.

<sup>2</sup> Dans les navigateurs d'appareil mobile, certaines fonctionnalités de mise en forme conditionnelle sont prises en charge, notamment le masquage des contrôles en fonction de conditions qui se produisent dans le formulaire. Il est cependant impossible d'utiliser la mise en forme conditionnelle pour modifier l'aspect d'un contrôle en fonction de certaines conditions dans le formulaire. Par exemple, vous ne pouvez pas utiliser la mise en forme conditionnelle pour modifier la couleur du texte dans un contrôle.

Le tableau suivant indique le niveau de prise en charge des fonctionnalités de sécurité InfoPath spécifiques par les navigateurs compatibles.

● = Support total   ◐ = Support partiel   ○ = Pas de support

Fonctionnalité InfoPath 2007	Navigateurs de poste de travail	Navigateurs d'appareil mobile
Niveau de sécurité du domaine	●	●
Niveau de sécurité Autorisation totale	●	●
Niveau de sécurité restreint	○	○
Signatures numériques <sup>1</sup>	◐	○
Protection de la création de formulaires	○	○
Possibilité de désactiver les options <b>Enregistrer</b> , <b>Imprimer</b> et d'autres commandes	●	●
Gestion des droits relatifs à l'information (IRM)	○	○

<sup>1</sup> Les signatures numériques ne sont pas prises en charge dans les modèles de formulaire activés pour le navigateur qui sont rendus dans des navigateurs d'appareil mobile. En outre, dans les navigateurs de poste de travail, les utilisateurs ne peuvent pas signer numériquement un formulaire entier. Vous pouvez toutefois concevoir le modèle de formulaire afin que les utilisateurs puissent signer numériquement une ou plusieurs sections du formulaire. Pour cela, vous devez activer les signatures numériques dans le contrôle des sections concernées. De plus, bien que les utilisateurs puissent afficher les signatures numériques d'un formulaire dans tous les navigateurs pris en charge, ils peuvent signer numériquement des sections de formulaire uniquement dans les versions prises en charge de Microsoft Internet Explorer.

Le tableau suivant indique le niveau de prise en charge des connexions de données InfoPath dans les modèles de formulaire activés pour le navigateur.

● = Support total ◐ = Support partiel ○ = Pas de support

Fonctionnalité InfoPath 2007	Navigateurs de poste de travail	Navigateurs d'appareil mobile
Interroger les données d'un fichier XML ou d'une URL qui renvoie du XML	●	●
Interroger les données d'une liste ou bibliothèque Microsoft Windows SharePoint Services	●	●
Interroger les données d'une base de données Microsoft SQL Server	●	●
Interroger les données d'une base de données Microsoft Office Access	○	○
Interroger les données d'un service Web	●	●
Interroger des données à l'aide des fichiers de connexion de données d'une bibliothèque de connexions de données	●	●

Le tableau suivant indique le niveau de prise en charge par les navigateurs compatibles des fonctionnalités InfoPath liées à l'envoi des formulaires.

● = Support total ◐ = Support partiel ○ = Pas de support

Fonctionnalité InfoPath 2007	Navigateurs de poste de travail	Navigateurs d'appareil mobile
Envoyer des données en tant que message électronique	●	●
Envoyer des données à une bibliothèque SharePoint <sup>1</sup>	●	●
Envoyer des données à une base de données Microsoft SQL Server	○	○
Envoyer des données à une base de données Microsoft Access	○	○
Envoyer des données à un service Web <sup>2</sup>	●	●
Envoyer des données à une application Web qui accepte les envois HTTP POST	●	●
Envoyer des données à l'aide des fichiers de connexion de données d'une bibliothèque de connexions de données	●	●

<sup>1</sup> Il est possible d'envoyer des données à une bibliothèque SharePoint, mais pas à une liste.

<sup>2</sup> Vous ne pouvez pas configurer la connexion de données d'envoi dans un modèle de formulaire compatible avec le navigateur afin de permettre aux utilisateurs d'envoyer des informations de modification avec leurs données de formulaire à un service Web qui accepte un DataSet Microsoft ADO.NET. Les informations de modification se composent des modifications qu'un utilisateur apporte aux données stockées dans une base de données. La base de données utilise un service Web pour connecter des utilisateurs à la base de données. ADO.NET peut utiliser les informations de modification pour déterminer comment mettre à jour les données dans la base de données.

Le tableau suivant indique le niveau de prise en charge par les navigateurs compatibles des fonctionnalités InfoPath liées au remplissage des formulaires.

● = Support total ◐ = Support partiel ○ = Pas de support

Fonctionnalité InfoPath 2007	Navigateurs de poste de travail	Navigateurs d'appareil mobile
Envoyer	●	●
Enregistrer <sup>1</sup>	●	○
Imprimer	●	○
Fusionner des données de plusieurs formulaires	○	○
Exporter des données vers Microsoft Office Excel	○	○
Exporter des données vers le Web	○	○
Prise en charge des entrées manuscrites	○	○
Prise en charge hors connexion	○	○
Rechercher et remplacer du texte	●	●
Saisie semi-automatique <sup>2</sup>	●	○
Vérification orthographique	○	○
Enregistrement automatique, récupération automatique ou fonctionnalités de détection des arrêts intempestifs	○	○
Volets Office personnalisés	○	○
Menus des compléments	○	○

<sup>1</sup> Dans un formulaire activé pour le navigateur, les utilisateurs peuvent cliquer sur les boutons **Enregistrer** ou **Enregistrer sous** pour enregistrer leurs formulaires dans une bibliothèque SharePoint. En revanche, ils ne peuvent pas enregistrer leurs formulaires sur leurs ordinateurs. En outre, les utilisateurs qui visualisent des formulaires dans un navigateur d'appareil mobile ne voient pas les boutons **Enregistrer** ou **Enregistrer sous**.

<sup>2</sup> Dans les navigateurs autres qu'Internet Explorer, la fonctionnalité Saisie semi-automatique n'est pas disponible, même si la case à cocher **Activer la saisie semi-automatique** est activée pour le contrôle correspondant.

Le tableau suivant indique le niveau de prise en charge par les navigateurs compatibles des fonctionnalités InfoPath liées aux vues.

● = Support total ◐ = Support partiel ○ = Pas de support

Fonctionnalité InfoPath 2007	Navigateurs de poste de travail	Navigateurs d'appareil mobile
Vues multiples	●	●
Vues en lecture seule	●	◐
Vues d'impression	●	○
Vues d'impression Word <sup>1</sup>	○	○
Imprimer des vues multiples simultanément <sup>2</sup>	○	○
Imprimer les sauts de page, les en-têtes et les pieds de page	○	○

<sup>1</sup> Les vues d'impression Microsoft Office Word ne sont pas prises en charge dans les modèles de formulaire activés pour le navigateur. Vous pouvez publier un modèle de formulaire sur InfoPath Forms Services sans supprimer la vue d'impression Word. Cependant, lorsque les utilisateurs impriment un formulaire, le navigateur Web ignore les paramètres de la vue d'impression Word.

<sup>2</sup> La fonctionnalité **Imprimer des vues multiples** n'est pas prise en charge dans les modèles de formulaire activés pour le navigateur. Vous pouvez publier un modèle de formulaire sur InfoPath Forms Services sans modifier les options de la boîte de dialogue **Imprimer des vues**

**multiples.** Cependant, lorsque les utilisateurs impriment un formulaire, le navigateur Web utilise une vue en lecture seule de la vue actuelle à des fins d'impression au lieu d'imprimer des vues multiples.

Les développeurs peuvent utiliser le modèle objet de code managé InfoPath pour créer une logique métier personnalisée dans des modèles de formulaire compatibles avec le navigateur. Lorsqu'elle est déployée sur un serveur exécutant InfoPath Forms Services, cette logique métier s'exécute à la fois dans un navigateur et dans InfoPath. Le tableau suivant indique le niveau de prise en charge par les navigateurs compatibles des fonctionnalités de programmation InfoPath.

● = Support total   ◐ = Support partiel   ○ = Pas de support

Fonctionnalité InfoPath 2007	Navigateurs de poste de travail	Navigateurs d'appareil mobile
Code managé <sup>1</sup>	●	●
Scripts <sup>2</sup>	○	○
Compléments COM	○	○

<sup>1</sup> Dans un modèle de formulaire compatible avec le navigateur, le code managé doit se baser sur le modèle objet du code managé InfoPath fourni par l'espace de noms **Microsoft.Office.InfoPath**. Un sous-ensemble des membres de l'espace de noms **Microsoft.Office.InfoPath** fournit des fonctionnalités, telles que la gestion des droits relatifs à l'information (IRM), qui ne sont pas prises en charge sur le serveur. Pour plus d'informations sur les membres du modèle objet pris en charge et non pris en charge, reportez-vous aux liens de la section **Voir aussi** au début de cet article.

<sup>2</sup> Tout code écrit en VBScript ou JScript ainsi que le modèle objet compatible avec InfoPath 2003 fourni par les membres de l'espace de noms **Microsoft.Office.Interop.InfoPath.SemiTrust** ne sont pas pris en charge sur le serveur.

## 8.2 Version 2010

Source (pas envie de reformater tout la page web de Microsoft dans Word...):

<http://office.microsoft.com/fr-fr/infopath-help/fonctionnalites-dinfopath-2010-non-disponibles-dans-les-formulaires-de-navigateur-web-HA101732796.aspx?CTT=1>

## 9. Applications pratiques (sans dev.)

Nous allons étudier MS InfoPath par des exemples allant du plus simple au plus compliqué. Nous étudierons les possibilités du logiciel dans le l'ordre sous-mentionné:

1. Création d'un formulaire de notes de frais
2. Création d'adresse de fournisseurs avec XSD
3. Connexion du formulaire effectué au point (2) avec une base de données MS Access
4. Création d'un formulaire de Notes de frais avec SharePoint Portal Server
5. Création d'un formulaire de news avec envoi à un webservice
6. Création d'un formulaire connecté à une base de données avec interactivité dans le panneau office.

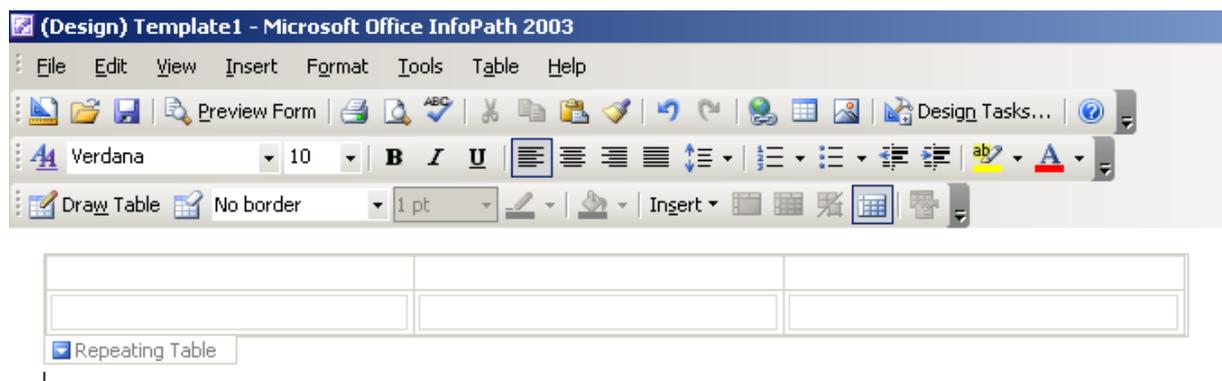
Commençons par l'exemple simple.

### 9.1 XML, XSL, XSD, XSF

Comme nous en avons parlé pendant les 4 jours du cours d'introduction aux technologies sur le XML, InfoPath 2003 est le produit MS Office qui fait le plus usage de ce standard. Au fait, il ne marche quasiment que sur cette base.

Nous allons reprendre un des exemples de base du cours XML pour voir cela.

Ouvrez InfoPath et allez et créez un nouveau formulaire en mode *Design*. Dans le panneau Office allez dans la boîte à outils contrôles et insérez en un du type *Repeating Table*. Quand vous insérez le contrôle spécifiez que vous souhaitez trois colonnes et cliquez sur *OK*:



Maintenant écrivez des légendes comme ci-dessous:

Name	Firstname	e-mail adress

Repeating Table

Vincent ISOZ

Faites ensuite un double clic sur chacun des champs et saisissez dans le champ *Field Name* les noms respectifs: *name*, *firstname*, *e-mail*

Une fois ceci fait, enregistrez votre formulaire dans un dossier de votre choix sous le nom *Demonstration.xsn*.



Dans le menu *File* de InfoPath choisissez maintenant l'option *Extract from files* et validez par *OK*:



Vous aurez alors dans le dossier choisi:



Description:

1. Le fichier *manifest.xsf* contient la structure et les propriétés du fichier ainsi que des vues du *xsn*. Ce genre de fichier n'a pas été étudié lors de notre cours XML car assez particulier à InfoPath. On peut faire des choses relativement intéressant avec ce fichier pour personnaliser les menus contextuels d'InfoPath mais ceci dépasse le cadre de ce cours.

```

manifest.xsf - Bloc-notes
Fichier Edition Format Affichage ?
<?xml version="1.0" encoding="UTF-8"?>
<!--
This file is automatically created and modified by Microsoft Office InfoPath.
Changes made to the file outside of InfoPath might be lost if the form template is modified in InfoPath.
-->
<xsf:documentClass trustSetting="automatic" trustLevel="restricted" solutionVersion="1.0.0.2" productVersion="11.0.6565" sol
4">
  <xsf:package>
    <xsf:files>
      <xsf:file name="myschema.xsd">
        <xsf:fileProperties>
          <xsf:property name="namespace" type="string" value="http://schemas.microsoft.com/offi
          <xsf:property name="editability" type="string" value="full"></xsf:property>
          <xsf:property name="rootElement" type="string" value="myFields"></xsf:property>
          <xsf:property name="useOnDemandAlgorithm" type="string" value="yes"></xsf:property>
        </xsf:fileProperties>
      </xsf:file>
      <xsf:file name="template.xml"></xsf:file>
      <xsf:file name="sampledata.xml">
        <xsf:fileProperties>
          <xsf:property name="fileType" type="string" value="sampledata"></xsf:property>
        </xsf:fileProperties>
      </xsf:file>
      <xsf:file name="view1.xsl">
        <xsf:fileProperties>
          <xsf:property name="lang" type="string" value="1036"></xsf:property>
          <xsf:property name="componentId" type="string" value="4"></xsf:property>
          <xsf:property name="xmlToEditName" type="string" value="1"></xsf:property>
        </xsf:fileProperties>
      </xsf:file>
    </xsf:files>
  </xsf:package>
  <xsf:importParameters enabled="yes"></xsf:importParameters>
  <xsf:documentVersionUpgrade>
    <xsf:useTransform transform="" minVersionToUpgrade="0.0.0.0"></xsf:useTransform>
  </xsf:documentVersionUpgrade>
  <xsf:views default="view 1">
    <xsf:view name="view 1" caption="view 1">
      <xsf:editing>
        <xsf:xmlToEdit name="group2_1" item="/my/myFields/my:group1/my:group2" container="/my/myField
        <xsf:editWith caption="group2" xd:autogeneration="template" component="xcollection">
          <xsf:fragmentToInsert>
            <xsf:chooseFragment innerFragment="my:group1/my:group2">
              <my:group1>
                <my:group2>
                  <my:name></my:name>
                  <my:firstname></my:firstname>
                  <my:email></my:email>
                </my:group2>
              </my:group1>
            </xsf:chooseFragment>
          </xsf:fragmentToInsert>
        </xsf:editWith>
      </xsf:editing>
    </xsf:view>
  </xsf:views>
</xsf:documentClass>

```

2. Le fichier *myschema.xsd* à une structure qui nous est parfaitement connue suite au cours sur le XML. Le participant ayant bien suivi le cours remarquera qu'il y a un namespace bien particulier pour InfoPath utilisant *my*

```

myschema.xsd - Bloc-notes
Fichier Edition Format Affichage ?
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xsd:schema targetNamespace="http://schemas.microsoft.com/office/infopath/2003/myXSD/2006-04-18T06:22:14" xmlns:my="http://scl
<xsd:element name="myFields">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="my:group1" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:attribute processContents="lax" namespace="http://www.w3.org/XML/1998/namespace"/>
</xsd:element>
<xsd:element name="group1">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="my:group2" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="group2">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="my:name" minOccurs="0"/>
      <xsd:element ref="my:firstname" minOccurs="0"/>
      <xsd:element ref="my:email" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="name" type="xsd:string"/>
<xsd:element name="firstname" type="xsd:string"/>
<xsd:element name="email" type="xsd:string"/>
</xsd:schema>

```

**Astuce!** Il est très intéressant de remarquer que nous avons l'attribut *minOccurs* et *maxOccurs* pour le noeud de *sequence* suivant:

```

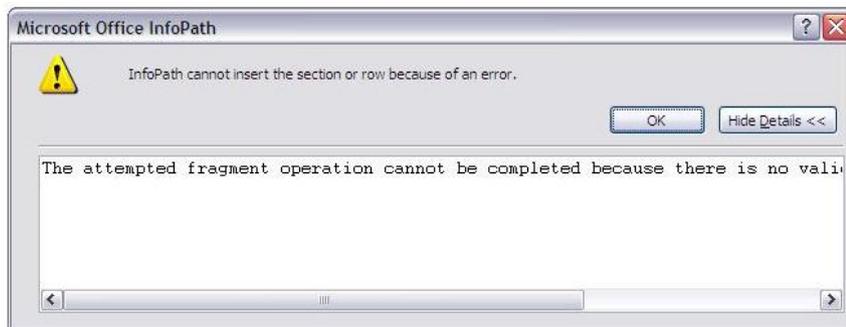
<xsd:sequence>
  <xsd:element ref="my:group2" minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>

```

Pourquoi est-ce intéressant? Simplement parce que dans InfoPath il n'existe pas à ce jour de moyen simple d'imposer un nombre de lignes précréées et de limiter la création de lignes dans une *sequence* (une section répétitive). Il suffit cependant de changer le code précédent de la manière suivante:

```
<xsd:element ref="my:group2" minOccurs="3" maxOccurs="10"/>
```

Ainsi, quand les utilisateurs généreront un nouveau formulaire à partir du modèle, ils auront trois lignes par défaut vides déjà préparées et s'ils essayent de créer plus de 10 lignes, ils auront le message suivant:



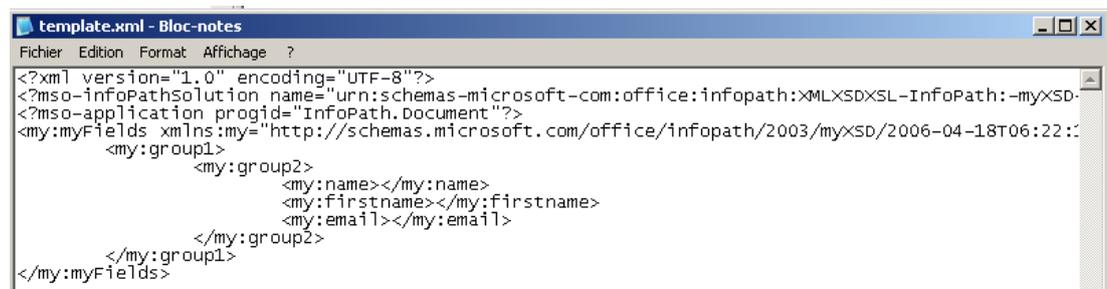
Remarque: Ce fichier ne peut cependant pas être ouvert avec XMLSpy car contenant des éléments propriétaires Microsoft en début de fichier. Nous avons cependant vu lors du cours XML comment créer un schéma XSD propre et l'utiliser dans une vingtaine de technologies différentes.

3. Le fichier *sampledata.xml* contient un exemple vide du fichier XML qui résultera de la saisie du formulaire par l'utilisateur ainsi que la déclaration du Namespace *myFields*

Remarque: Personnellement je n'utilise jamais ce fichier (et n'en voit pas l'intérêt)



4. Le fichier *template.xml* contient les données de bases telles qu'elles apparaîtront lorsque l'utilisateur enregistrera les données du formulaire dans un fichier XML



5. Le fichier *view1.xsl* contient simplement le code effectuant la mise en page de la vue par défaut *View 1*. Nous retrouvons un élément que nous avons bien exercé lors de notre cours sur le XML et nous voyons qu'InfoPath utilise le HTML pour la

mise en forme des vues, ce qui est assez remarquable:

```

view1.xsl - Bloc-notes
Fichier Edition Format Affichage ?
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:my="http://schemas.microsoft.com/office/infopath/2003/myXSD/2006-04-18T06:22:14" xmlns:xsl="http://schemas.xmlsoap.org/xsl/">
  <xsl:output method="html" indent="no"/>
  <xsl:template match="my:myFields">
    <html>
      <head>
        <style tableEditor="tableStyleRulesID">TABLE.xdLayout TD {
          BORDER-RIGHT: medium none; BORDER-TOP: medium none; BORDER-LEFT: medium none; BORDER-BOTTOM: medium none
        }
        TABLE {
          BEHAVIOR: url (#default#urn::tables/NDTable)
        }
        TABLE.msoucTable TD {
          BORDER-RIGHT: 1pt solid; BORDER-TOP: 1pt solid; BORDER-LEFT: 1pt solid; BORDER-BOTTOM: 1pt solid
        }
      </style>
      <meta http-equiv="Content-Type" content="text/html"></meta>
      <style controlStyle="controlStyle">@media screen {
        .ghostedTextNoBUI{BEHAVIOR: url(#default#urn::controls/Binder) url(#default#ghostedText);}
        .xdBelPageBreak{display: none;}
        BODY{margin-right: 21px;}
        .xdTextBoxRTL{display: inline-block; white-space: nowrap; text-orientation: vertical; background-color: window; overflow-x: hidden; word-wrap: break-word; text-overflow: ellipsis; text-align: right;}
        FONT-SIZE: 10pt; FONT-FAMILY: verdana
      }
      TABLE {
        FONT-SIZE: 10pt; FONT-FAMILY: verdana
      }
      SELECT {
        FONT-SIZE: 10pt; FONT-FAMILY: verdana
      }
      .optionalPlaceholder {
        PADDING-LEFT: 20px; FONT-WEIGHT: normal; FONT-SIZE: xx-small; BEHAVIOR: url(#default#xoptional); COLOR: #333333; FONT-FAMILY: verdana
      }
      .langFont {
        FONT-FAMILY: verdana
      }
      .defaultInDocUI {
        FONT-SIZE: xx-small; FONT-FAMILY: verdana
      }
      .optionalPlaceholder {
        PADDING-RIGHT: 20px
      }
    </style>
    </head>
    <body>
      <div>
        <table class="xdRepeatingTable msoucTable" title="" style="TABLE-LAYOUT: fixed; WIDTH: 100%; border-collapse: collapse;">
          <colgroup>
            <col style="WIDTH: 203px;"></col>
            <col style="WIDTH: 204px;"></col>
            <col style="WIDTH: 222px;"></col>
          </colgroup>
        </table>
      </div>
    </body>
  </html>
  
```

Ce fichier est très utile pour ajouter des *xsl:sort* à la main car par défaut InfoPath ne sait pas trier le contenu de liste déroulantes!!!

Remplissons maintenant notre formulaire avec au moins deux enregistrements:

Name	Firstname	e-mail adress
Isoz	Vincent	isoz@sciences.ch
Demarlière	Laurianne	laurianne.demarliere@infolearn.ch

Insert item

Enregistrez ensuite les données sous le fichier XML proposé dans le dossier courant. Vous obtiendrez en l'ouvrant dans un éditeur de texte:

```

Demonstration.xml - Bloc-notes
Fichier Edition Format Affichage ?
<?xml version="1.0" encoding="UTF-8"?><?mso:infopath:solution solutionVersion="1.0.0.1" productVersion="11.0.6565" PIVersion="11.0.6565" />
  <my:group1>
    <my:group2>
      <my:name>Isoz</my:name>
      <my:firstname>vincent</my:firstname>
      <my:email>isoz@sciences.ch</my:email>
    </my:group2>
    <my:group2>
      <my:name>Demarlière</my:name>
      <my:firstname>Laurianne</my:firstname>
      <my:email>laurianne.demarliere@infolearn.ch</my:email>
    </my:group2>
  </my:group1>
</my:myFields>
  
```

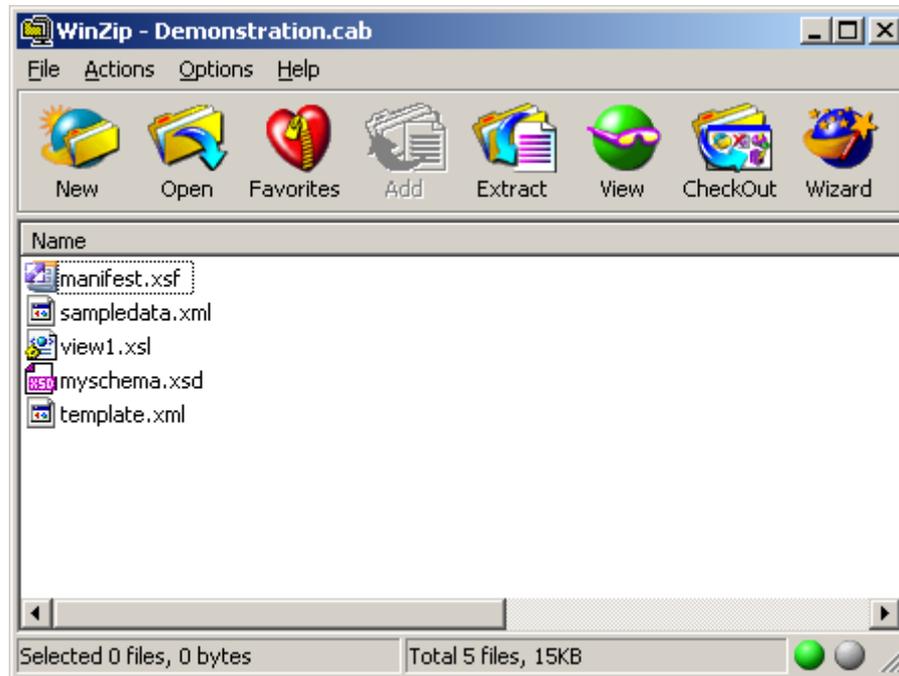
Nous avons donc ici un fichier XML habituel et très facilement exploitable.

Par ailleurs, InfoPath est le premier logiciel à se comporter comme le fera l'ensemble de la suite MS Office à l'avenir (depuis 2003, année à laquelle j'ai écrit le présent texte, Office 2007

Vincent ISOZ

a commencé à intégrer le XML dans Word, Excel et PowerPoint) et comme le font déjà de nombreuses applications concurrentes.

Si vous faites une copie de votre formulaire .xsn et que vous le renommez en .cab et que vous faites un double clic dessus, vous obtiendrez:



Donc nous voyons ici que le fichier xsn n'est qu'un conteneur.

Remarque: si l'on utilise des webservices ainsi que du JScript, nous aurons alors quelque chose de plus conséquent:

Name	Modified	Size	Ratio	Packed	Path
template.xml	4/20/2003 10:22 PM	528	?	?	
manifest.xsf	4/20/2003 10:22 PM	7,748	?	?	
view1.xsl	4/20/2003 10:16 PM	17,236	?	?	
script.js	4/20/2003 10:16 PM	2,278	?	?	
sampledata.xml	4/20/2003 10:16 PM	231	?	?	
myschema.xsd	4/20/2003 10:16 PM	1,579	?	?	
GetRequestTypes2.xsd	4/20/2003 10:16 PM	1,017	?	?	
GetRequestTypes1.xsd	4/20/2003 10:16 PM	1,012	?	?	
GetRequestTypes.xsd	4/20/2003 10:16 PM	1,298	?	?	
GetRequestTypes.xml	4/20/2003 10:16 PM	545	?	?	
GetEmpList2.xsd	4/20/2003 10:16 PM	904	?	?	
GetEmpList1.xsd	4/20/2003 10:16 PM	908	?	?	
GetEmpList.xsd	4/20/2003 10:16 PM	1,195	?	?	
GetEmpList.xml	4/20/2003 10:16 PM	427	?	?	

**Attention!!** Lorsque vous mettez des images en tant qu'arrière plan de vos formulaires et que vous les supprimez ensuite, la taille de votre fichier \*.xsn ne changera plus... Pour contourner ce problème voici la procédure à suivre:

1. Extraire tous les fichiers comme montré précédemment dans un dossier de votre choix

2. Supprimer les images qui auront aussi été décompressées dans le dossier crée dans l'étape précédente
3. Dans le fichier \*.xsf supprimer la balise faisant référence au nom de l'image
4. Ensuite recompiler le fichier \*.xsn en faisant un clic droit sur le \*.xsf et choisir *Create*. Cela va ouvrir le formulaire en mode *Design*. Il ne reste plus qu'à l'enregistrer...

## 9.2 Formulaire simple (notes de frais)

Nous allons nous passer ici des modèles de bases d'InfoPath pour créer un formulaire très spécifique à nos besoins.

Le cahier des charges sera de créer un formulaire avec:

1. La possibilité pour le collaborateur de choisir son nom et prénom dans une liste déroulante à partir d'un fichier XML tu type suivant:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Employees>
- <Employee typ="Manager">
  <name>Nicolas Mayer</name>
</Employee>
- <Employee typ="Manager">
  <name>Forster Laurence</name>
</Employee>
- <Employee typ="Consultant">
  <name>Isoz Vincent</name>
</Employee>
- <Employee typ="Consultant">
  <name>Weber Olivier</name>
</Employee>
- <Employee typ="Consultant">
  <name>Fournier Fabrice</name>
</Employee>
- <Employee typ="Manager">
  <name>Ravinet Valérie</name>
</Employee>
</Employees>
```

2. La possibilité pour le collaborateur de choisir le nom de son responsable dans une liste déroulante à partir du même fichier XML
3. La possibilité pour le collaborateur de spécifier le mois et l'année pour laquelle le formulaire est valide (dans deux listes déroulantes respectives)
4. Enfin, une section répétitive où il peut saisir une date de début et de fin, l'intervention, le responsable de l'intervention et les notes de frais distinctes (kilométrage, repas, hébergement, divers)
5. Le formulaire doit calculer la somme des notes de frais pour le mois en cours
6. Le formulaire doit contenir trois champs d'images pour insérer les signatures
7. Si l'adresse e-mail est fournie, l'employé doit pouvoir envoyer ses notes de frais par e-mail via une pièce jointe XML

Vincent ISOZ

8. Le tout avec une jolie mise en page si possible... et deux vues distinctes: une pour les employés, et une pour les responsables.

Voyons comment procéder (c'est tellement simple que j'ai pas souhaité l'écrire...) nous allons faire cela ensemble pendant la première journée de la formation.

Voici un exemple de ce dont à quoi cela pourrait ressembler à la fin:

La note de frais se remplit uniquement du 1er au dernier jour du mois.  
Les justificatifs doivent être joints à la note de frais.

### FORMULAIRE DE NOTES DE FRAIS

Nom Prénom:  Nom Prénom du responsable:

Mois:  Année:

Date d'envoi: 18.04.2006

### PARTICIPATION AUX NOTES DE FRAIS

DATES	LIEU	RESPONSABLE, NOM DU CLIENT ET PRESTATION	FRAIS
Début: <input type="text"/> *	Départ: <input type="text"/> *	Responsable: <input type="text"/>	Kilomètres: <input type="text"/>
Fin: <input type="text"/> *	Arrivée: <input type="text"/> *	Nom Client: <input type="text"/> *	Prix (0.7.-/km) <input type="text"/>
		Prestation: <input type="text"/>	Repas: <input type="text"/>
			Hébergement: <input type="text"/>
			Divers: <input type="text"/>
			<b>Total:</b> <input type="text"/>
			CHF 0.00

Insert item

### SOUSSION DU FORMULAIRE

Mail supérieur hiérarchique:  Mail responsable notes de frais:

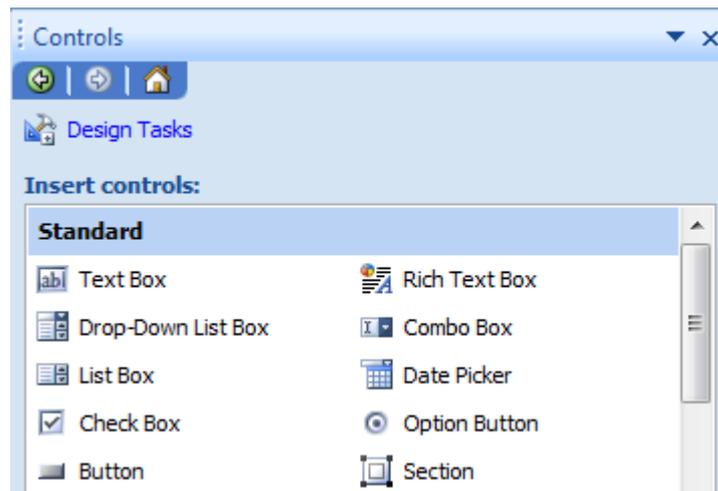
Signature collaborateur:  Signature supérieur hiérarchique:  Visa Direction:

Click here to insert a picture  Click here to insert a picture  Click here to insert a picture

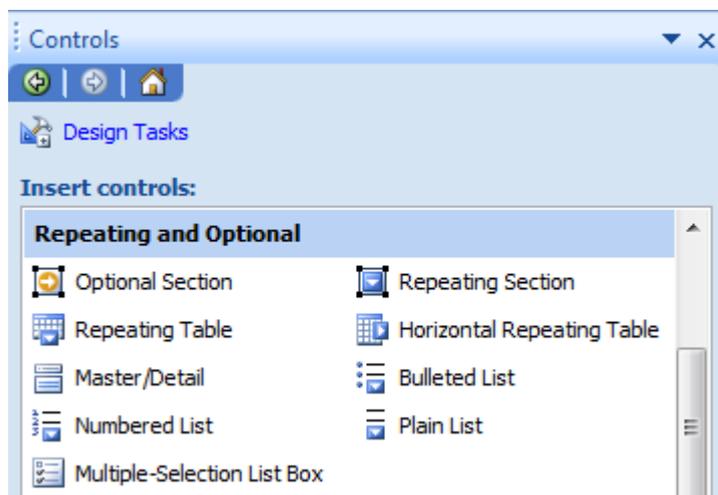
## 9.2.1 Types de contrôles

Dans InfoPath 2003/2007 ou 2010 la majorité de mes clients trouvent la majorité des contrôles évidents et simples à utiliser excepté quelques uns. Le but de ce petite chapitre est de parler de ceux qui posent problème.

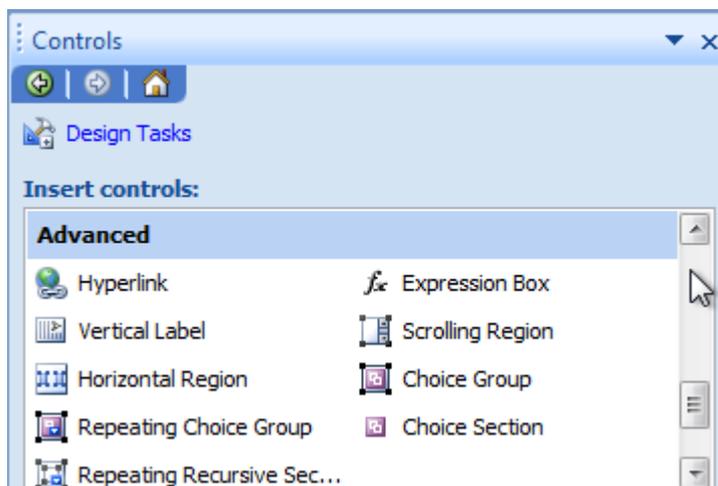
Nous avons donc la première catégorie de contrôles *Standard*:



Le seul contrôle sur lequel de nombreux clients s'interrogent est celui nommé *Section*. Dans la catégorie *Repeating and optional*:

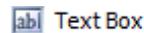


les clients s'interrogent sur l'utilité de tous les contrôles. Il en est de même pour la catégorie *Advanced*:



### 9.2.1.1 Zone de texte

Le contrôle de type zone de texte n'a rien de difficile



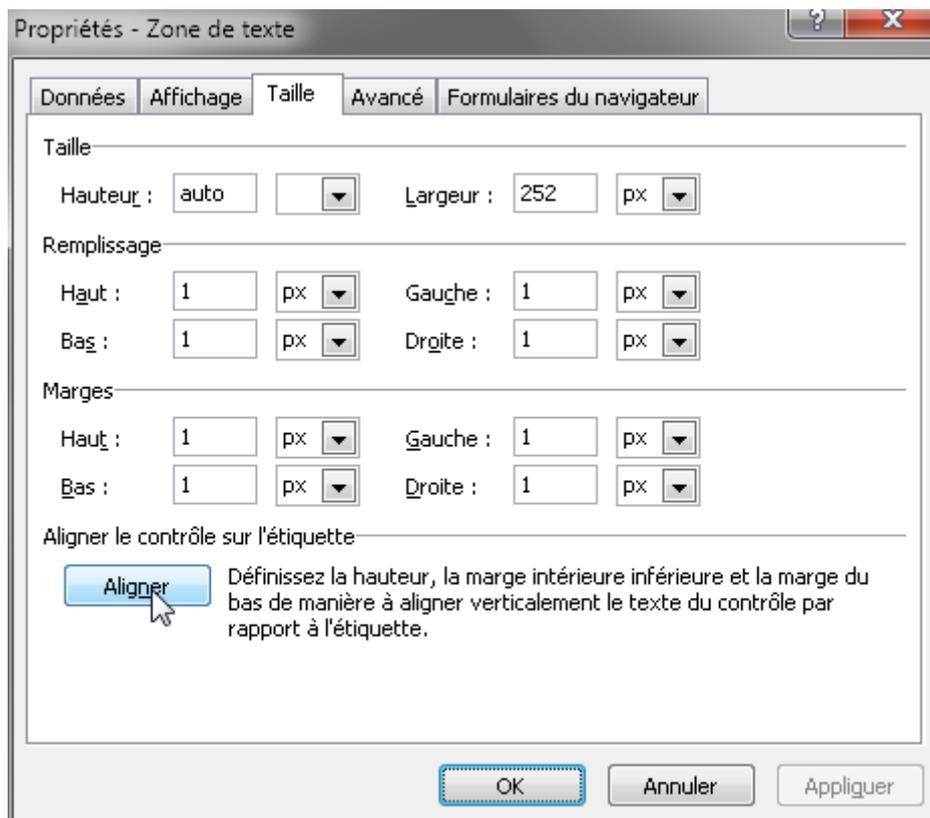
La majorité des propriétés de ce champ sont triviales. Nous allons donc présenter ici seulement quelques problématiques relatives à la jeunesse d'InfoPath et souvent demandées dans les entreprises relativement à ce champ.

#### 9.2.1.1.1 Aligner avec le texte d'étiquette à l'impression

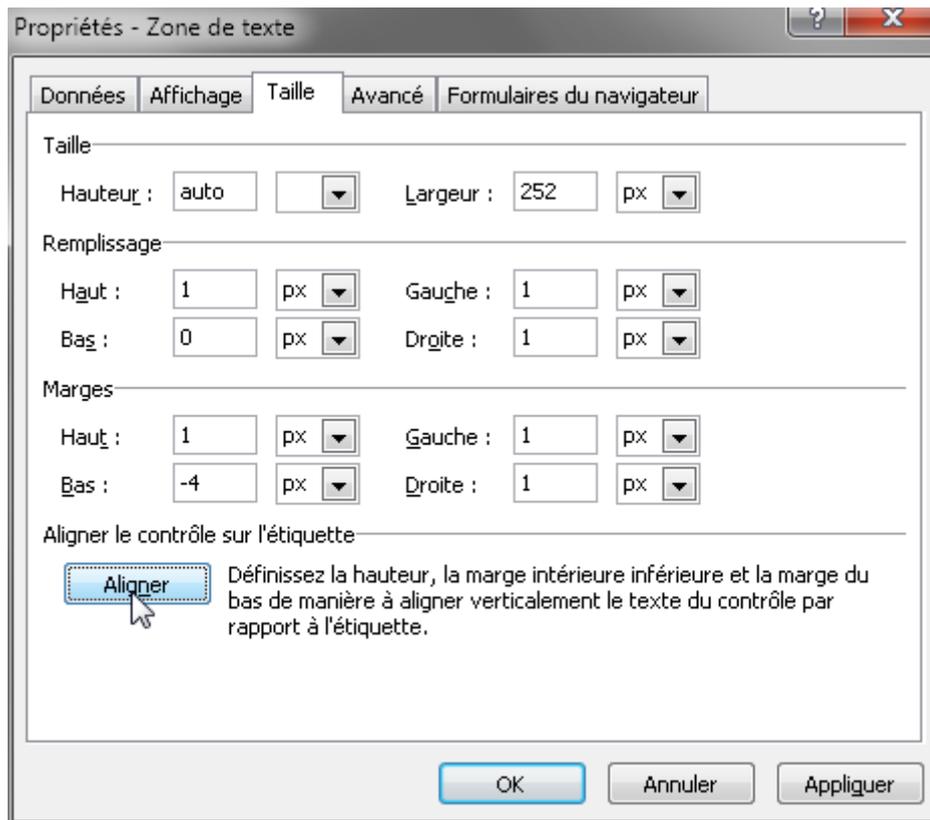
Avant InfoPath 2007 il fallait écrire les étiquettes des Zones de texte dans une cellule avec alignement vertical centré pour et une cellule avec la zone de texte pour ne pas avoir un décalage tel que ci-dessous à l'impression (ou créer une vue indépendante... mais bref c'était dans tous les cas du bricolage):

Nom:

Depuis la version 2010, dans les propriétés de la Zone de texte (**et sur certains autres contrôles mais pas tous!!!**) il existe une option *Aligner le contrôle sur l'étiquette* qui permet justement d'éviter cette effet visuel désastreux rapidement:



ce qui après un clic donne:

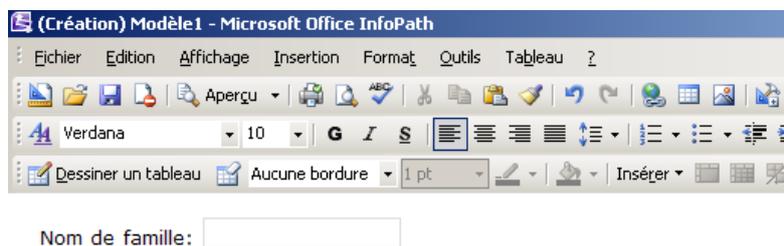


et à l'impression (après avoir enlevé les bordures et mis un fond gris comme le fond ceux qui travaillent avec MS Word...):

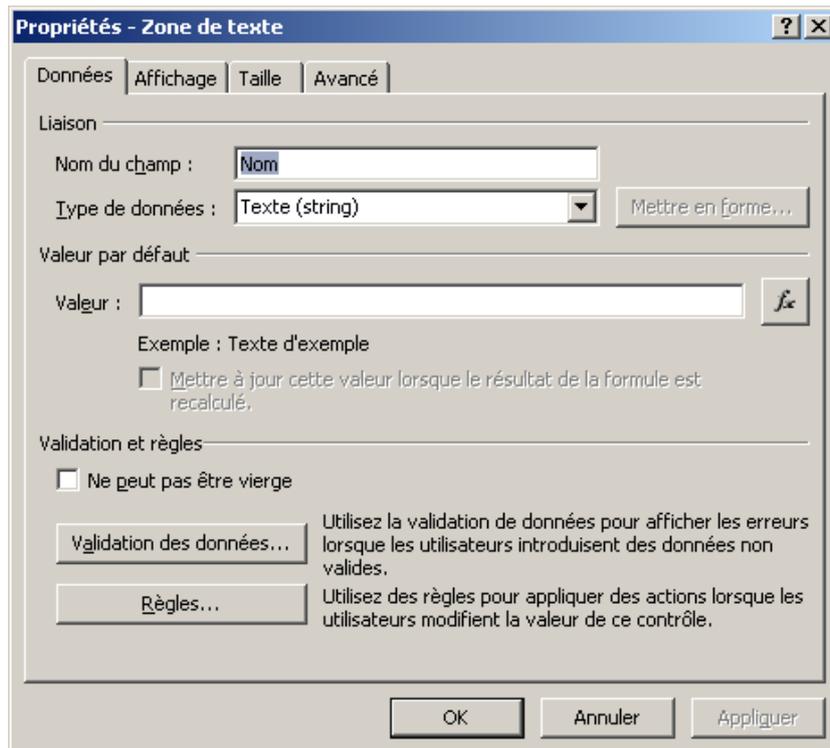
Nom: ISOZ

#### 9.2.1.1.2 Forcer les majuscules

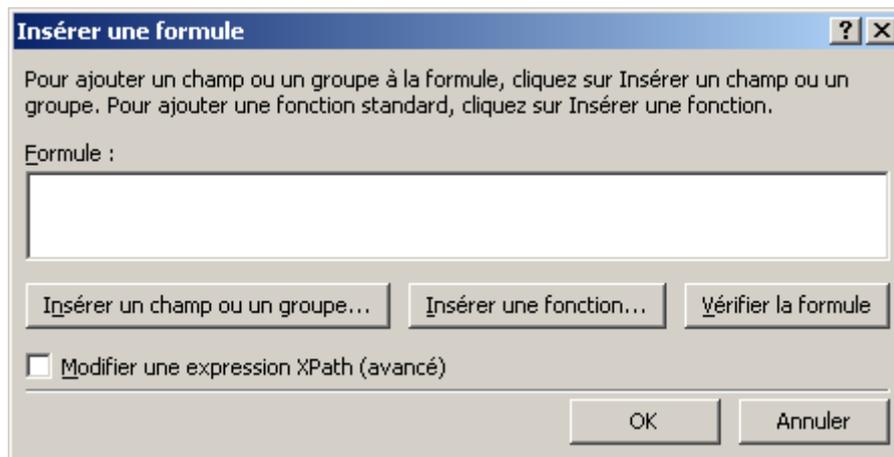
Malheureusement il n'existe pas dans InfoPath 2003/2007 ou 2010 de case à cocher toute simple ou de masque de saisie comme dans MS Access pour forcer les utilisateurs à saisir des données en majuscule. Il faut alors faire la manipulation suivante:



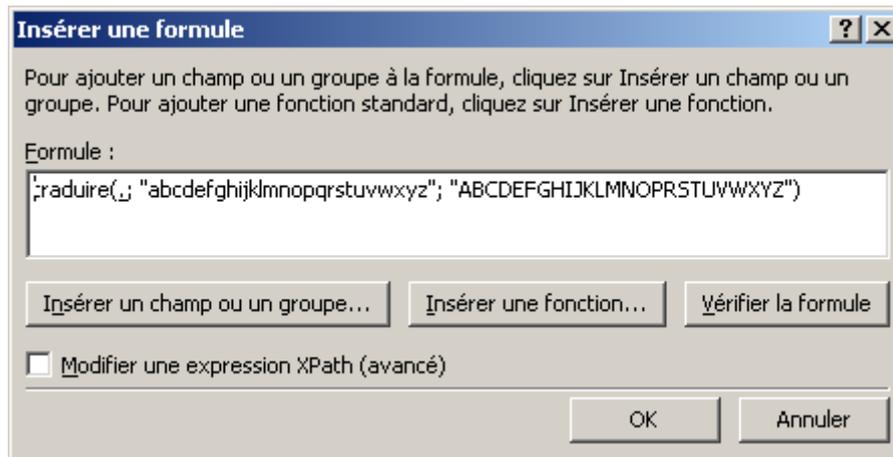
Nous double cliquons sur le champ:



Nous cliquons sur le bouton  et nous aurons:



Nous écrivons (après avoir fait plusieurs essais chez plusieurs clients avec différentes versions d'InfoPath, il semblerait que c'est cette manière d'écrire la formule qui fonctionne le mieux):



Pour avoir que la première lettre en majuscules:

```
concat(translate(substring(NomChamp, 1, 1), "abcdefghijklmnopqrstuvwxyz",  
"ABCDEFGHIJKLMNOPQRSTUVWXYZ"), substring(NomChamp, 2, string-length(NomChamp)))
```

Évidemment, libre à l'utilisateur d'y ajouter d'autres lettres avec les accents par exemple. Dès lors, quand l'utilisateur fera la saisie et que le champ perdra le focus, nous aurons automatiquement les caractères changés en majuscules:



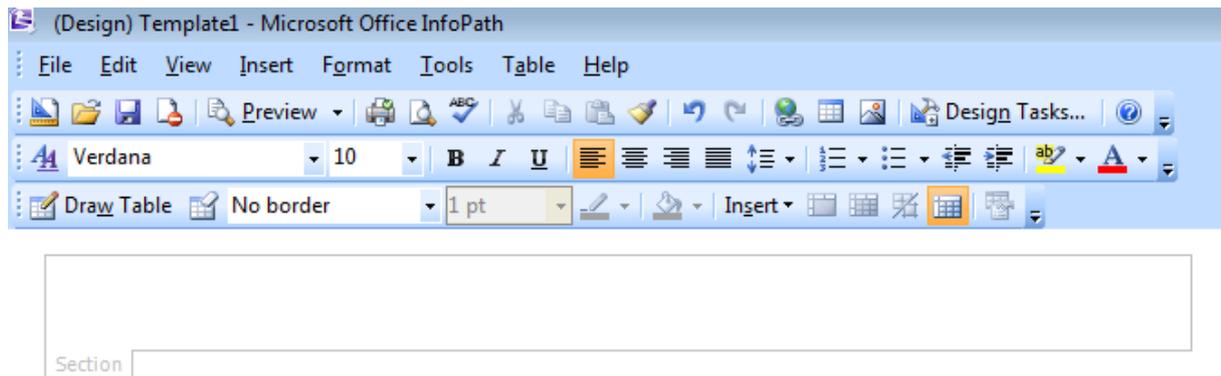
### 9.2.1.2 Contrôles de Section et Section optionnelle

Commençons par les deux contrôles suivants:

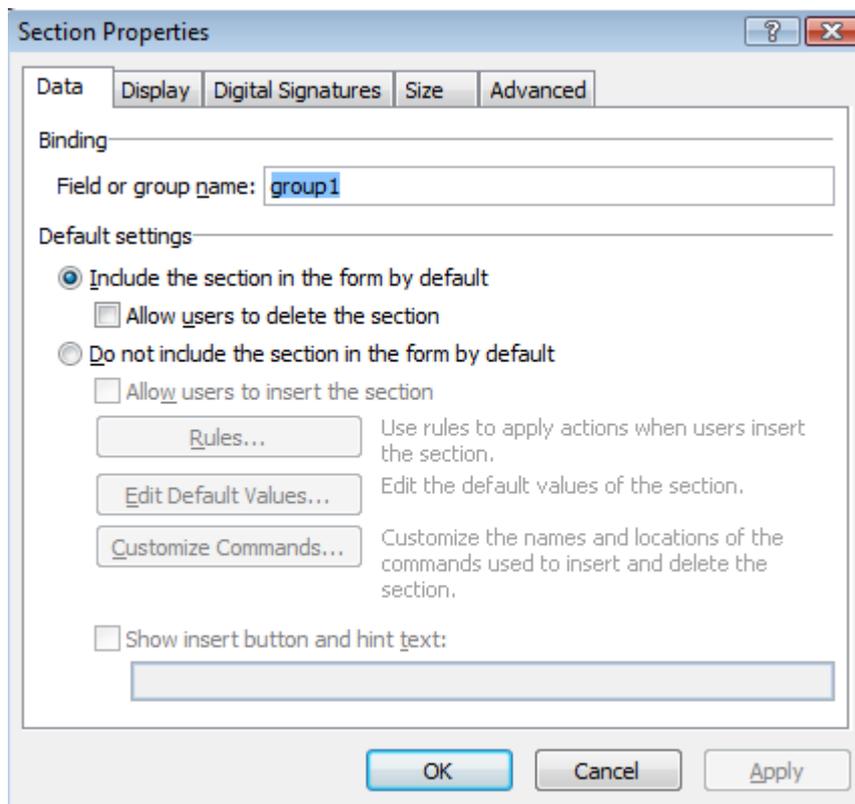


Pourquoi ces deux-là? Car ce sont exactement les mêmes à une petite différence près. Quand vous insérez une *Section*:

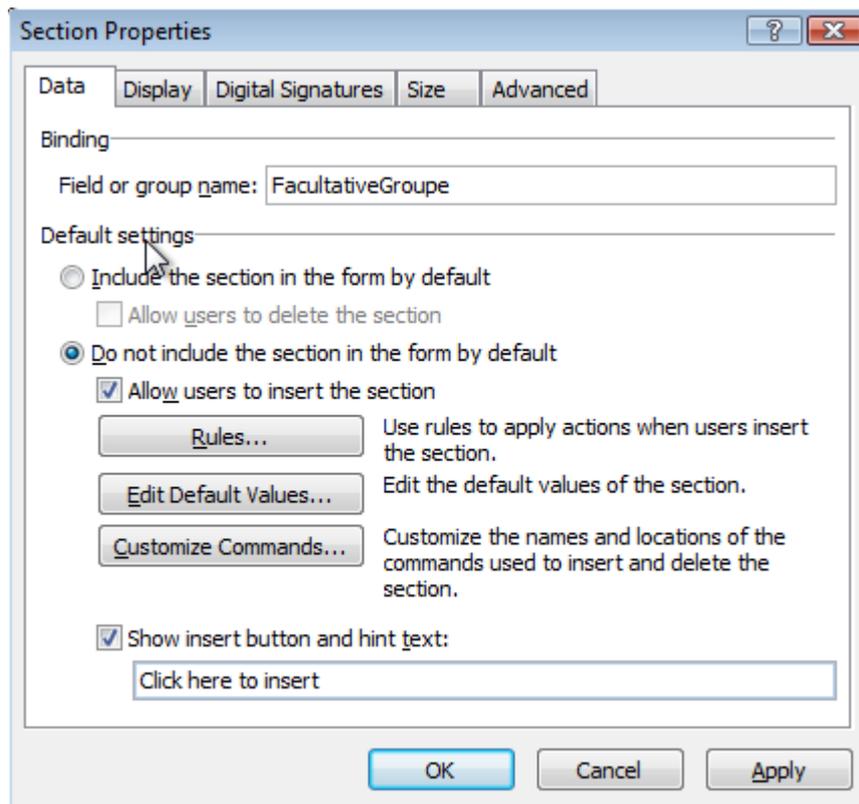
Vincent ISOZ



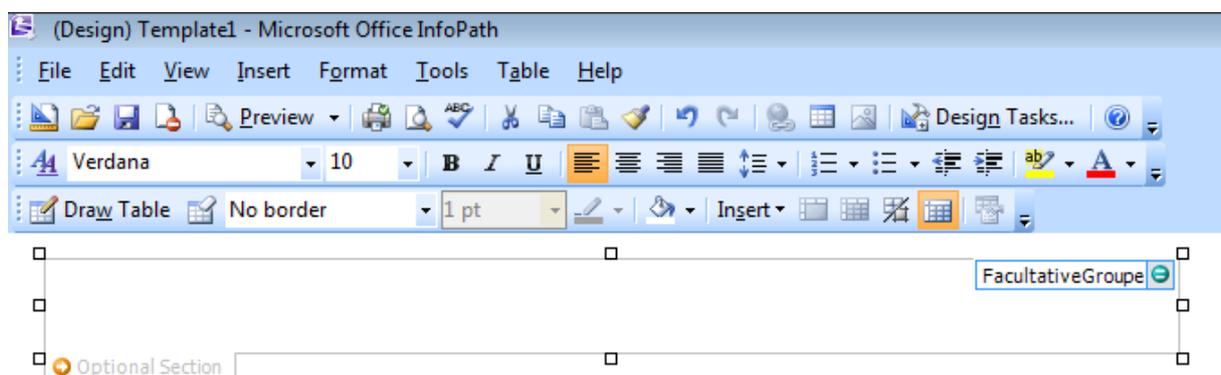
Il vous suffit de double cliquer dessus:



et d'activer *Do not include the section in the form by default* et *Allow users to insert te section* et de taper une légende dans *Show insert button and hint text*:

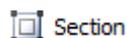


et si vous validez par OK vous aurez:

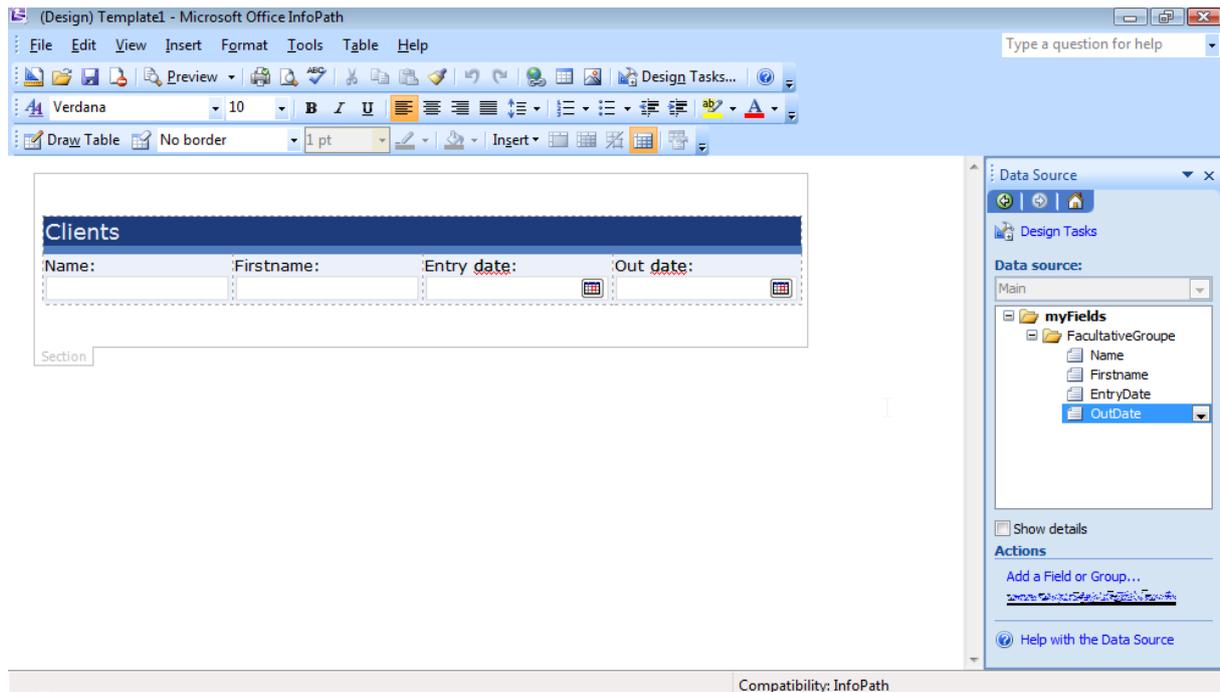


Soit exactement la même chose qu'une optional section. La différence entre les deux est juste une question de gain de temps.

Sinon quel est l'intérêt d'utiliser une *Section* non facultative:



eh bien c'est simple mais il faut le savoir... cela permet de respecter une structure XML/XSD qui vous serait imposée. Effectivement, si vous regardez la structure de votre source de données, vous aurez:



### 9.2.1.3 Contrôles Tableau extensible et Section extensible

Les deux autres champs qui portent souvent à confusions sont:



En fait la différence fonctionnelle est simple! Un tableau extensible contient au moins une ligne alors que la section peut ne pas être utilisée du tout.

Il y a une autre différence qui est purement esthétique. Le tableau se présente en colonnes, comme un saisi en mode feuille de données, la section étant plus adaptée à une forme de sous-formulaire, d'encadré.

Cependant, la structure XML obtenue derrière est la même!

### 9.2.1.4 Contrôle Section récursive extensible

Rarement utilisé car peu exploitable mais utile pour ceux qui n'ont pas MS Office Visio et qui doivent proposer des structures d'organisation de projets, de produits, d'organisation, de compétences ou autres:

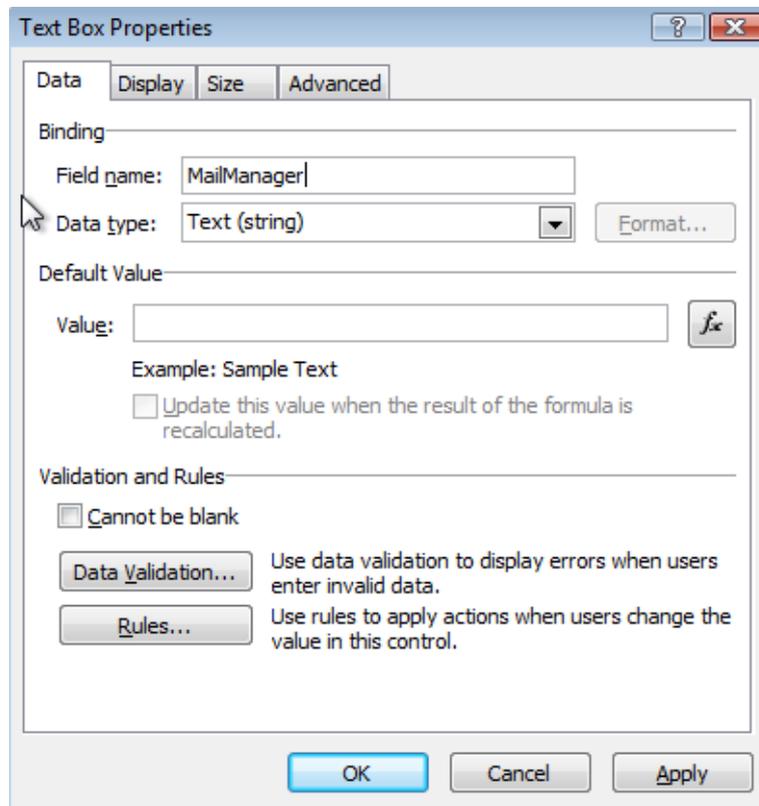


Une section récursive extensible est en fait une section extensible que l'on peut insérer dans elle-même. Par défaut, elle est créée en contenant déjà une deuxième occurrence d'elle-même.

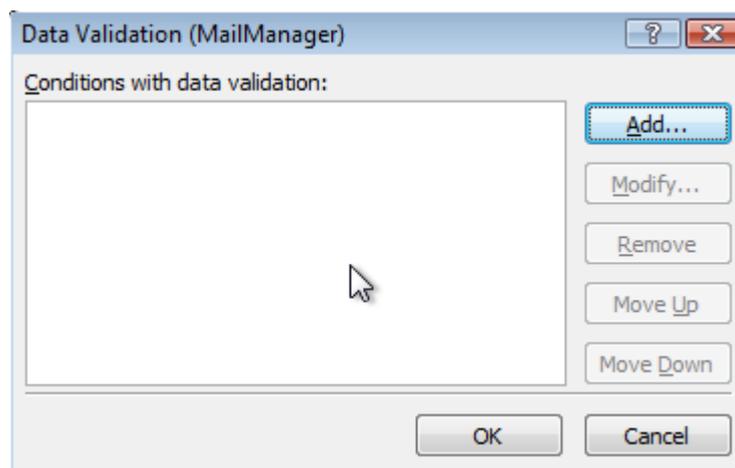
Elle peut être utilisée pour les organigrammes du type PBS, WBS, OBS, etc. (voir mon livre électronique sur la gestion de projets).

## 9.2.2 Utiliser les Regular Expression

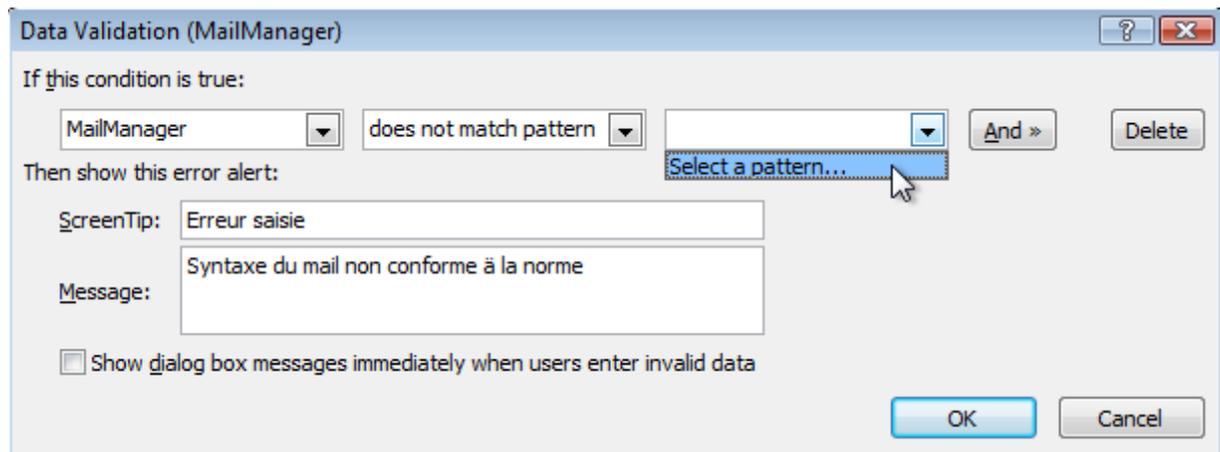
Le lecteur aura remarqué qu'il y a deux champs qui attendent la saisie d'un e-mail. Malheureusement il arrive encore en 2012 que des employés dans les entreprises ne savent pas écrire correctement une adresse e-mail (ou autre chose). Pour contrôler cela, il vous suffira de double cliquer sur le champ:



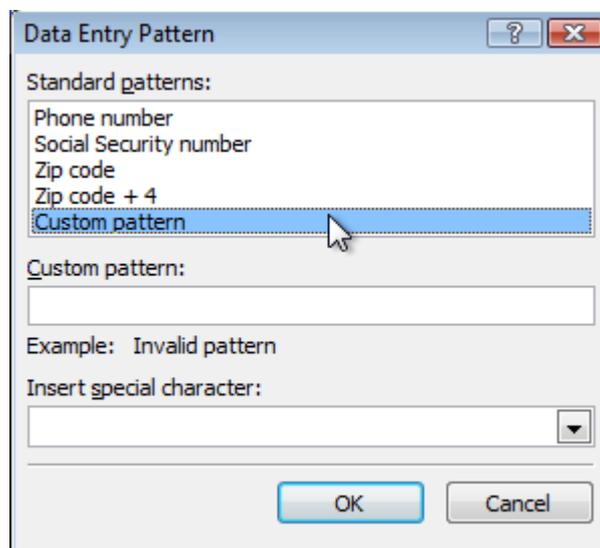
D'aller cliquez sur le bout *Data Validation...*:



de cliquer sur *Add...*:



Il vaut mieux activer *Show dialog box messages immediately when users enter invalid data*. En cliquant sur *Select a pattern...* vous aurez alors:



Par exemple pour la majorité des e-mail vous pourrez mettre:

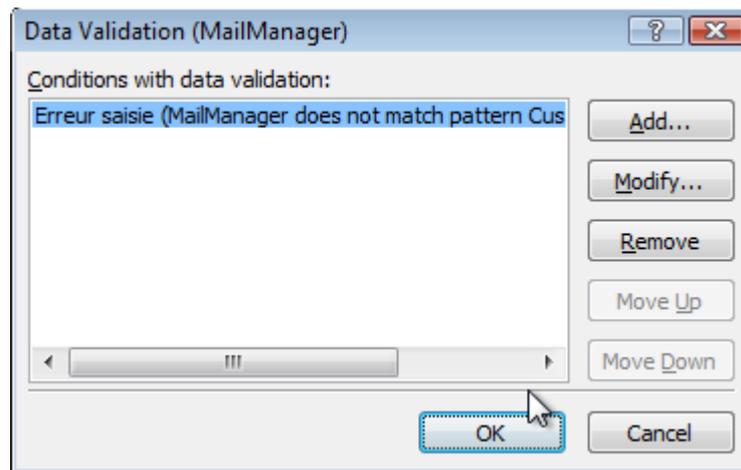
$$[a-zA-Z0-9_-\.\.]+\@[a-zA-Z0-9-\.\.][a-zA-Z]{2,4}$$

mais il faut faire attention aux DNS avec les sous-domaines. Il existe une autre solution un peu plus flexible:

$$.+@.\.+$$

Raison pour laquelle la Regular Expression des e-mail qui était disponible par défaut dans InfoPath 2003, n'y est plus dans InfoPath 2007 (comme l'atteste la capture d'écran ci-dessus faite avec InfoPath 2007).

Après validation deux fois par *OK* nous aurons:



Il n'y pas plus qu'à tester et vous verrez que cela fonctionne comme une lettre à la poste. On trouvera ici comment construire des RegEx complexes:

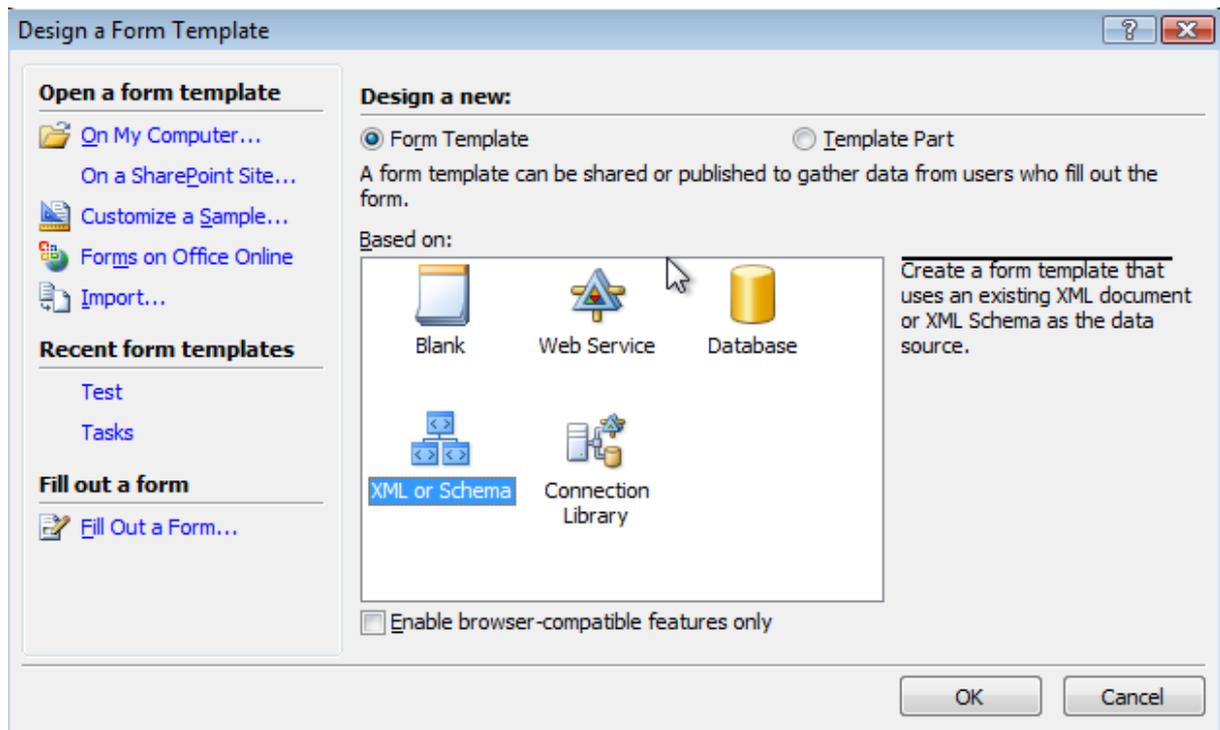
<http://www.regexlib.com>

mais malheureusement Microsoft ne respecte visiblement pas les standards internationaux. Il faut systématiquement s'arracher les cheveux ou faire du code.

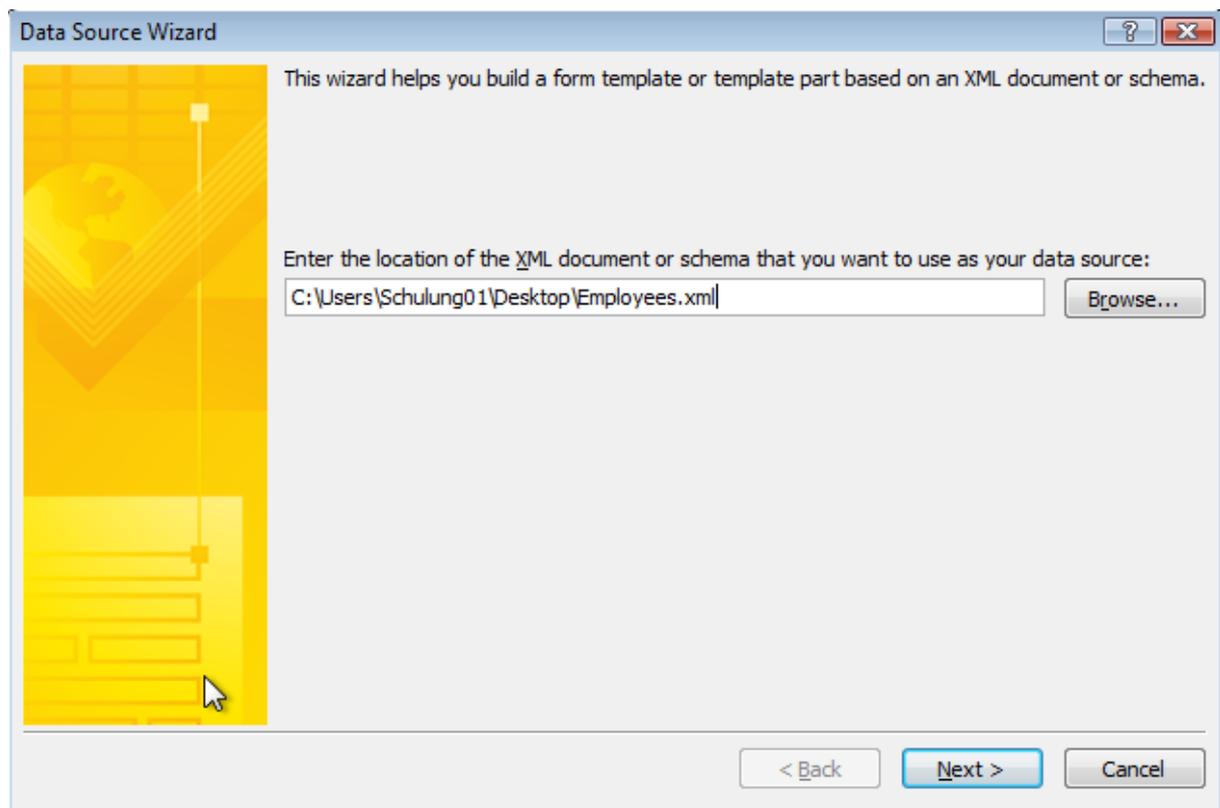
### 9.2.3 Liste LookUp

Nous allons juste dans cet exercice indiquer une chose qui est trop souvent demandée sur les forums: nous aimerions savoir comment lorsqu'un employé est choisi dans une combobox faire apparaître dans un champ de texte son type de poste automatiquement (nous partons sur la base du même fichier XML que montré plus haut).

Créez donc un nouveau formulaire vierge en sélectionnant l'option *New from a XML document or Schema*:



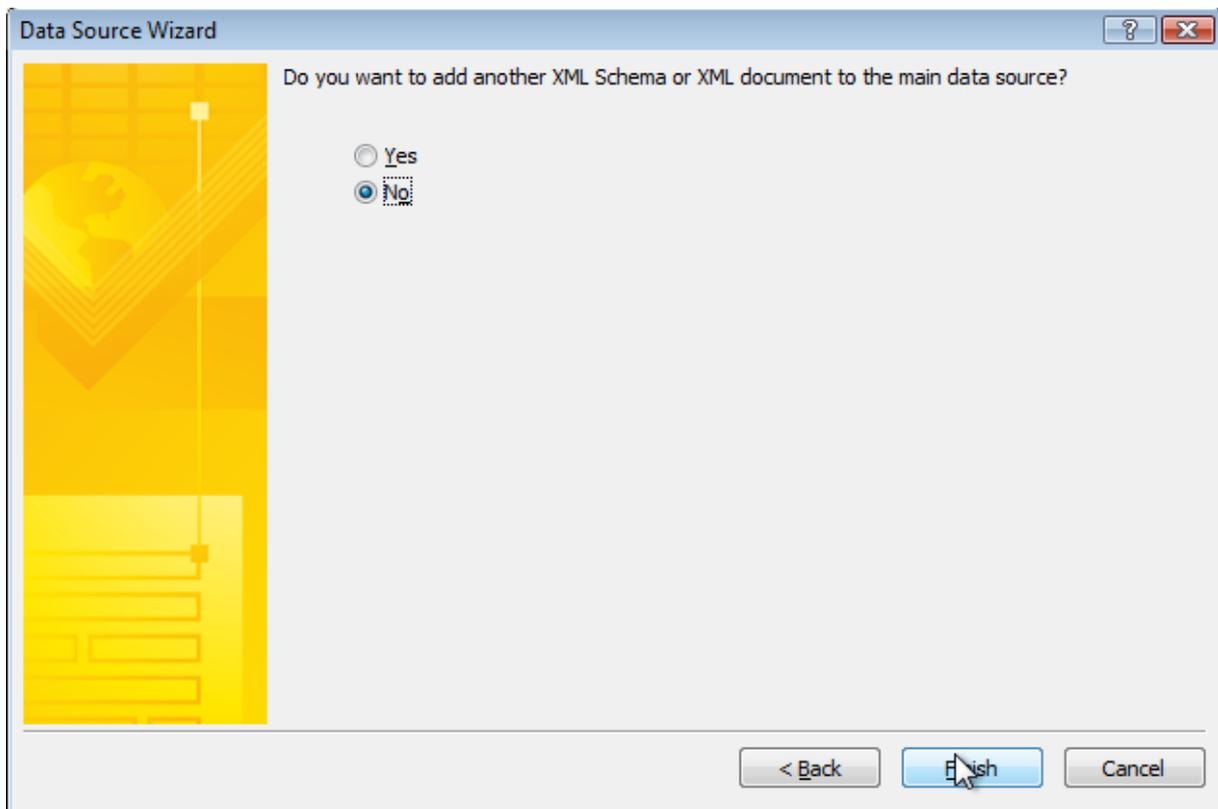
cliquez sur *OK*:



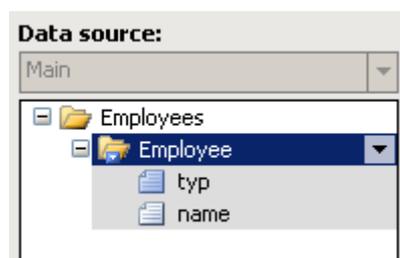
et dans l'assistant, allez chercher le fichier *Employees.xml*:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Employees>
- <Employee typ="Manager">
  <name>Nicolas Mayer</name>
</Employee>
- <Employee typ="Manager">
  <name>Forster Laurence</name>
</Employee>
- <Employee typ="Consultant">
  <name>Isoz Vincent</name>
</Employee>
- <Employee typ="Consultant">
  <name>Weber Olivier</name>
</Employee>
- <Employee typ="Consultant">
  <name>Fournier Fabrice</name>
</Employee>
- <Employee typ="Manager">
  <name>Ravinet Valérie</name>
</Employee>
</Employees>
```

Lorsque l'assistant vous demande si vous souhaitez avoir le fichier comme *Data Source* principale, dites *No*:s



Vous devriez alors avoir dans l'onglet *Data Source*:



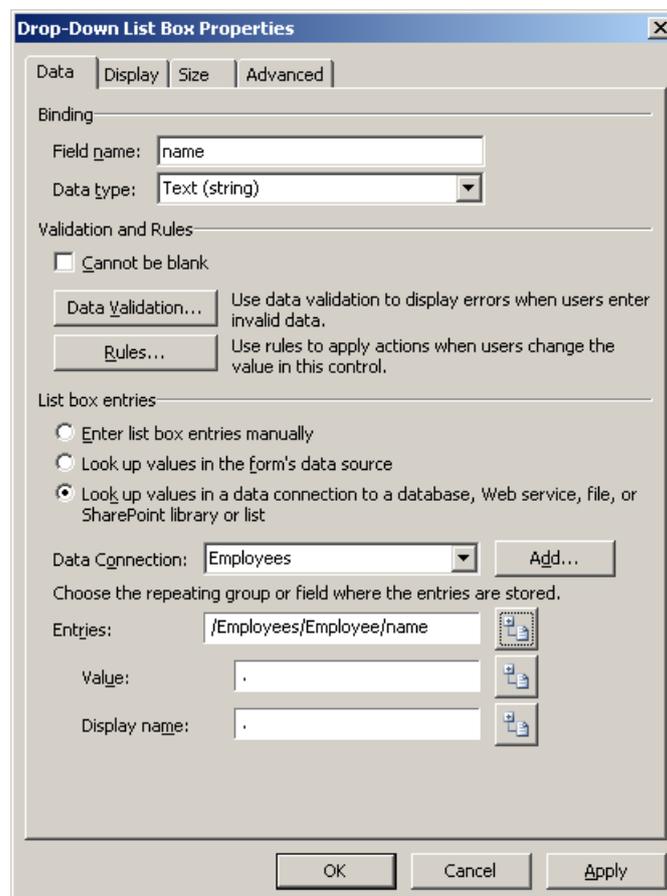
Vincent ISOZ

Faites un clic droit sur le groupe répétitif *Employee* et insérez-le en tant que *Repeating Table*. Mais supprimez les deux champs déjà créés.

En-dessous de la légende *Name* insérez un contrôle de type *Drop-Down* et faites comme l'assistant vous le demande, un binding avec le nœud *name* de la data source *Main*.

En-dessous de la légende *Typ* insérez un contrôle de type *Text box* et faites comme l'assistant vous le demande, un binding avec le nœud *typ* de la data source *Main*.

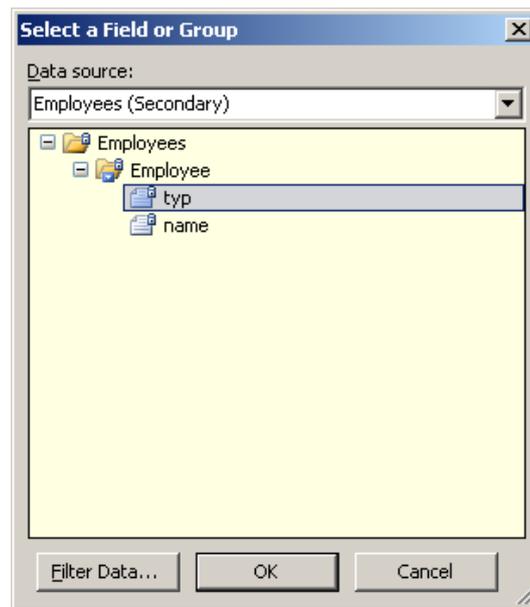
Ensuite, faites un double clic sur la liste déroulante et sélectionnez l'option *Look up values in a data connection, ...* et cliquez sur *Add*. Allez chercher le document XML *Employees.xml* et en particulier le nœud *name* de celui-ci (libre à vous de choisir si vous voulez fusionner le fichier XML dans le fichier InfoPath ou pas):



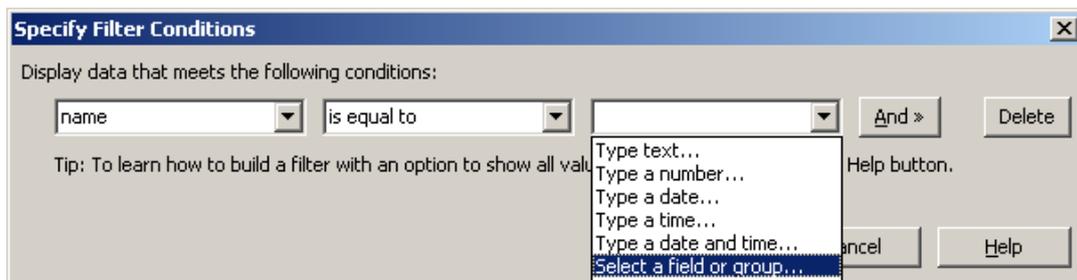
Vous avez maintenant une liste déroulante dans laquelle on peut choisir le nom des employés. Il ne nous reste plus qu'à travailler sur le champ *Text Box* qui devrait automatiquement afficher le type d'employé.

Pour ce faire, effectuez un double clic sur ce dernier champ et dans l'onglet *Display* mettez qu'il est en *Read Only*. Revenez dans l'onglet *Data* et dans la zone *Default Value* sélectionnez le bouton .

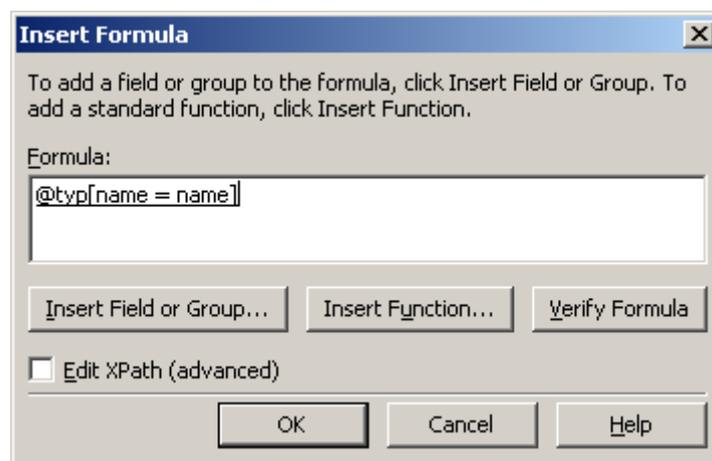
Ensuite, dans la boîte de dialogue qui apparaît, cliquez sur le bouton *Insert Field or Group...* Sélectionnez la source *Employees* et le nœud *typ*:



et cliquez sur *Filter Data* et ensuite sur *Add*:



et prenez le nœud *name* de la source *Main*. En validant par *OK* plusieurs fois, vous aurez:



et voilà... il ne vous reste plus qu'à tester.

### 9.2.4 Publier versus Enregistrer

L'étape de création étant terminée, il vous reste à enregistrer (ou) publier votre formulaire. Ces deux concepts sont souvent source de confusions chez les utilisateurs d'InfoPath.

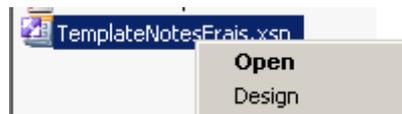
Vincent ISOZ

Une fois votre formulaire terminé, il est d'abord recommandé de l'enregistrer comme un fichier normal. Le document \*.xsn pourra alors être vu comme l'élément d'origine du développeur de formulaire.

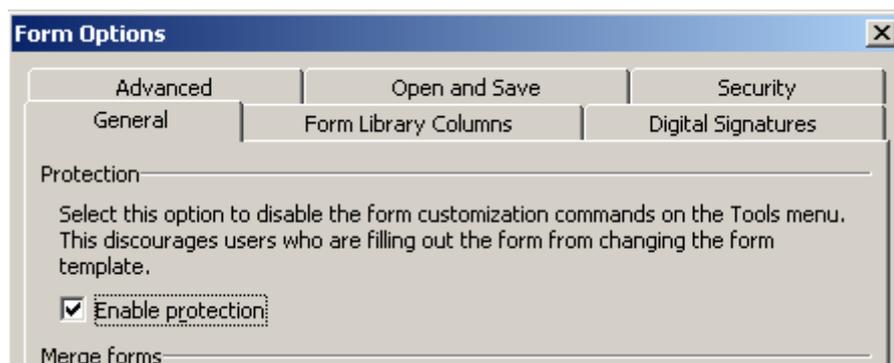
En d'autres termes, dans une entreprise il y aura (normalement) toujours le formulaire local en possession du développeur de formulaire (sur sa machine uniquement) seulement en cas de modification à effectuer) et le formulaire publié (sur un disque réseau, sur SharePoint ou via msi c'est selon) et qui sera utilisé par les personnes ayant de la saisie à effectuer.

#### 9.2.4.1 Protection contre la conception

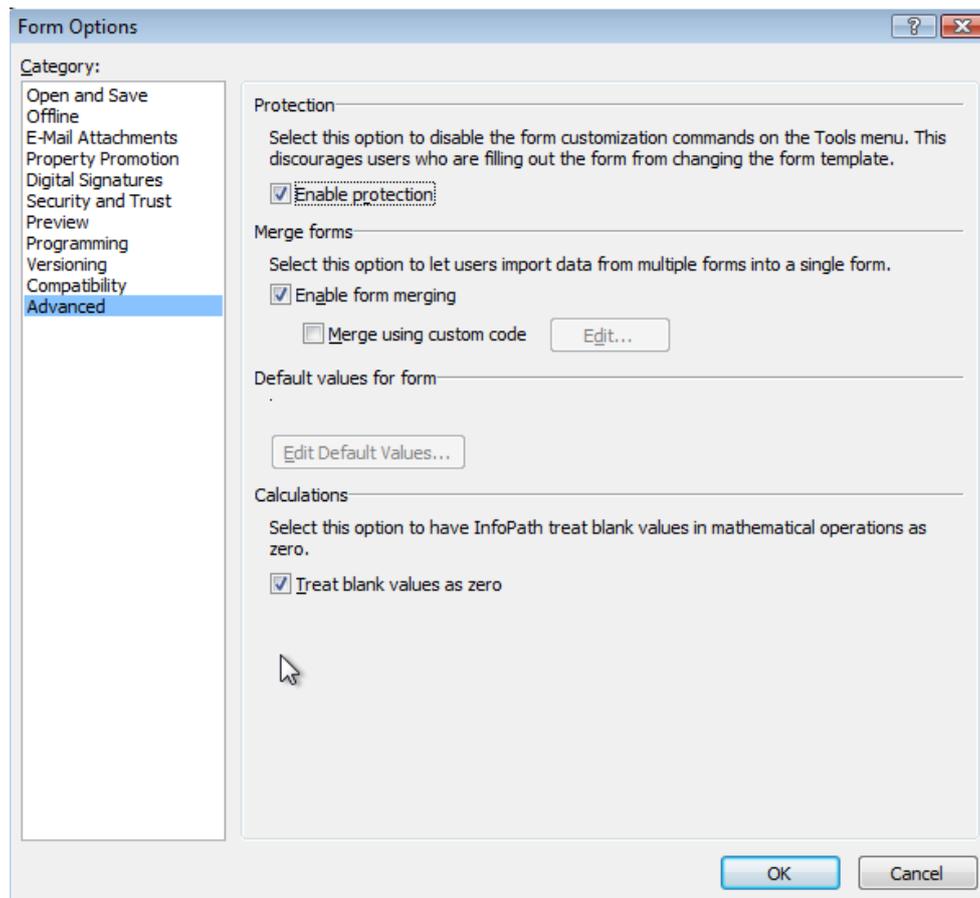
Prenons comme exemple notre formulaire de notes de frais qui est enregistré dans *Mes Documents*. Pour l'ouvrir en mode création faisons un clic droit et choisissons *Design*:



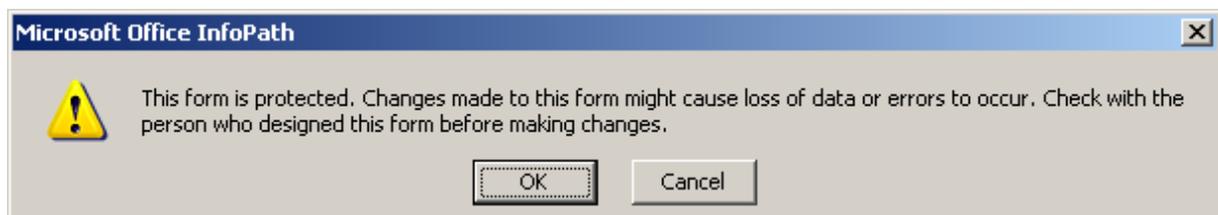
Si le formulaire a été protégé initialement par le développeur via le menu *Tools/Form Options/Enable protection*:



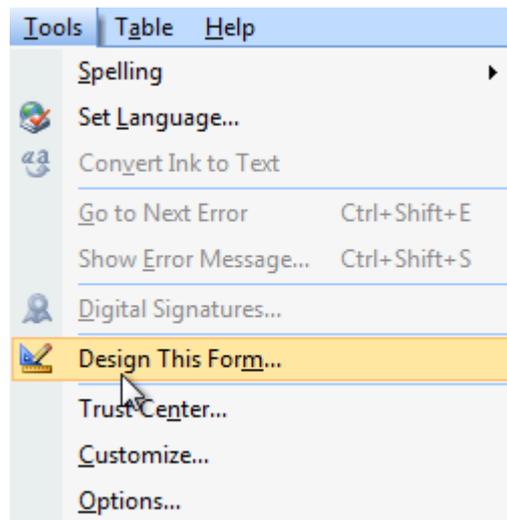
ou dans InfoPath 2007:



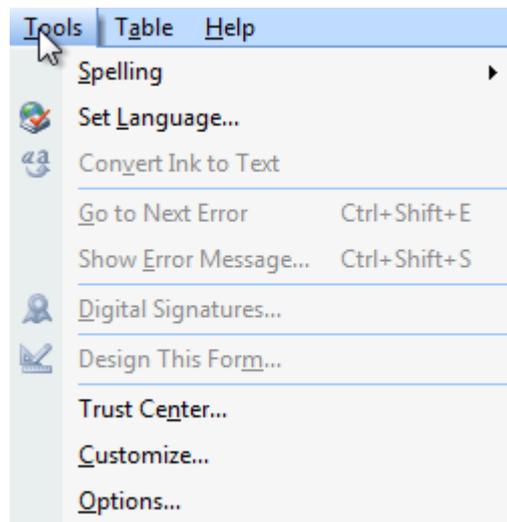
alors vous aurez toujours le message suivant à l'ouverture en mode *Design*:



Dès lors, le formulaire est ouvert en mode création. Rien à signaler pour le développeur! Alors à quoi est utile la case *Enable protection* important avant la publication??? Eh bien simplement à éviter qu'un utilisateur un peu malin (mais pas assez pour deviner l'option *Design* au clic droit dans MS Windows) fasse la chose suivante lorsqu'il double clique sur un formulaire publié pour le remplir:



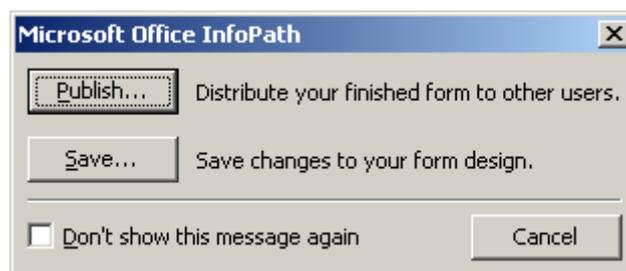
Car effectivement, si vous ne protégez pas votre formulaire, il va probablement voir l'option et en modifier la structure. Par contre, si vous activez la protection, l'utilisateur qui double clique sur le modèle de formulaire pour le remplir aura alors dans le menu *Tools*:



Bingo! Étape préliminaire terminée!

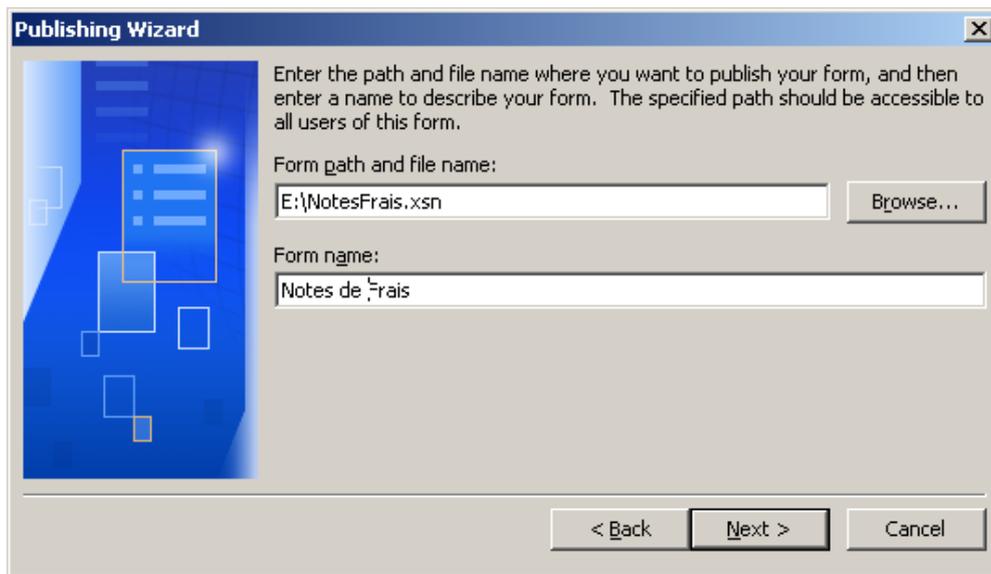
#### 9.2.4.2 Publication

Si le formulaire est terminé nous le publions sur un disque réseau ou sur SharePoint. Prenons l'exemple du disque réseau qui sera arbitrairement choisi comme ayant la lettre *E*. Pour publier, nous allons donc dans le menu *File/Save as*:

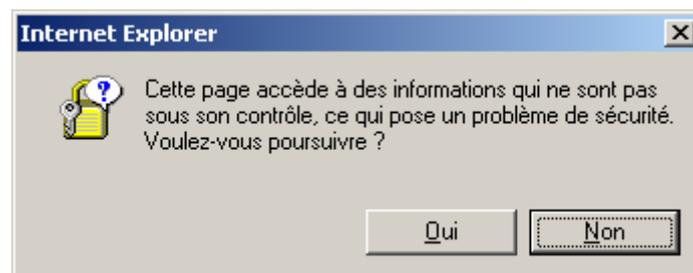


Vincent ISOZ

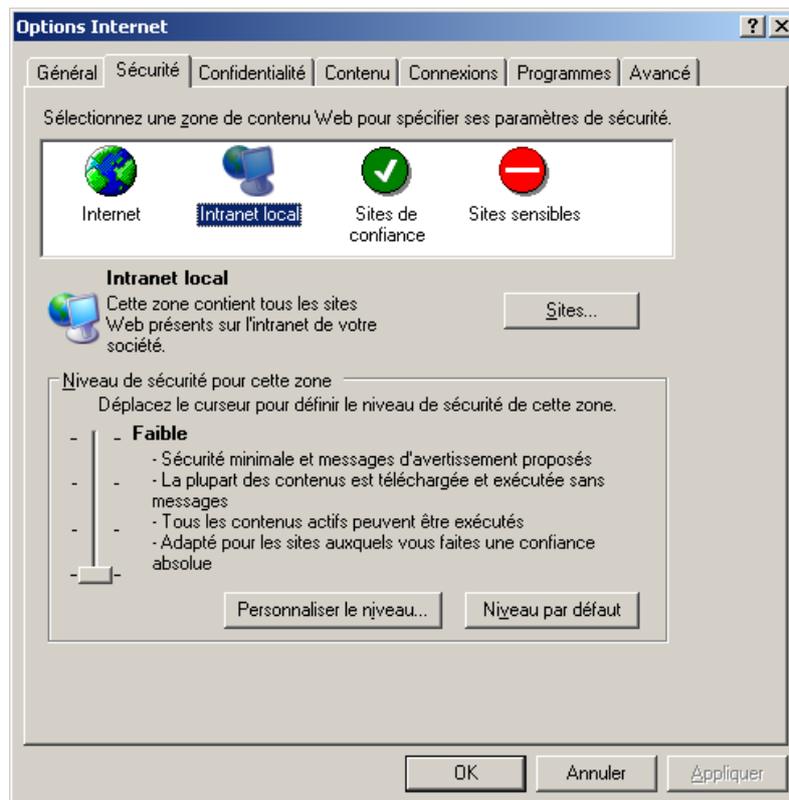
Sélectionnez *Publish* et suivez l'assistant (nous nous intéresserons pas à l'enregistrement qui fait un simple enregistrement...)... pour le publier finalement sous le nom *NotesDeFrais*:



Ensuite, si les utilisateurs (personne ayant juste à saisir l'information) voient le message suivant à l'écran lors de l'ouverture du formulaire à partir de l'endroit publié:



c'est que le formulaire utilise des sources externes comme notre fichier XML contenant les employés. Il est alors recommandé dans Internet Explorer de mettre le niveau de sécurité de l'intranet local sur *Faible*:



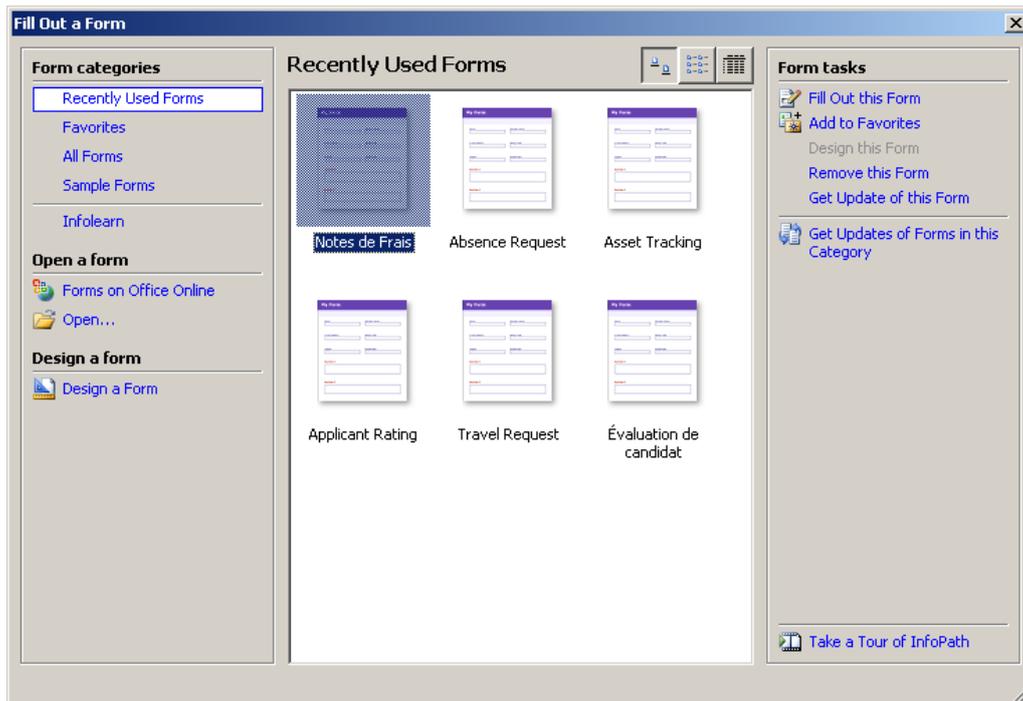
après quoi ce message n'apparaîtra plus.

### 9.2.4.3 Versions des formulaires

Une fois le formulaire ouvert une fois par un utilisateur pour la saisie, il est bon de s'avoir qu'il se trouvera (en cache<sup>1</sup>) dès lors dans la fenêtre suivante à l'ouverture d'InfoPath:

---

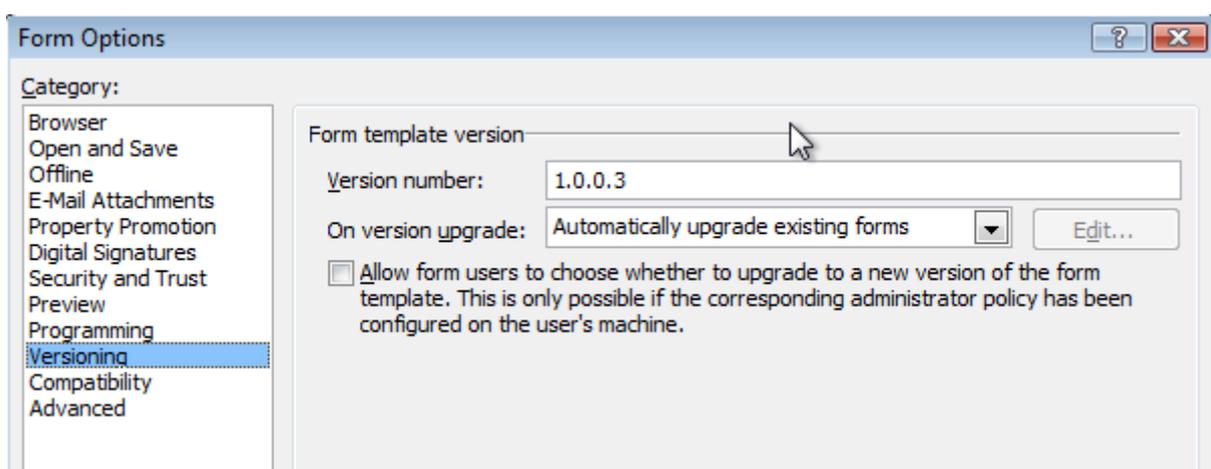
<sup>1</sup> En d'autres termes, si le développeur supprime le formulaire de sa source il est toujours disponible sur les postes clients l'ayant déjà ouvert.



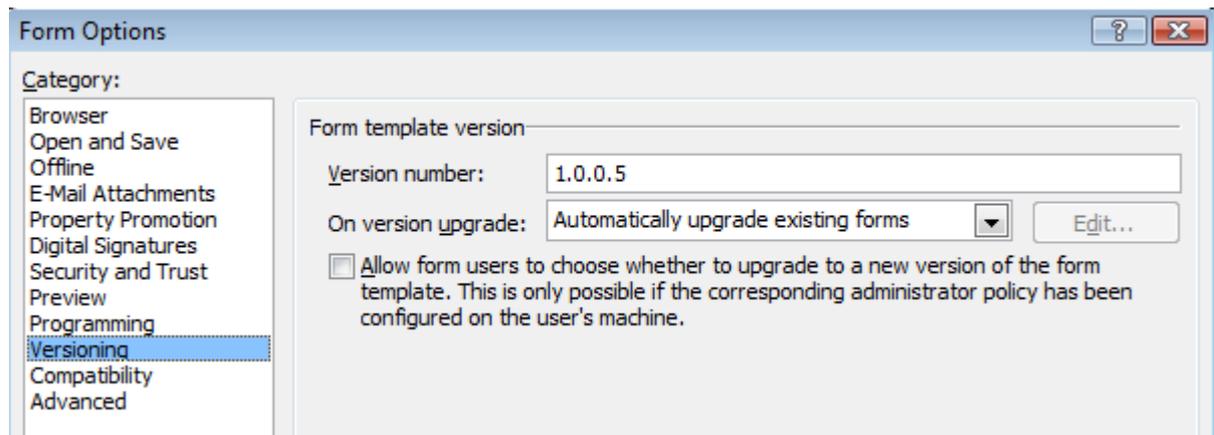
Il est alors possible pour l'utilisateur:

1. En cliquant sur le lien *Remove this Form* cela l'enlèvera de la liste
2. En cliquant sur le lien *Add to favorites* cela l'ajoutera dans la section des favoris disponible depuis le lien *Favorites* sur la gauche de la fenêtre
3. Enfin, le lien *Get Update of this form* sera présenté un peu plus bas car il est dangereux pour les utilisateurs qui oublient de cliquer dessus.

Mais bon... pourquoi parlons-nous de versions? Eh bien c'est simple! Quand vous créez un formulaire et que vous allez dans *Tools/Form Options* vous aurez:



Si vous enregistrez et ensuite publiez le formulaire comme on l'a vu avant vous aurez:



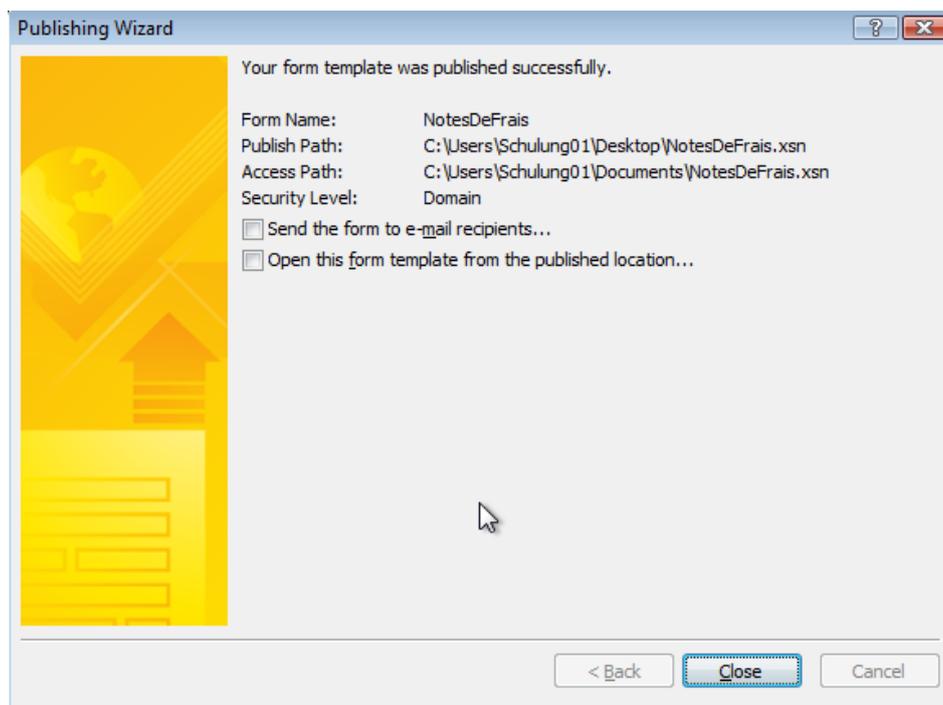
etc. car les numéros s'incrément automatiquement d'une manière spécifique au choix de Microsoft (à moins que vous tapiez votre propre numéro). Pourquoi avons-nous une incrémentation à deux niveaux dans le cas présent? Eh bien parce que l'enregistrement est considéré comme une publication et étant donné que nous avons enregistré et ensuite publié, la version s'est incrémentée de deux unités. Deux questions surviennent alors souvent:

**Question 1:** Quel est l'intérêt d'utiliser *Publish* plutôt que *Save* lorsqu'on travaille avec un disque réseau (dans les autres situations mes clients ont toujours trouvé la différence évident)

**Question 2:** Pourquoi ces versions?

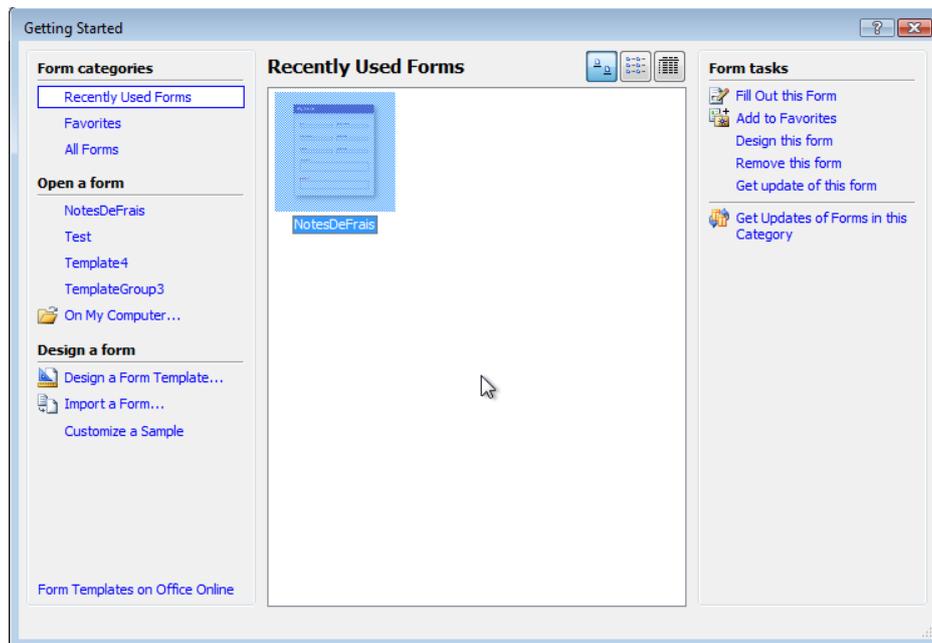
Voici les réponses:

**Réponse 1:** L'intérêt de faire un *Publish* plutôt qu'un *Save* sur un disque réseau est simplement qu'à la fin de l'assistant vous aurez la possibilité de gagner quelques secondes pour envoyer un lien vers le nouveau formulaires à vos collègues et éventuellement tester celui-ci avant d'envoyer la mail:

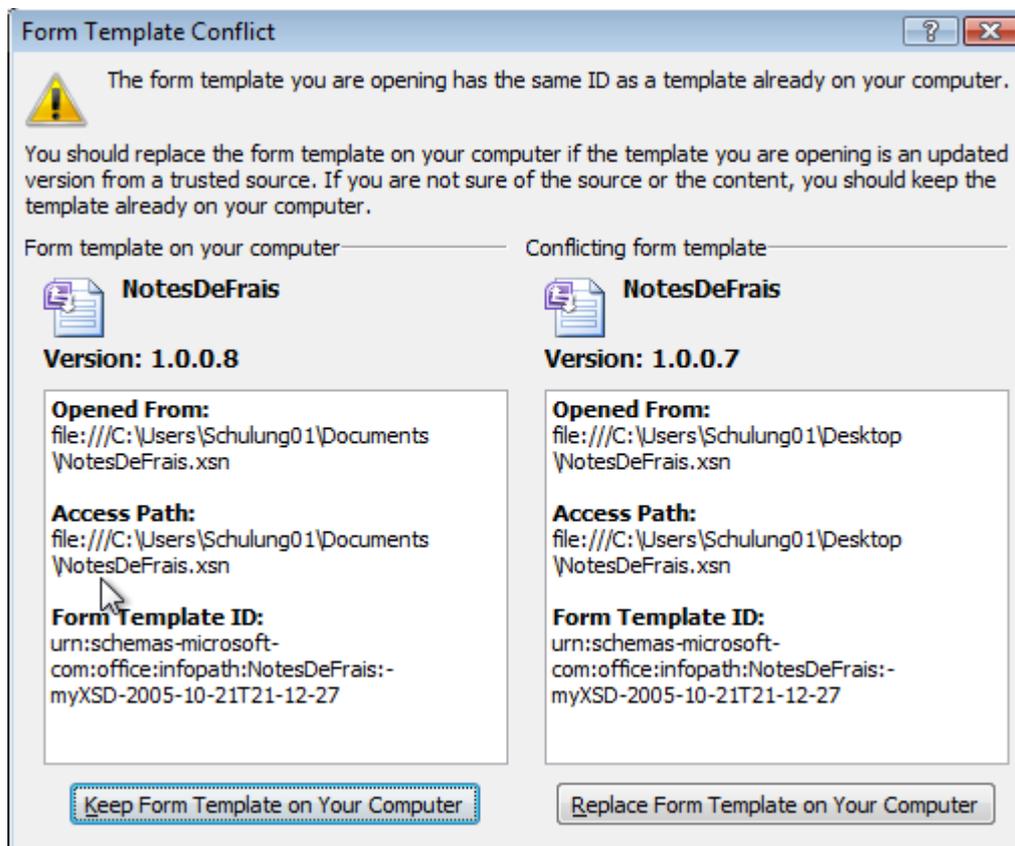


Vincent ISOZ

**Réponse 2:** Si vos collègues ont ouvert et utilisé la première version de votre formulaire ils auront donc dans le cache de leur InfoPath (nous le savons car nous en avons déjà parlé):



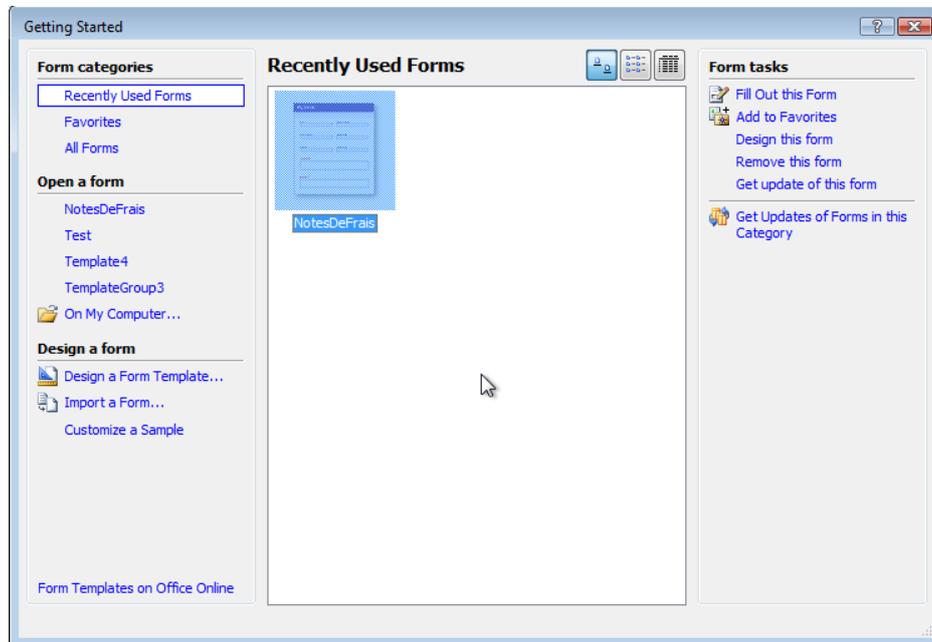
Si de votre côté vous publiez une nouvelle version de votre formulaire et que vos collègues souhaitent remplir le formulaire en passant par le fichier se trouvant sur le disque réseau ils auront alors comme message:



Vincent ISOZ

InfoPath voit que l'utilisateur a un formulaire identique en cache mais dont la version est obsolète et propose à l'utilisateur de la changer. Évidemment ce comportement est hyperrrrr dangereux car si l'utilisateur ne fait pas le bon choix, il va remplir un formulaire obsolète.

De plus, nous voyons aussi le danger que génèrent les utilisateurs qui passent par habitude toujours pas la fenêtre suivante pour saisir un formulaire:



car s'ils ne cliquent pas sur *Get update of this form*, ils utiliseront toujours la vieille version en cache.

Il n'y a de solution parfaite et universelle à ce jour à ce danger. La bonne démarche est de former (éduquer) les utilisateurs finaux ou d'utiliser InfoPath Services.

### 9.2.5 Vider le cache

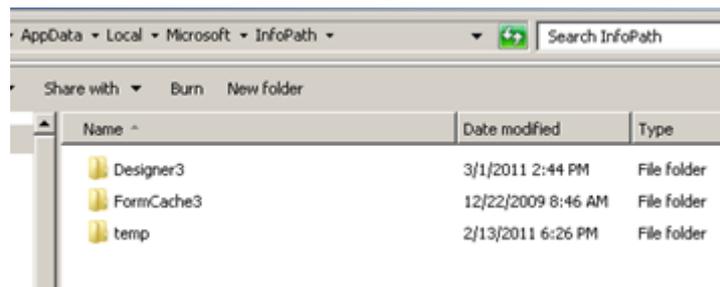
Il est parfois nécessaire pour des questions de démonstration en formation ou lors de l'utilisation avec SharePoint de vider le cache d'InfoPath afin que le logiciel charge la nouvelle version d'un formulaire.

Pour cela, dans **Shift+R**, on tapera:

*InfoPath /cache clearall*

Le cache d'InfoPath 2010 se trouve par dans le dossier:

C:\Documents and Settings\[User]\Local Settings\Application  
Data\Microsoft\InfoPath\FormCache2



## 9.2.6 Signature

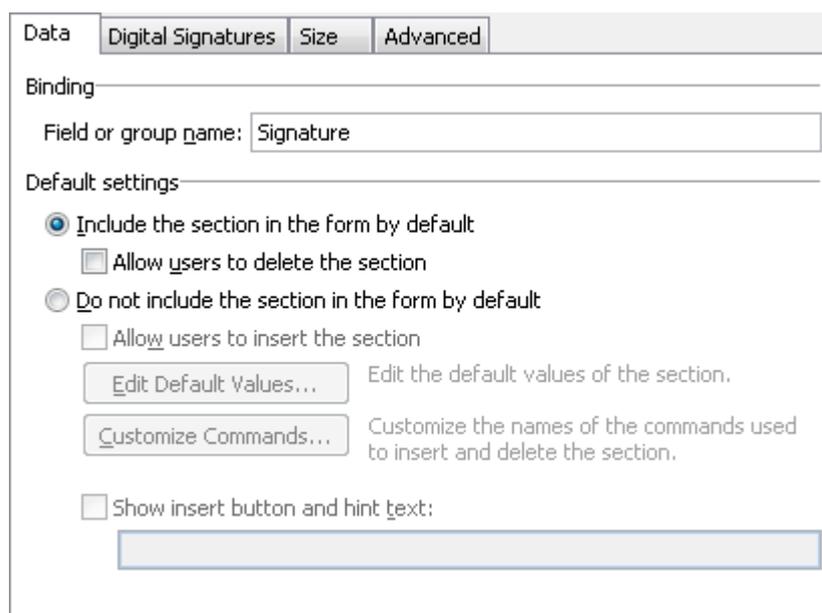
De plus en plus de mes clients signent les formulaires et documents électroniquement. Il est donc devenu très important dans MS InfoPath de créer des zones de signatures. Lorsque nous concevons un modèle de formulaire, nous pouvons donc préciser si les utilisateurs peuvent ajouter des signatures numériques lorsqu'ils complètent un formulaire. Comme nous allons le voir, nous pouvons indiquer si les utilisateurs peuvent signer l'intégralité du formulaire ou juste une partie de celui-ci. Si nous activons les signatures numériques, nous allons voir qu'elles peuvent être ajoutées à une partie (sous-ensemble) du formulaire à choix.

**Une fois signé, le formulaire (ou la section) ne pourra plus être modifié sans que la signature ne soit invalidée.**

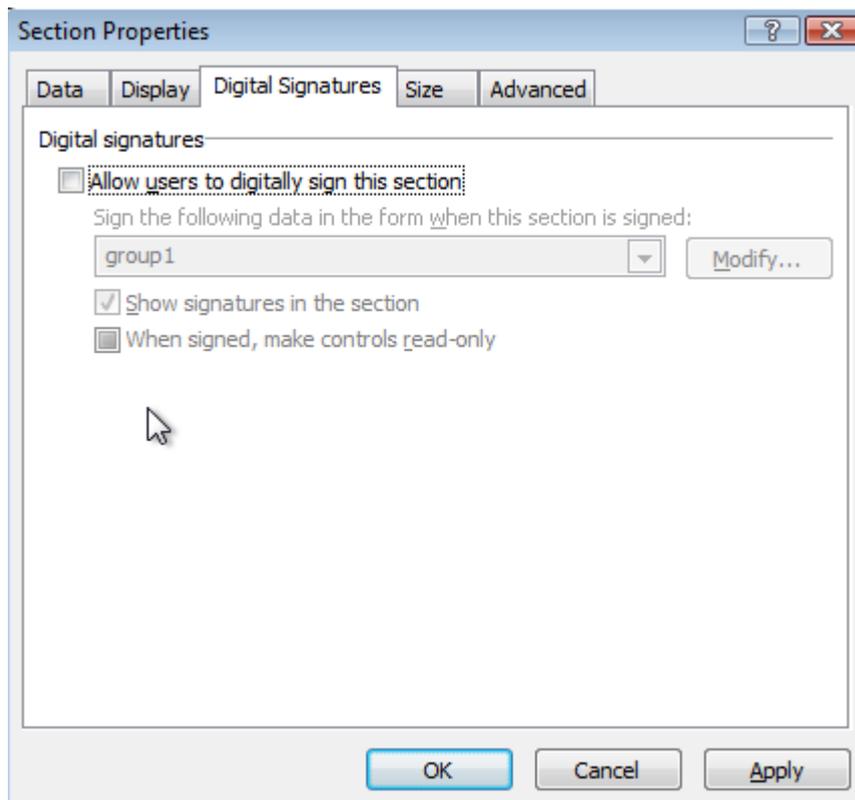
La création de signatures est basée sur le contrôle de type *Section*. Si vous en insérez un dans le formulaire vous aurez:



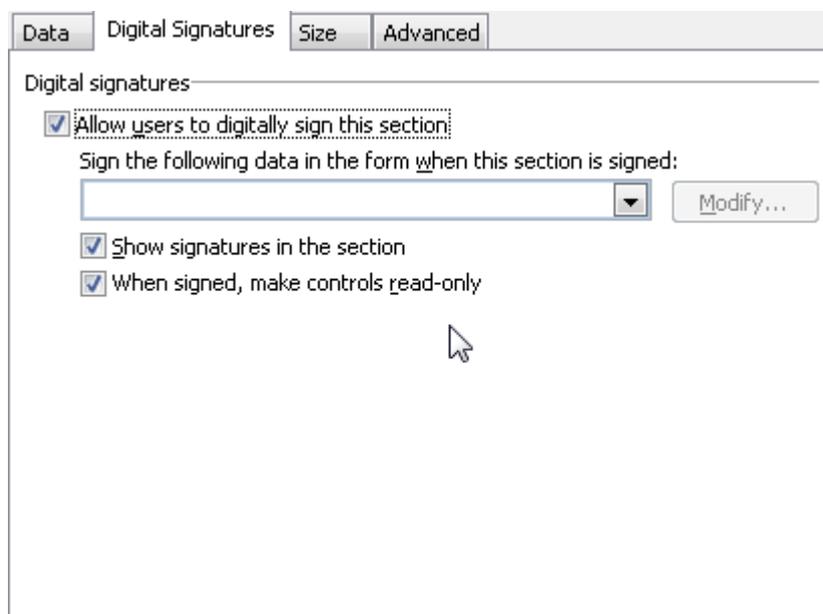
Faites alors un double clic dessus et allez d'abord donner un nom au champ:



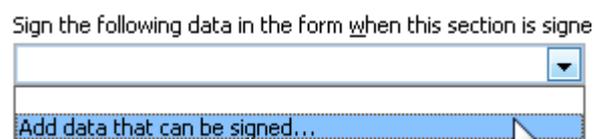
Validez par *OK* et retournez dans les propriétés de la *Section* et allez dans l'onglet *Digital Signatures*:



on coche *Allow users to digitally sign this section* et on aura:



Il faut obligatoirement sélectionner les données à signer:



Vincent ISOZ

On aura alors:



On valide par *OK* et on aura alors lors du remplissage du formulaire:



et cela marche très bien lorsque vous maîtrisez votre clé USB de signature.

Malheureusement, il n'existe cependant pas de case à cocher simple pour forcer la signature du formulaire avant l'envoi dans SharePoint ou par e-mail. Donc voici comment procéder:

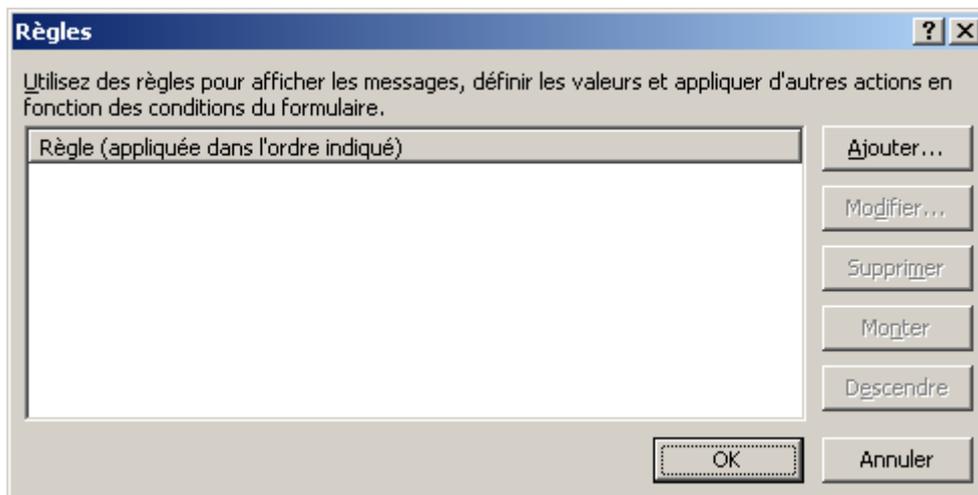
Faites un double clic sur le bouton *Envoyer* de votre formulaire:



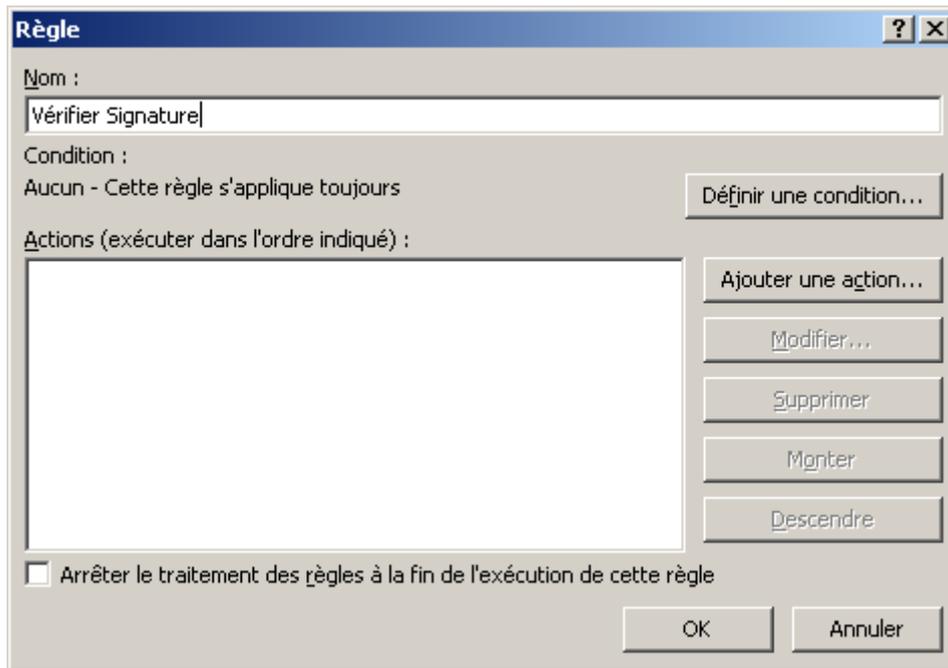
En dans *Action*, choisissez *Règles et code personnalisé*:



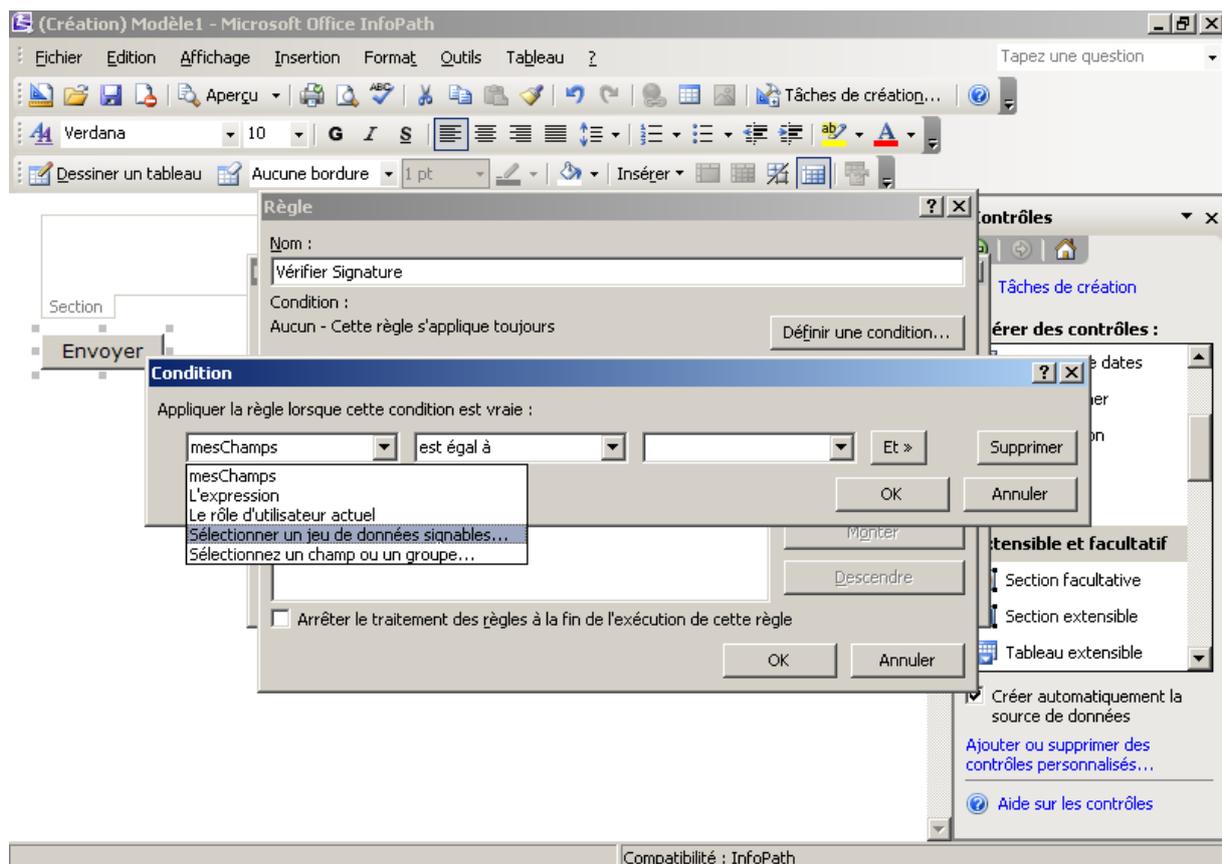
et cliquez sur le bouton *Règles...*:



Cliquez sur *Ajouter...*:



Donnez un nom à votre règle et cliquez sur *Définir une condition...*:



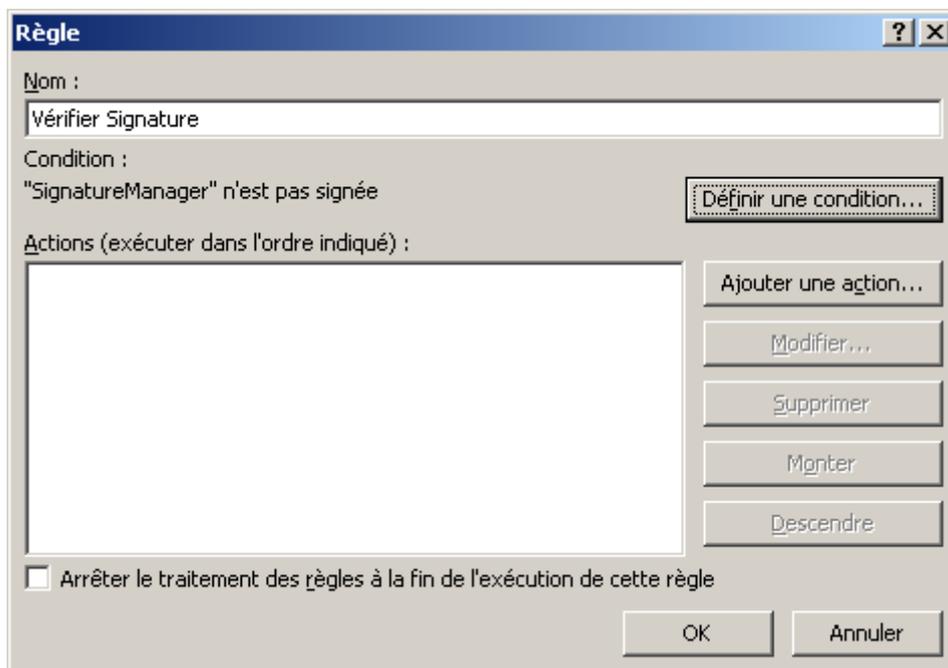
et dans la première liste déroulante, prenez *Sélectionner un jeu de données signables...*:



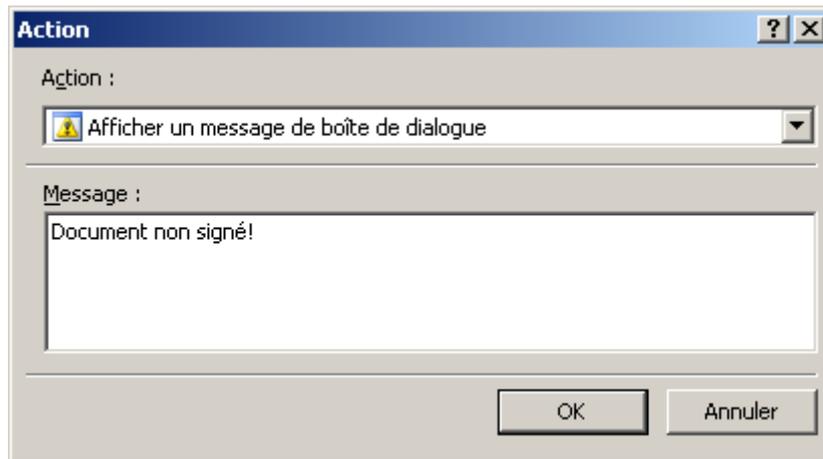
Le système vous propose la section de signature que vous avez créée précédemment:



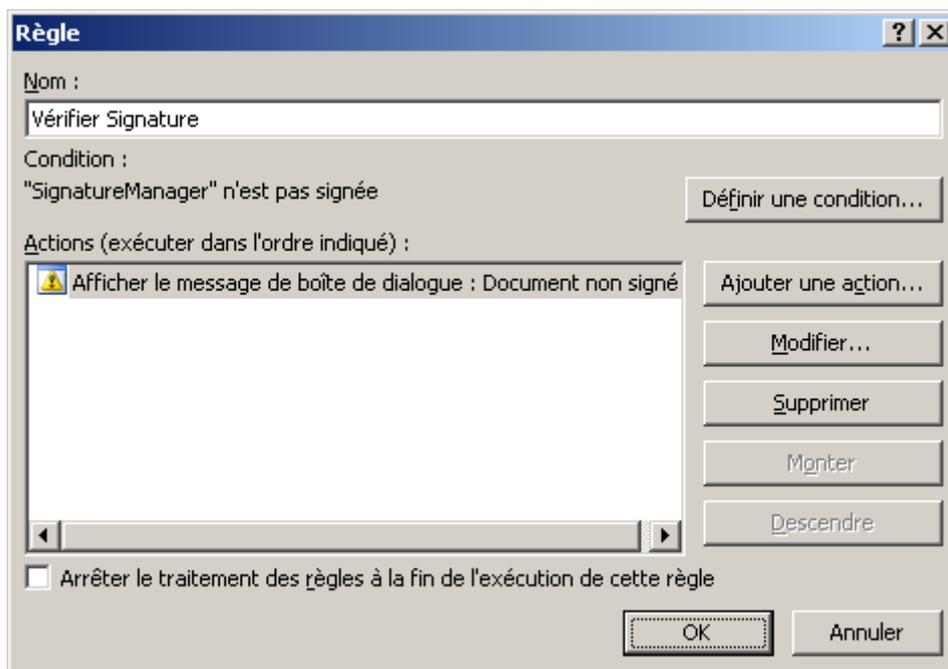
ensuite choisissez le critère *n'est pas signée* et validez par *OK* pour revenir dans la boîte de règles:



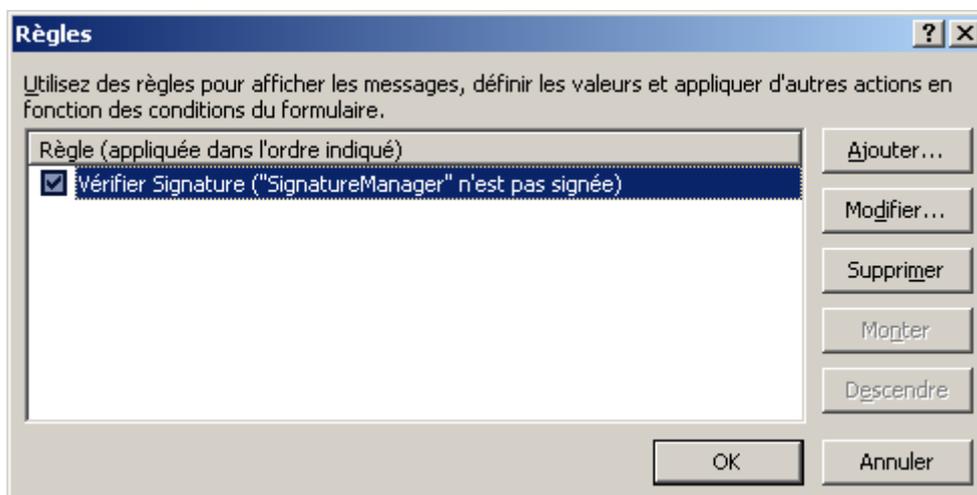
Cliquez sur *Ajouter une action...*:



et prenez par exemple *Afficher un message de boîte de dialogue* et validez par *OK*:

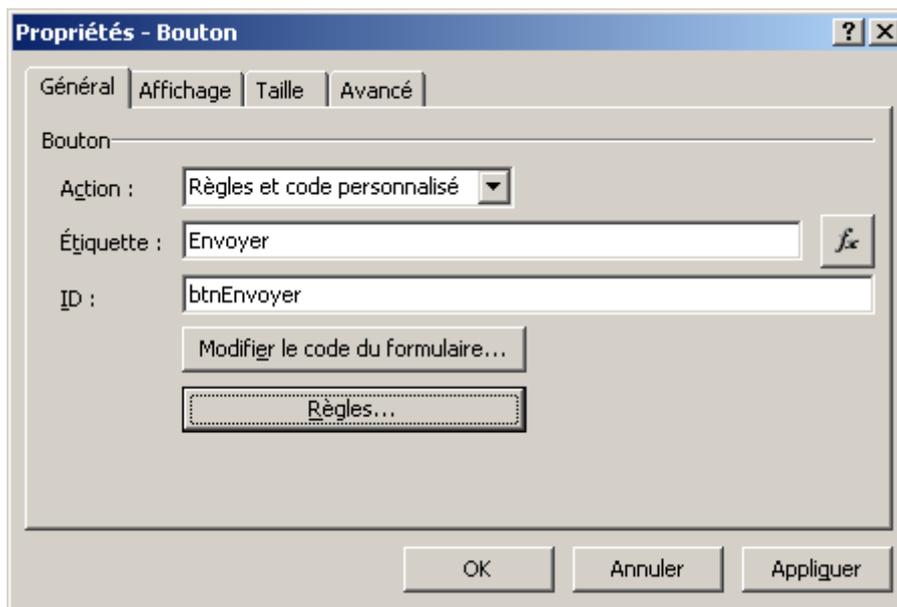


Validez par *OK*:



Vincent ISOZ

et une dernière fois par *OK*:



et maintenant, InfoPath vérifiera si le document est signé ou non.

## 9.2.7 Export d'un formulaire vers MS Word

Il est possible de manière relativement simple d'exporter le contenu et le contenant d'un formulaire InfoPath sans passer par du WordML.

Considérons un morceau de notre formulaire ci-dessous:

La note de frais se remplit uniquement du 1er au dernier jour du mois.  
Les justificatifs doivent être joints à la note de frais.

FORMULAIRE DE NOTES DE FRAIS			
Nom Prénom:		Nom Prénom du responsable:	
Nicolas Mayer		Weber Olivier	
Mois:	Mars	Année:	2006
Date d'envoi: 23.05.2006			

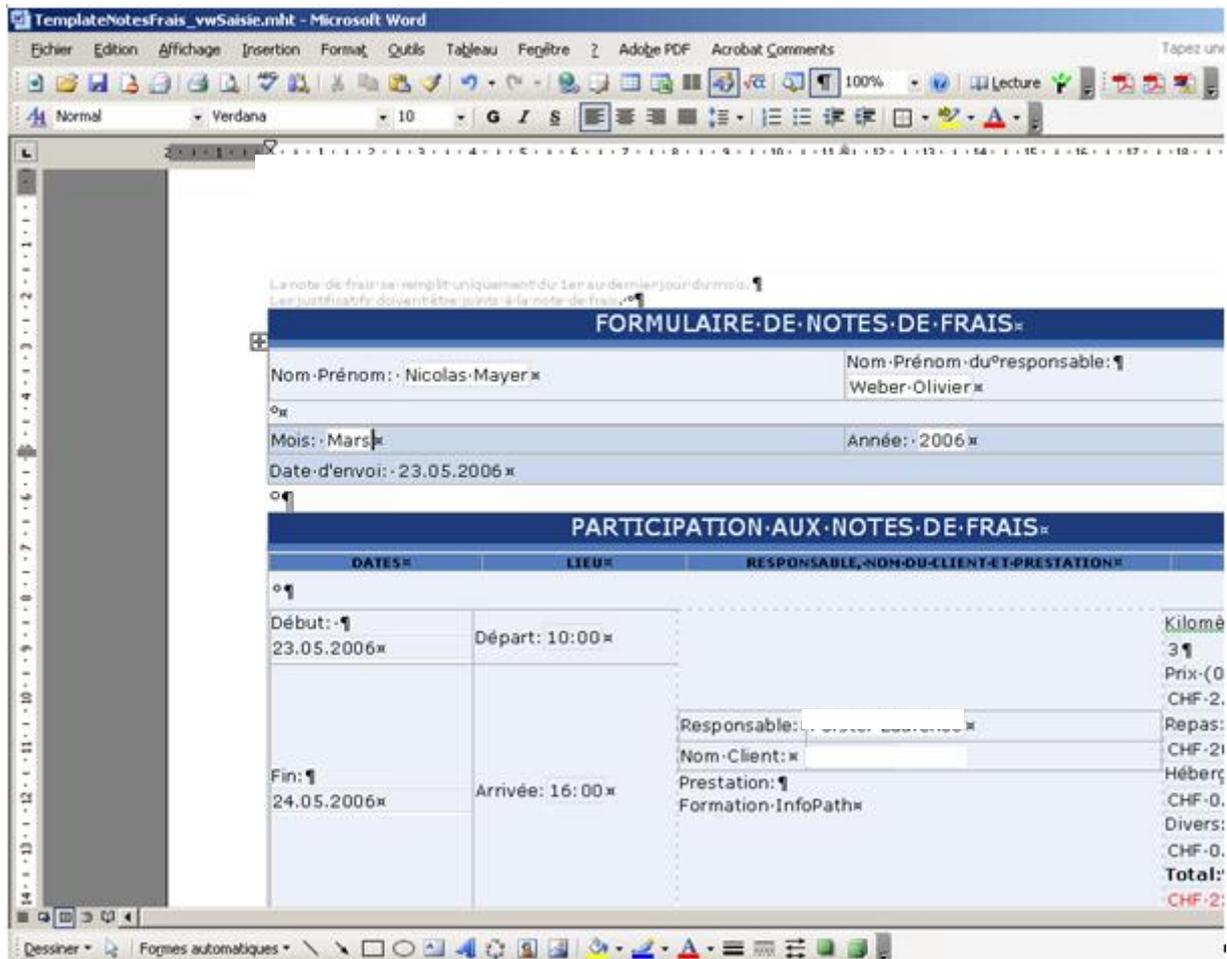
PARTICIPATION AUX NOTES DE FRAIS			
DATES	LIEU	RESPONSABLE, NOM DU CLIENT ET PRESTATION	FRAIS
Début:	Départ:	Responsable:	Kilomètres:
23.05.2006	10:00	Forster Laurence	3
		Nom Client:	Prix (0.7.-/Km)
		Infolearn	CHF 2.10
		Prestation:	Repas:
		Formation InfoPath	CHF 20.00
			Hébergement:
			CHF 0.00
			Divers:
			CHF 0.00
			<b>Total:</b>
			CHF 22.10

Insert item

Vincent ISOZ

et allons dans le menu *Fichier/Exporter vers/Web...* enregistrez votre fichier MHT à l'endroit voulu et ensuite ouvrez le fichier dans MS Word.

Le résultat est loin d'être satisfaisant mais c'est un début...



Cet exemple montre quelque chose d'important que ce soit pour l'impression ou l'export dans MS Word: **il vaut mieux faire des formulaires en hauteur qu'en largeur!!!**

### 9.2.8 Sélecteur de contact (Custom control)

L'idée de base des contrôles personnalisés ou *Custom controls* est que la toolbox d'InfoPath n'est plus limitée aux contrôles standard InfoPath, elle est extensible, tout comme la boîte à outils de Visual Studio !

Pour illustrer cette fonctionnalité, amusons-nous à concevoir un formulaire dans lequel un utilisateur pourra sélectionner une liste d'utilisateur provenant d'un site SharePoint (**car à ce jour ce contrôle ne fonctionne qu'avec la liste d'utilisateur SharePoint**).

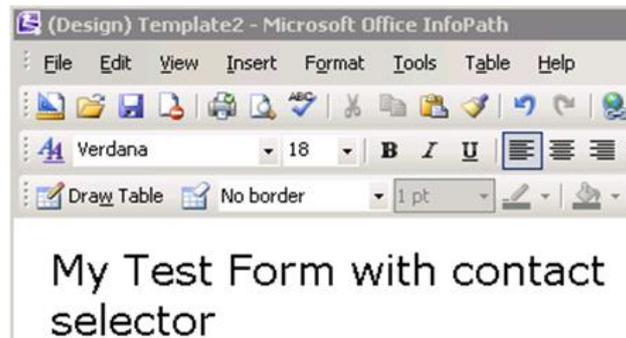
**Attention!!!**

1. Le texte qui suit est devenu inutile depuis InfoPath 2010 car Microsoft a infiniment simplifié la procédure.

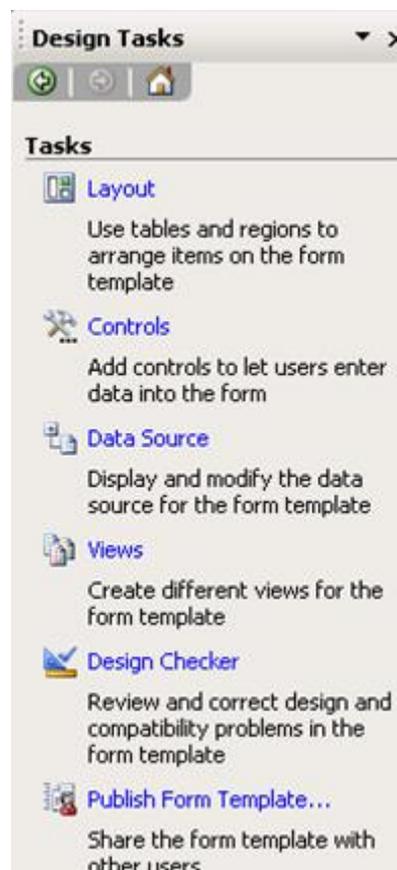
Vincent ISOZ

2. Ce contrôle ne s'imprime pas!!! Il faudra jouer avec un champ et une formulaire dans une autre vue spécialisée pour l'impression.

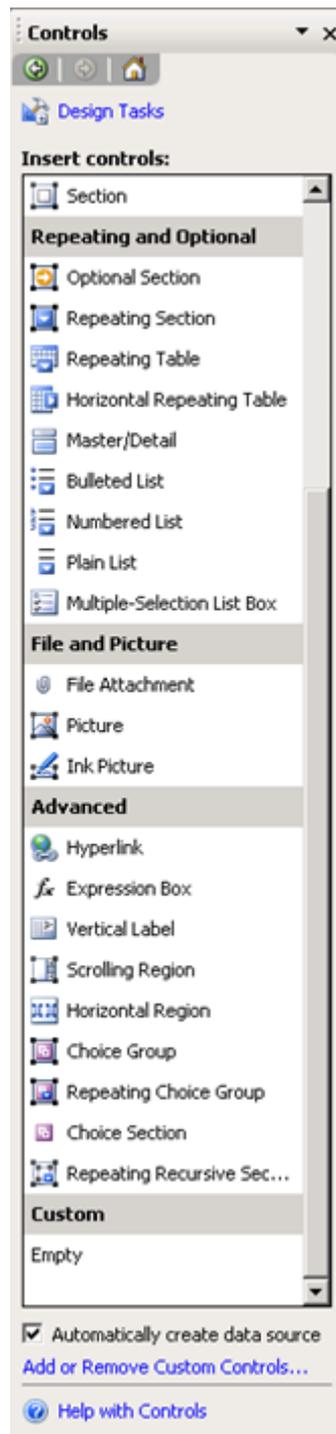
Nous créons donc un formulaire vierge:



Nous allons ensuite dans le panneau des tâches courantes:



Cliquez sur le lien *Controls*. Vous aurez alors comme à l'habitude:



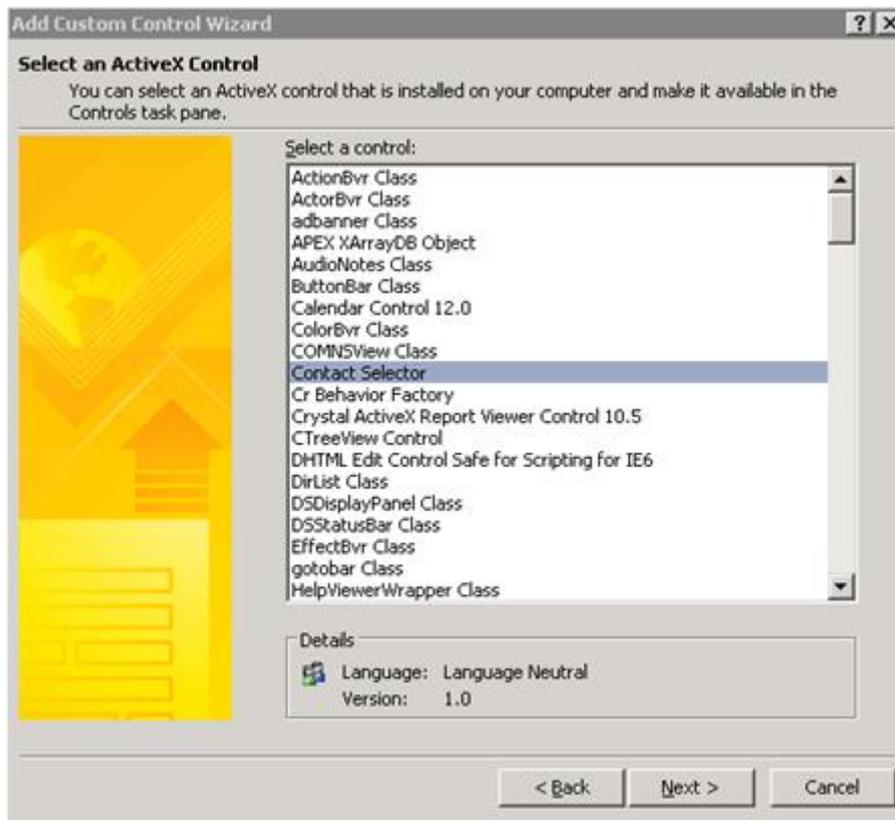
Cliquez sur le lien *Add or Remove Custom Controls...* se trouvant tout en bas. Vous aurez alors:



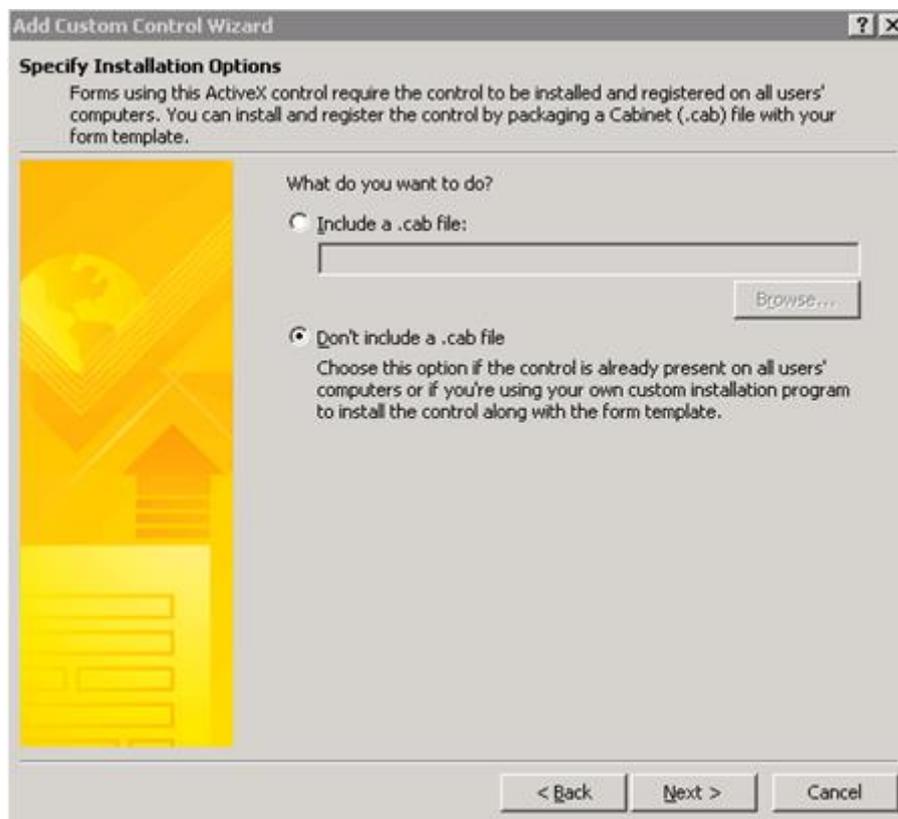
Cliquez sur *Add...*:



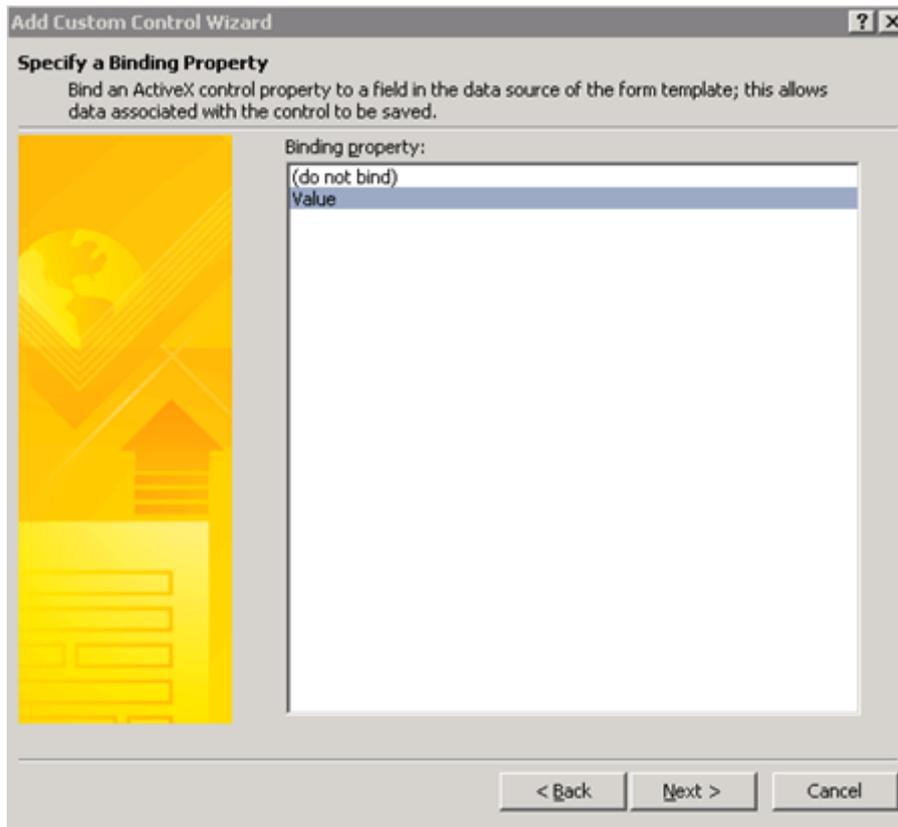
et choisissez *ActiveX Control* et cliquez sur *Next*:



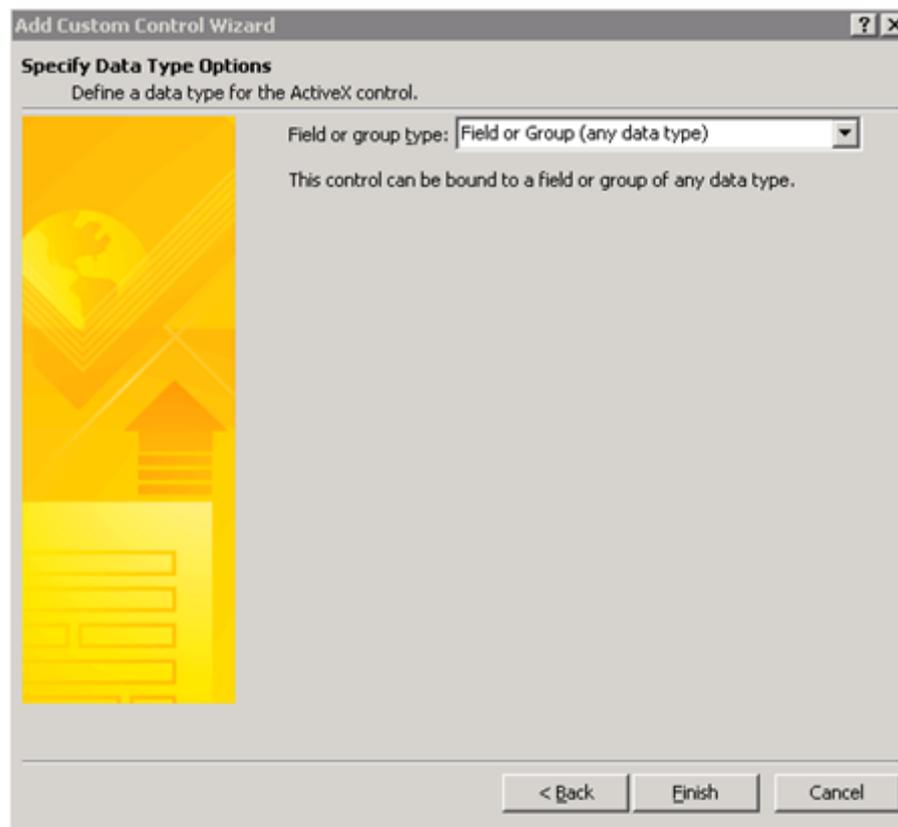
Sélectionnez *Contact Selector* et cliquez sur *Next*:



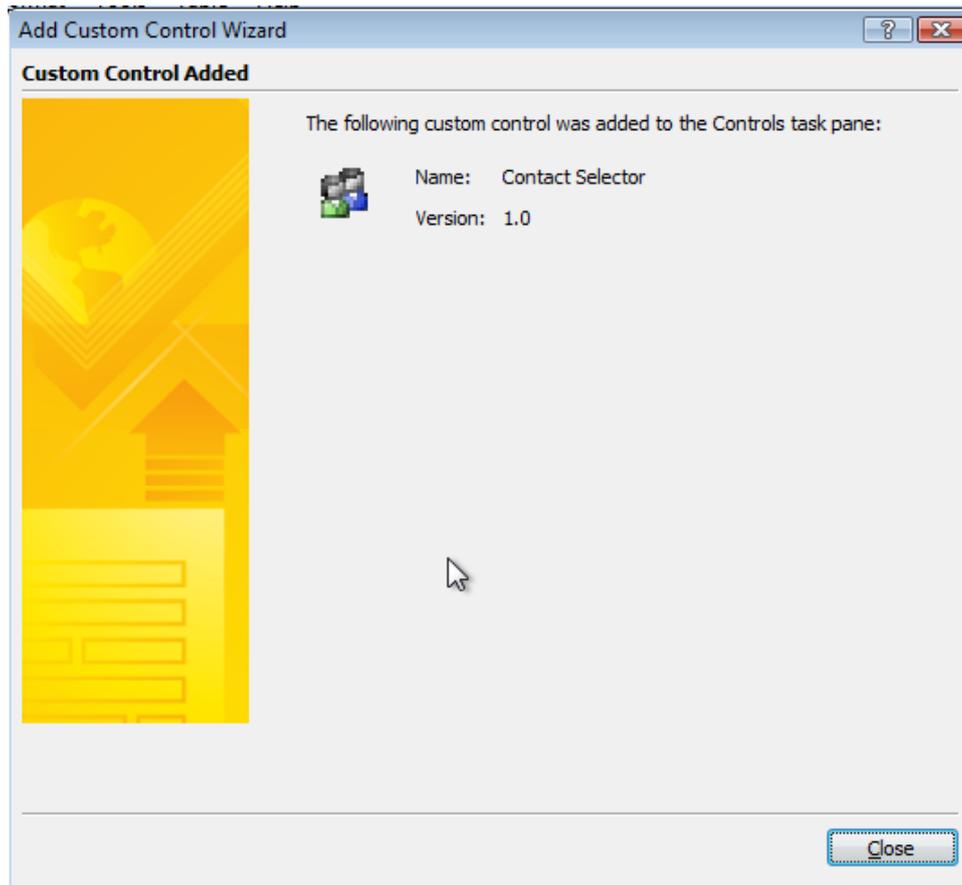
Cliquez sur *Don't include a .cab file*:



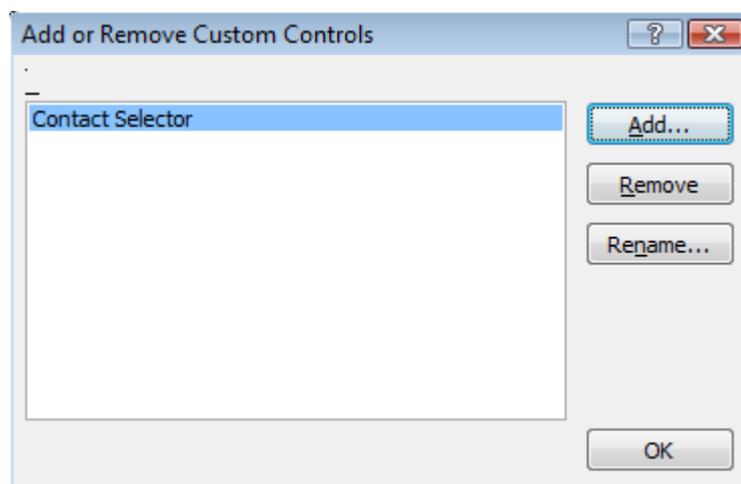
Nous sélectionnons *Value* et cliquons sur *Next*:



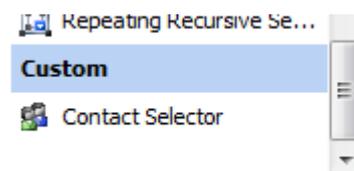
Nous sélectionnons *Field or Group (any data type)* et cliquons sur *Finish*:



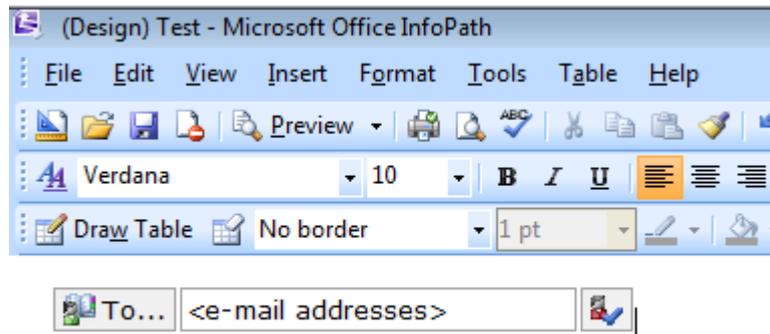
et nous cliquons sur *Close*:



et *OK*. Nous avons alors dans le volet des contrôles:



Insérez-le dans le formulaire:



Ensuite, il faut créer le schéma XML qu'InfoPath 2003 et 2007 ne crée pas tout seul:

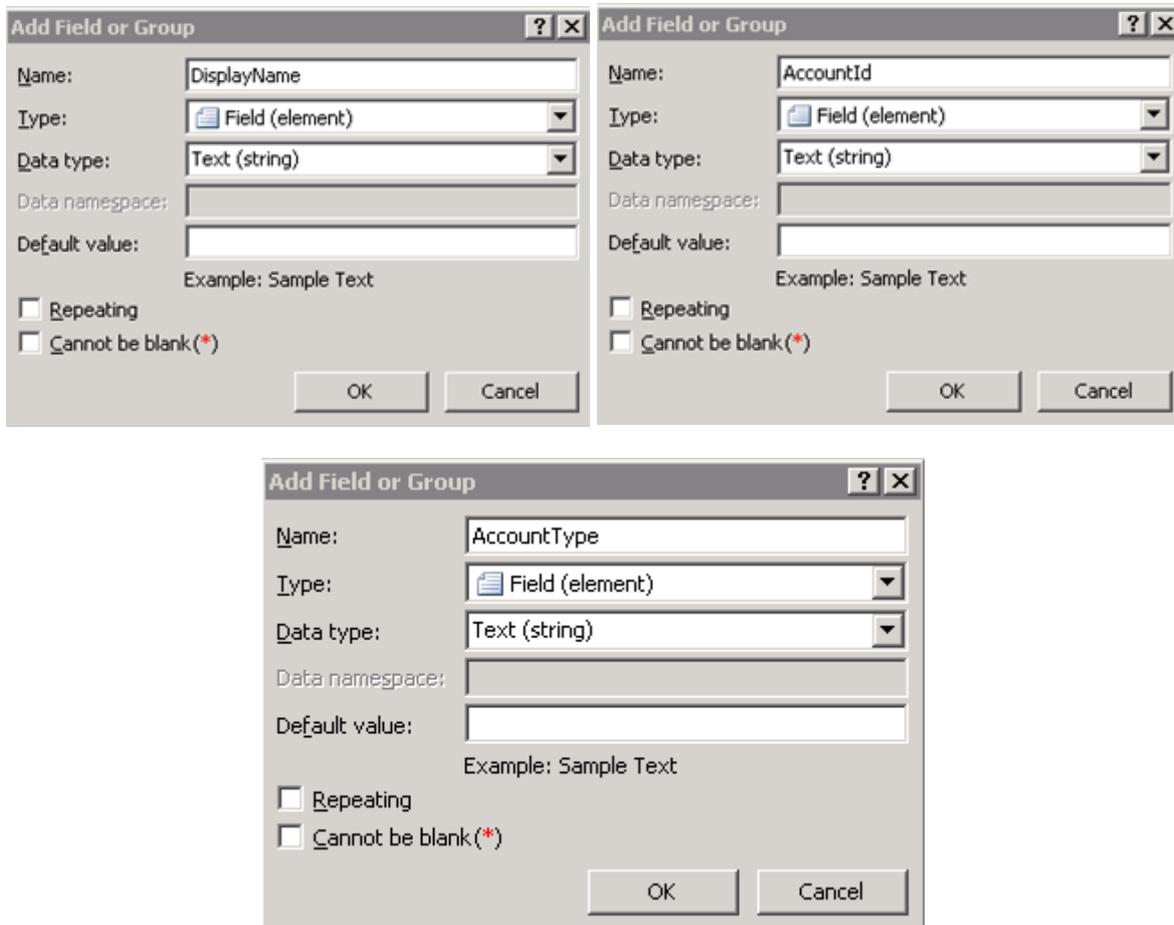


Donc d'abord, créez sous *group1* un groupe répétitif que l'option *Repeating* et renommez le *Person*:

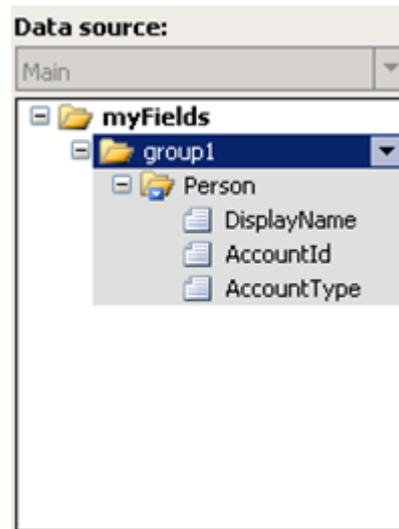
Vincent ISOZ



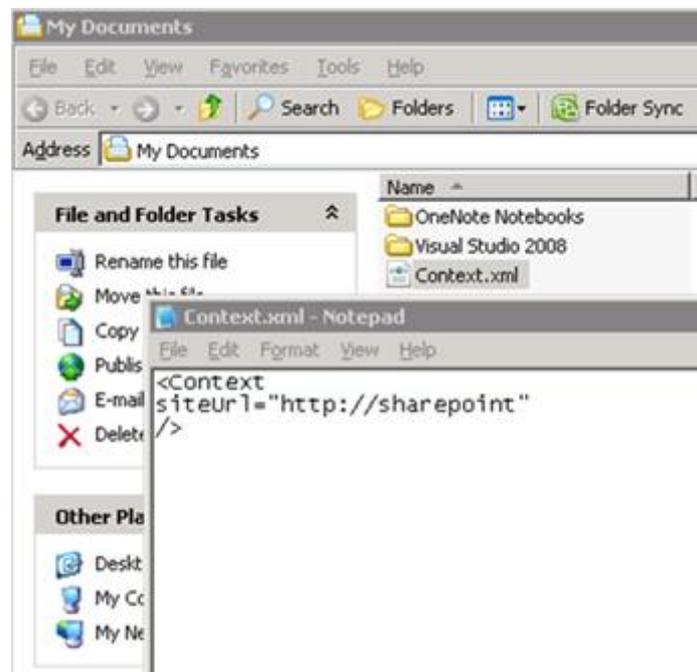
Ensuite, créez dans le groupe les trois champs suivants:



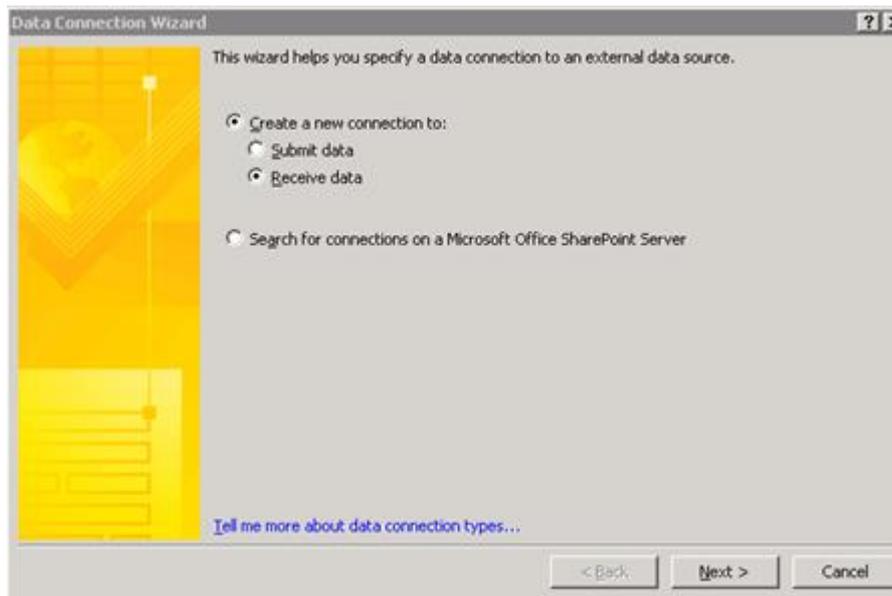
ce qui donnera:



Ensuite, créez le fichier *Context.xml* suivant:



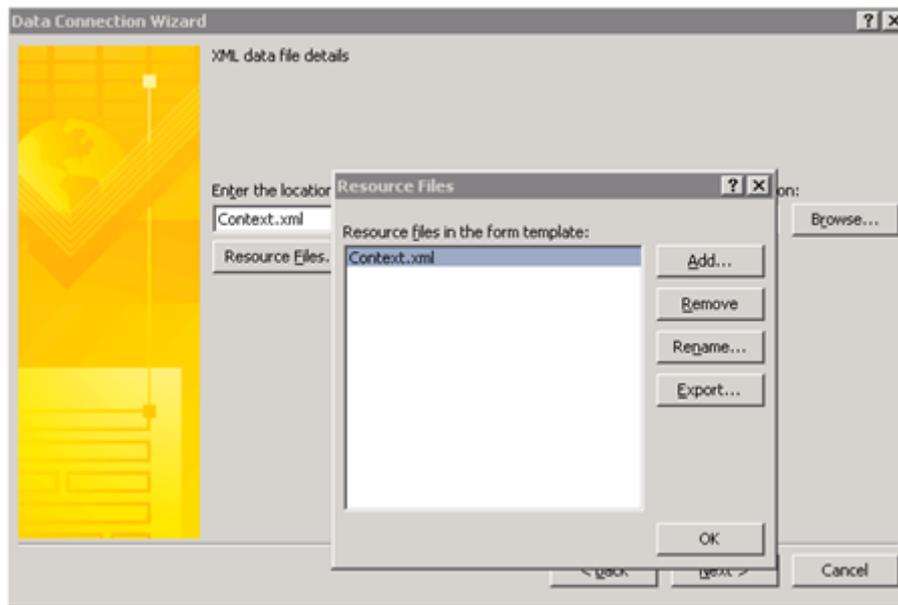
Ensuite, dans votre formulaire, lancez l'assistant de connexion (*Tools/Data connection et Add...*):



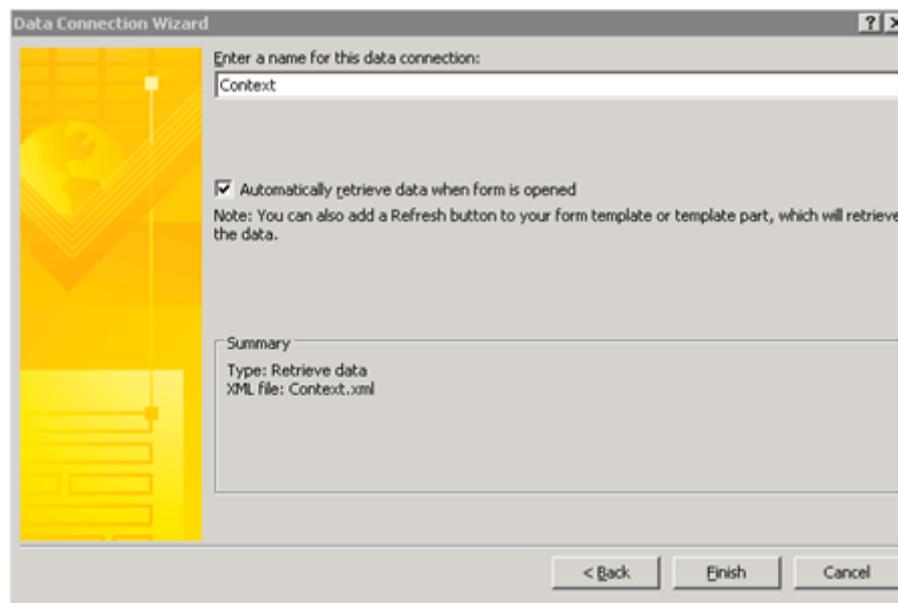
Sélectionnez *Receive data* et cliquez sur *Next*:



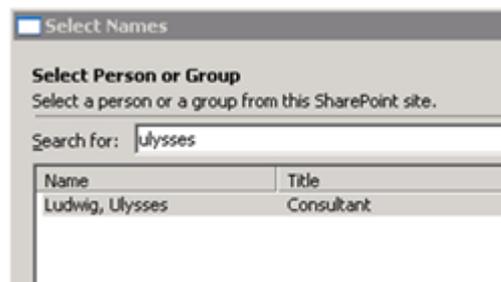
et sélectionnez *XML document*. Cliquez sur *Next*:



Cliquez sur *Resource Files...* et allez chercher le fichier *Context.xml*. et cliquez sur *Next*:

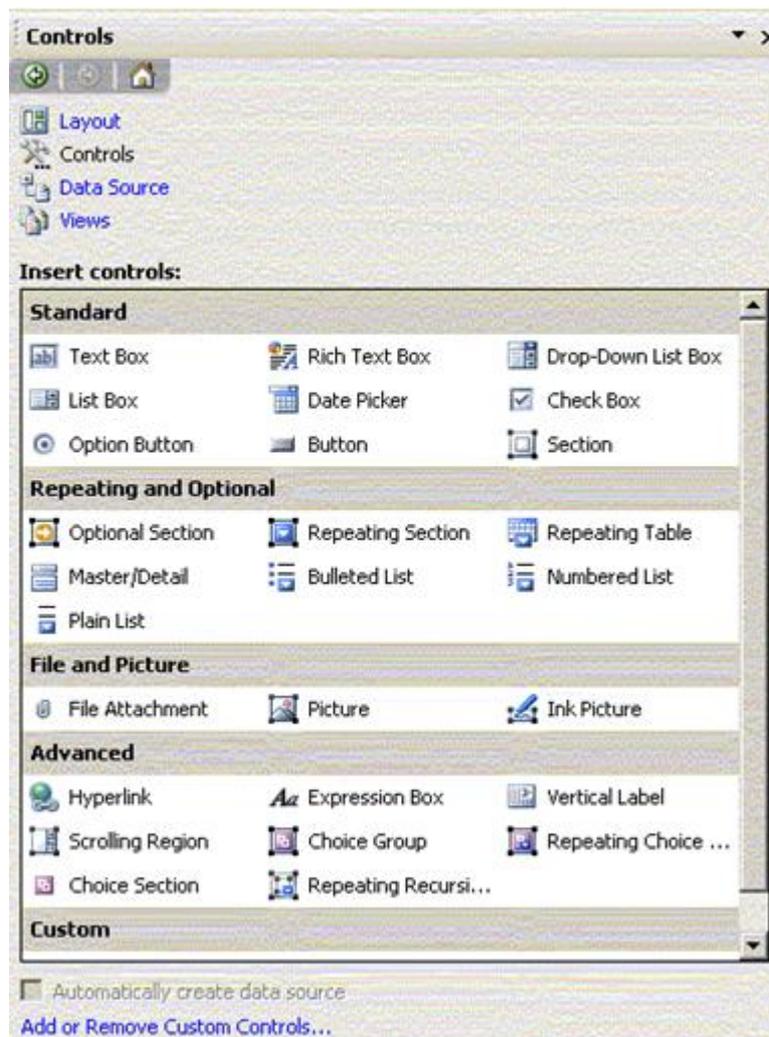


et sur *Finish*. Et lancez l'aperçu du formulaire pour test votre champ:

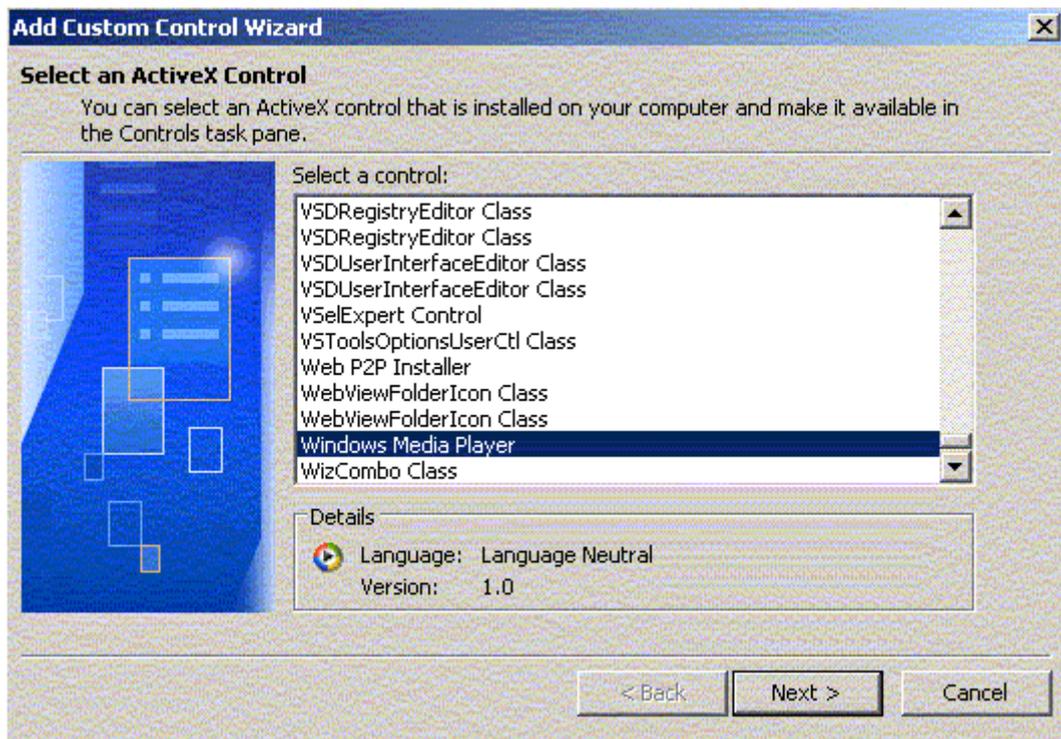


## 9.2.9 Media Player (Custom control)

Nous allons utiliser ici Windows Media Player comme custom control dans InfoPath. La première étape est d'afficher la toolbox et de sélectionner *Add or remove custom controls ...*:



Un assistant apparaît, demandant de sélectionner un contrôle ActiveX: Sélectionnons le Windows Media Player:

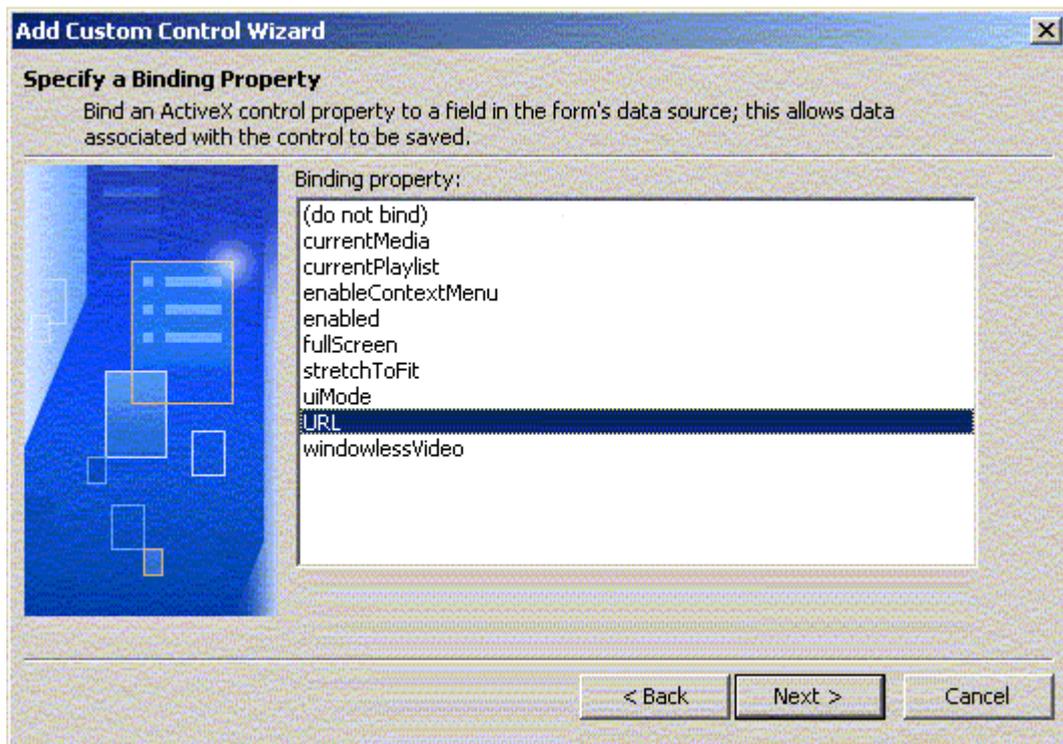


L'écran suivant de l'assistant concerne le déploiement: si un formulaire référence un composant ActiveX, alors il faut s'assurer que ce composant est présent sur tous les postes sur lesquels le formulaire sera redistribué. Voici pourquoi l'assistant offre la possibilité d'inclure un fichier CAB d'installation de l'ActiveX. Une fois cette option sélectionnée, le CAB sera inclus dans tout modèle de formulaire utilisant le contrôle personnalisé.

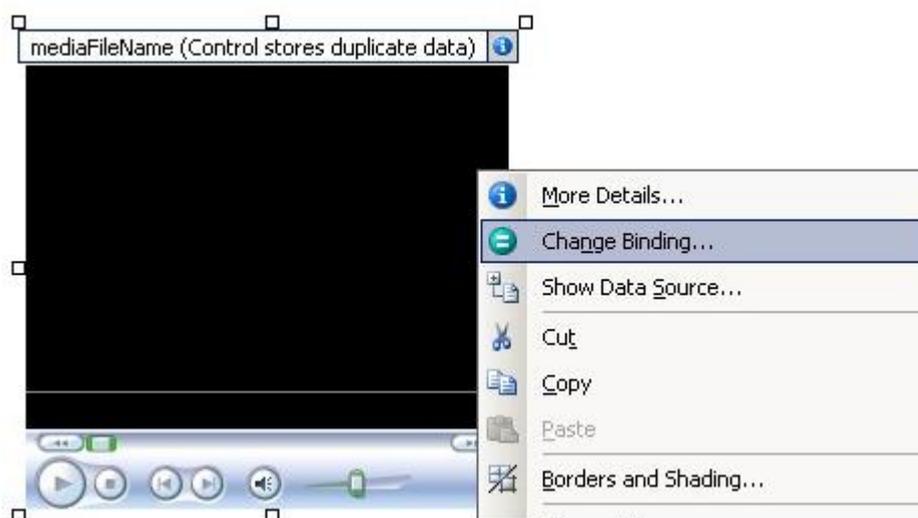


Dans le présent exemple, nous utilisons le Windows Media Player dont il est de notoriété publique qu'il est présent sur chaque poste Windows, nous pouvons donc nous permettre de

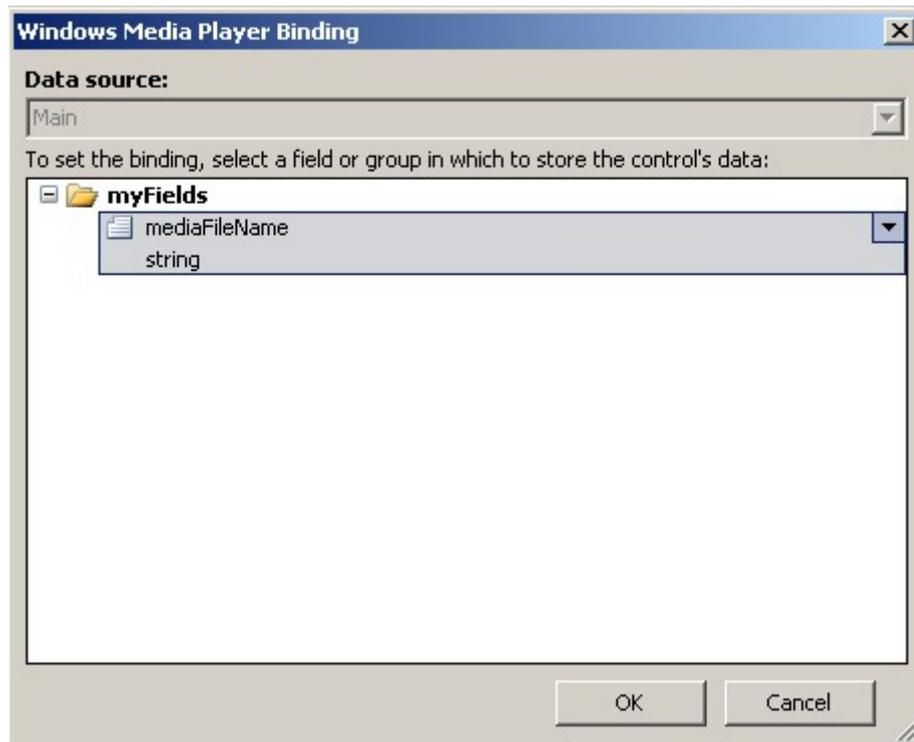
passer sur cet écran. Le suivant liste toutes les propriétés publiques exposées par le composant sélectionné, et me permet d'en choisir une qui pourra être liée à un champ du formulaire. Ce que nous voulons ici est que WMP affiche une vidéo sur la base d'un nom de fichier saisi ailleurs dans le formulaire ; la propriété qui nous intéresse est donc *URL*:



Nous voilà prêts à utiliser ce contrôle WMP dans le formulaire: il apparaît dans la toolbox donc nous n'avons plus qu'à le glisser-déposer sur la surface du concepteur de formulaires. Ceci étant fait, il ne nous reste plus qu'à attacher le contrôle à la valeur saisie dans une text box positionnée ailleurs sur le formulaire. Pour cela, utilisons l'entrée *Change Binding ...* du menu contextuel:



InfoPath demande alors de sélectionner le champ de données auquel WMP doit être lié (par le truchement de sa propriété URL). Ici nous spécifions *mediaFileName* (lui-même alimenté par une text box) comme l'élément source qui alimente le Media Player:



Après sauvegarde et prévisualisation, sitôt que nous saisissons un chemin valide vers un fichier média dans la boîte de texte, Windows Media Player le joue dès que la textbox perd le focus:



Vincent ISOZ

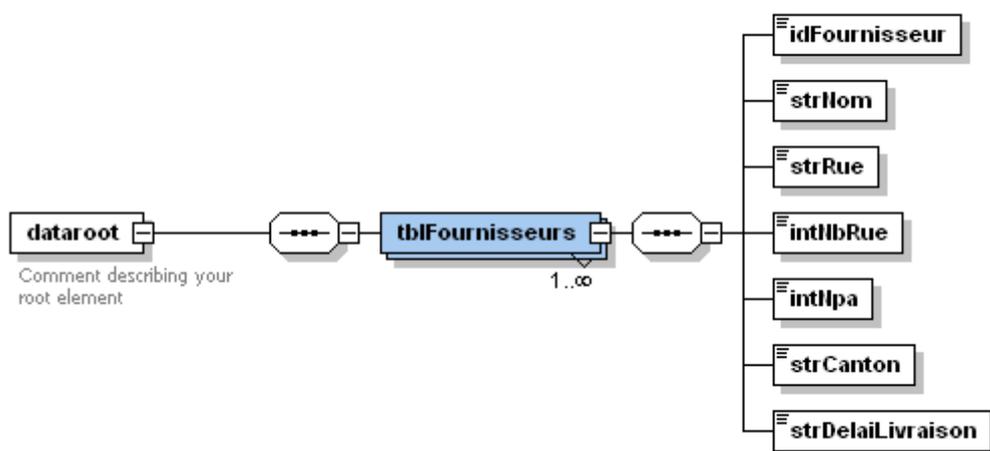
Cette fonctionnalité particulièrement séduisante signifie qu'InfoPath prend de plus en plus les traits d'un environnement de développement de formulaires.

### 9.3 Formulaire XSD

Nous allons créer un formulaire InfoPath qui permettra d'ajouter à notre base MS Access manuellement de nouveaux fournisseurs (nous faisons donc référence à la base de données utilisée dans mon livre sur MS Access).

Le but de ce document n'état pas d'apprendre MS InfoPath mais de voir le lien qu'il y a entre lui et le XML, je ne m'étendrai pas sur la manière de créer le formulaire en soi (c'est accessible même a quelqu'un qui débute en informatique).

Mais avant de créer le formulaire, il nous faut le validateur XSD de la table *tblFournisseurs* de notre base \*.mdb (voir mon livre sur le XML pour apprendre à construire des fichiers XSD avec XMLSpy). Rappelons qu'il est le suivant:



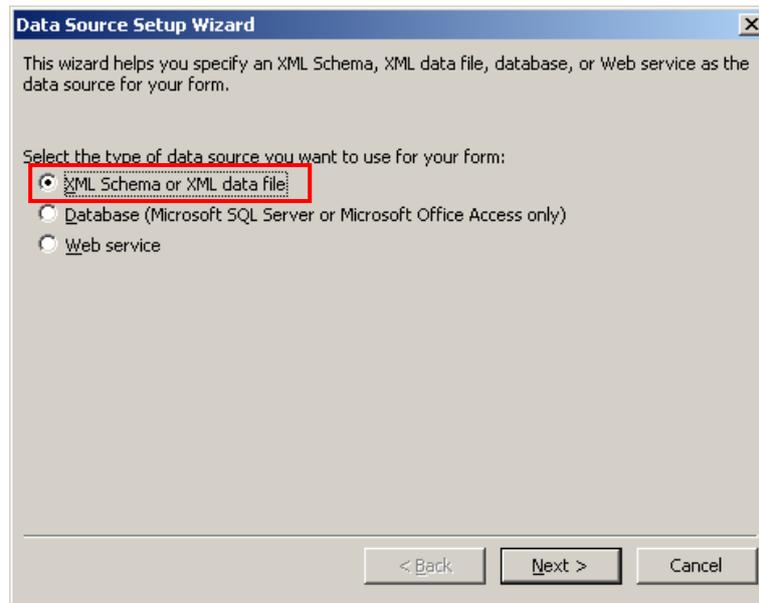
Maintenant, quand vous ouvrez MS InfoPath, dans le volet Office choisissez:



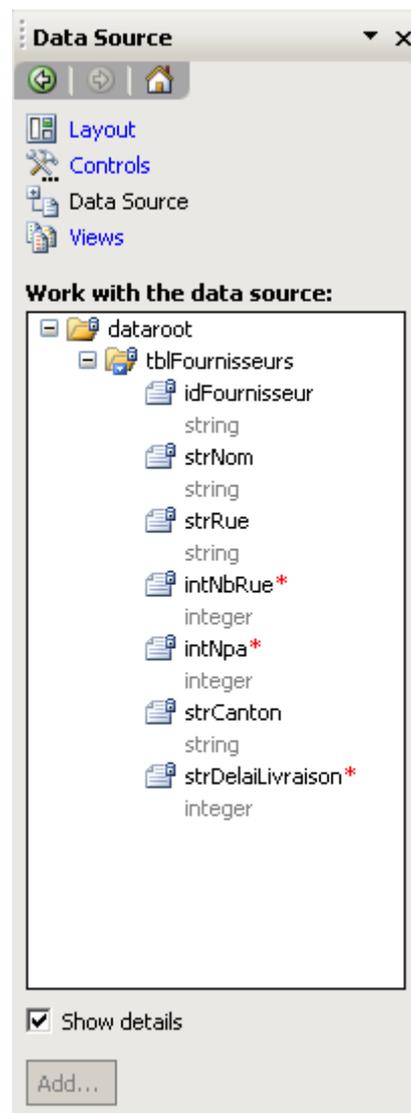
et ensuite toujours dans le volet Office:



Dans la fenêtre qui apparaît sélectionnez:



et allez chercher le fichier XSD. Une fois ceci fait vous devriez avoir dans le volet Office:



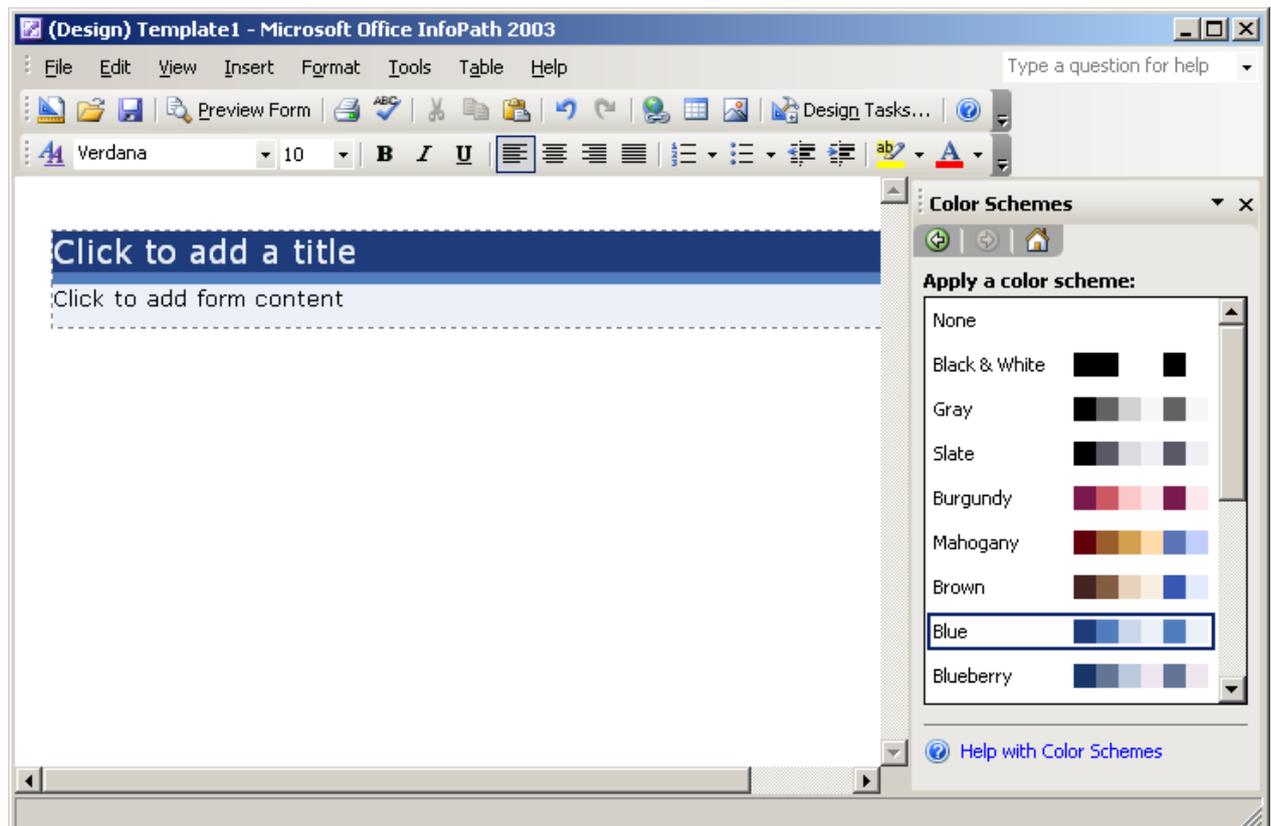
Vincent ISOZ

Il s'agit bien de la structure de notre XSD.

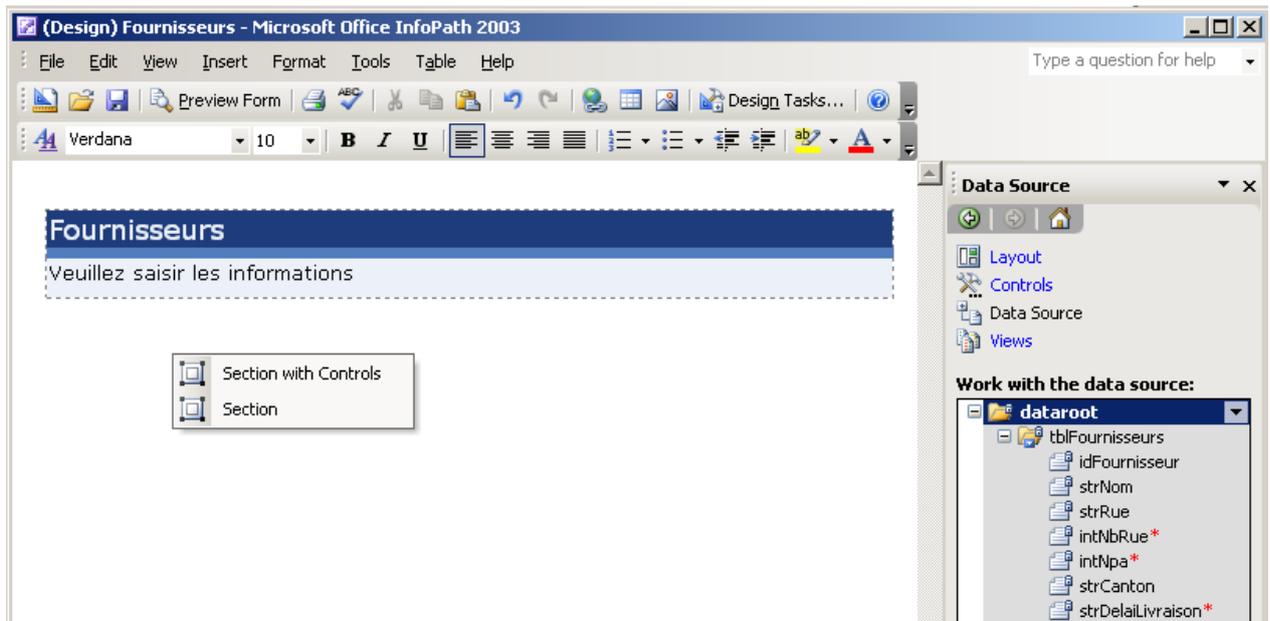
Ensuite dans le *Layout* du formulaire choisissez *Table with Title*:



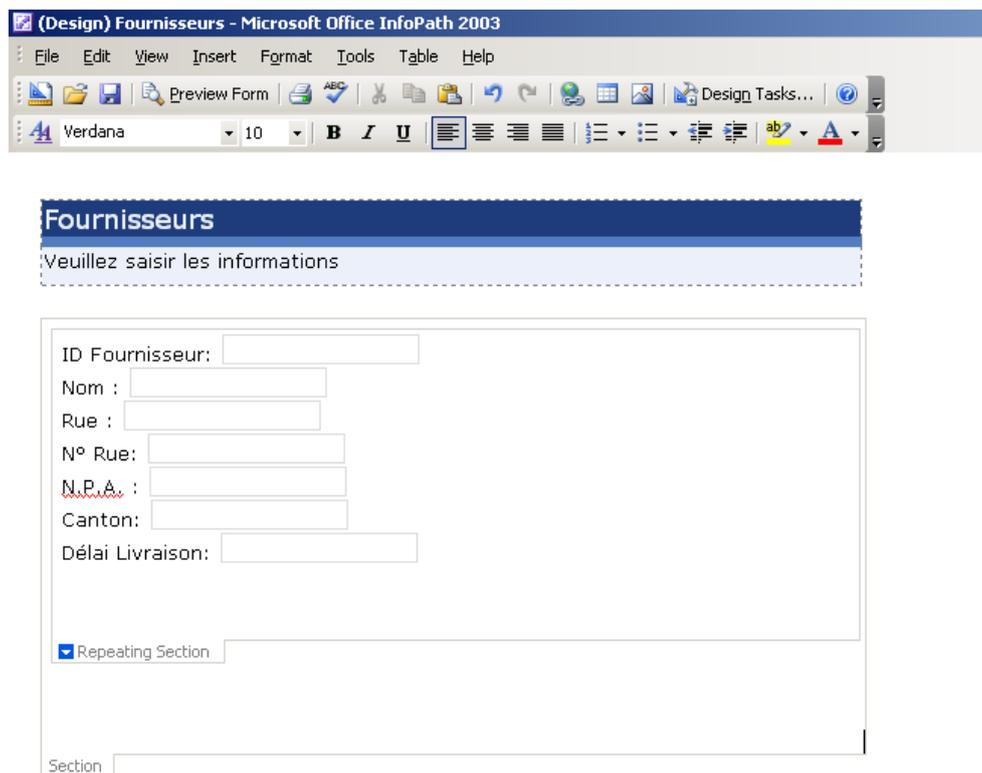
et ensuite, choisissons un joli design:



Ensuite retournez dans *Data Source* et faites un glisser-déplacer du nœud *dataroot* dans la feuille en sélectionnant *Section with controls*:



et voilà le résultat (après avoir un peu changé les texte des libellés):



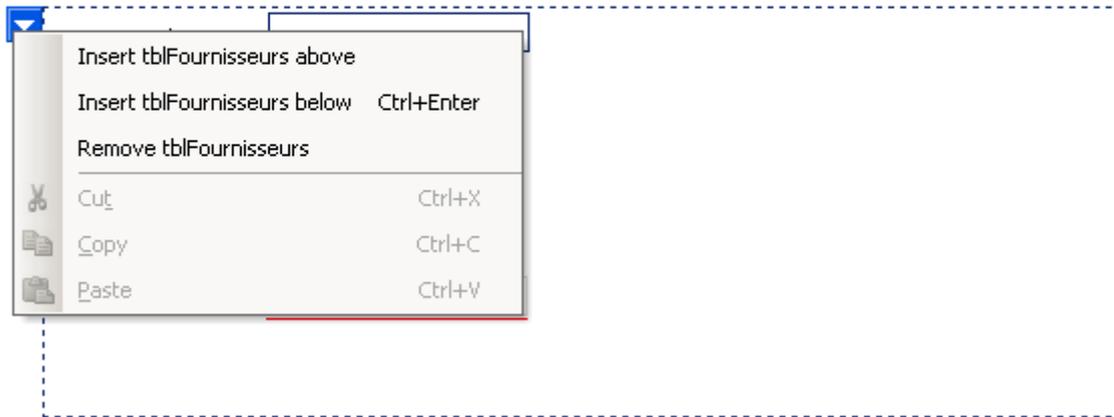
Enregistrez les modifications et fermez InfoPath. Faites ensuite un double clic sur le fichier InfoPath:



et saisissez les données. Au besoin, en faisant un clic sur la flèche bleue, vous pouvez ajouter des nouveaux enregistrements (c'est ce que fait d'InfoPath un outil très puissant):

## Fournisseurs

Veillez saisir les informations



et donc après avoir saisi quelques valeurs:

## Fournisseurs

Veillez saisir les informations

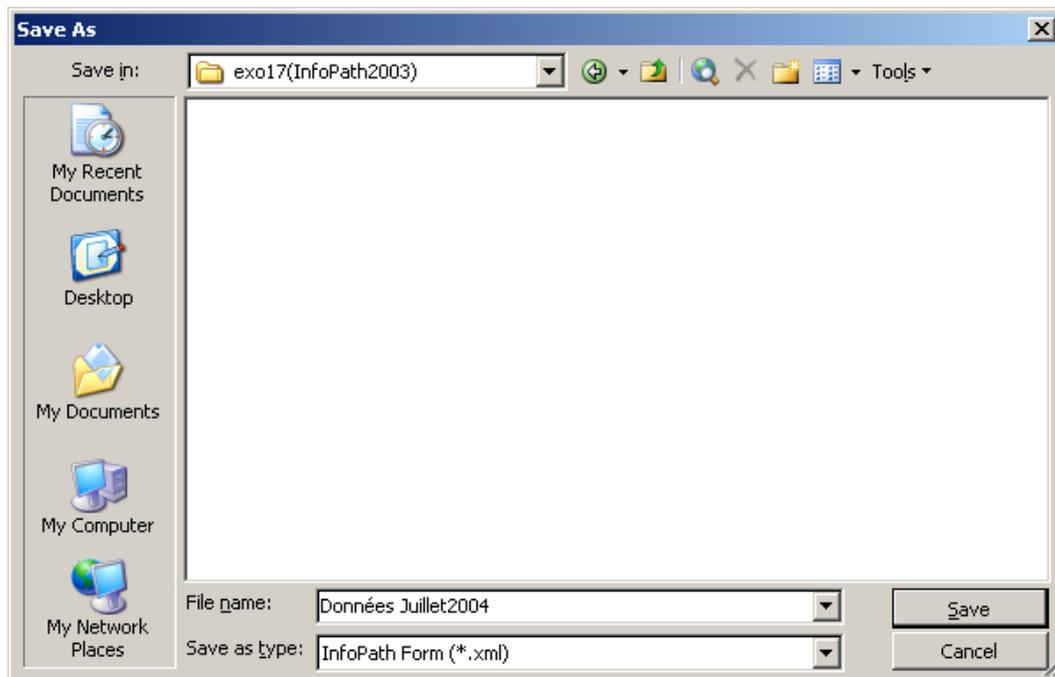
Nom :   
Rue :   
N° Rue:   
N.P.A. :   
Canton:   
Délai Livraison:

Nom :   
Rue :   
N° Rue:   
N.P.A. :   
Canton:   
Délai Livraison:

L'utilisateur du formulaire clique sur *Enregistrer*:



et InfoPath propose automatiquement de sauvegarder les données en XML:



et voici le fichier XML sortant:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?mso-infoPathSolution solutionVersion="1.0.0.3" productVersion="11.0.5531" PIVersion="1.0.0.0"
href="file:///C:\Documents%20and%20Settings\Isoz\Mes%20documents\Professionel\Cours/XML\exo17(InfoPath2003)\
Fournisseurs.xsn" language="fr" ?>
```

```
<?mso-application progid="InfoPath.Document"?>
```

```
<dataroot>
  <tblFournisseurs>
    <idFournisseur/>
    <strNom>TSR</strNom>
    <strRue>Ch. des Vernets</strRue>
    <intNbRue>8</intNbRue>
    <intNpa>1006</intNpa>
    <strCanton>Genève</strCanton>
    <strDelaiLivraison>30</strDelaiLivraison>
  </tblFournisseurs>
  <tblFournisseurs>
    <idFournisseur/>
    <strNom>SSR</strNom>
    <strRue>Burgstrasse</strRue>
    <intNbRue>20</intNbRue>
    <intNpa>8005</intNpa>
    <strCanton>Zürich</strCanton>
    <strDelaiLivraison>1</strDelaiLivraison>
  </tblFournisseurs>
</dataroot>
```

Vincent ISOZ

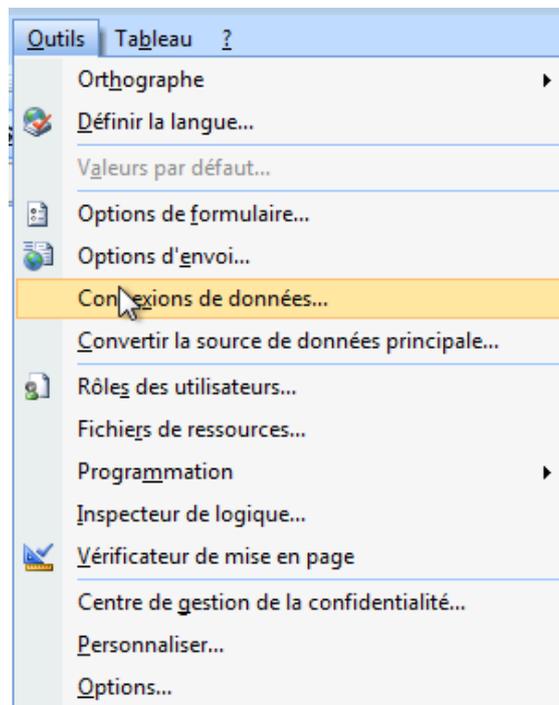
Tout est parfait (mis à part la ligne propriétaire de Microsoft...) pour un traitement par MS Excel 2003 et ultérieur, pour un formatage web par un XSL, pour une utilisation de pagination dans MS Word 2003 ou ultérieur ou pour un import dans MS Access 2003 ou ultérieur. Rien de nouveau donc de ce côté.

L'employé n'a qu'à envoyer ce document XML par e-mail à son responsable ou à des collègues.

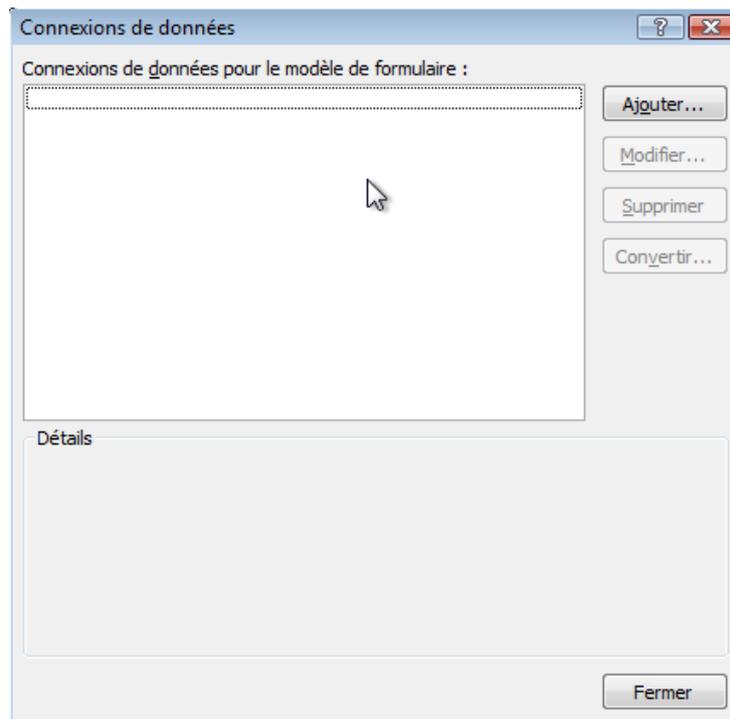
#### ***9.4 Ajout d'une liste avec les utilisateurs SharePoint***

Voici comment procéder pour avoir une liste déroulante automatique avec les utilisateurs d'un site SharePoint. Il existe un contrôle Person or Group Picker dans les contrôles ActiveX d'InfoPath 2003 et 2007 qui est standard maintenant dans 2010 mais son utilisation et sa configuration dans InfoPath 2003 et 2007 est un peu... nulle... et de plus ce champ ne marche pas avec InfoPath Services sans faire de multiples bricolages avec des WebParts. En attendant la version 2010 voici comment procéder.

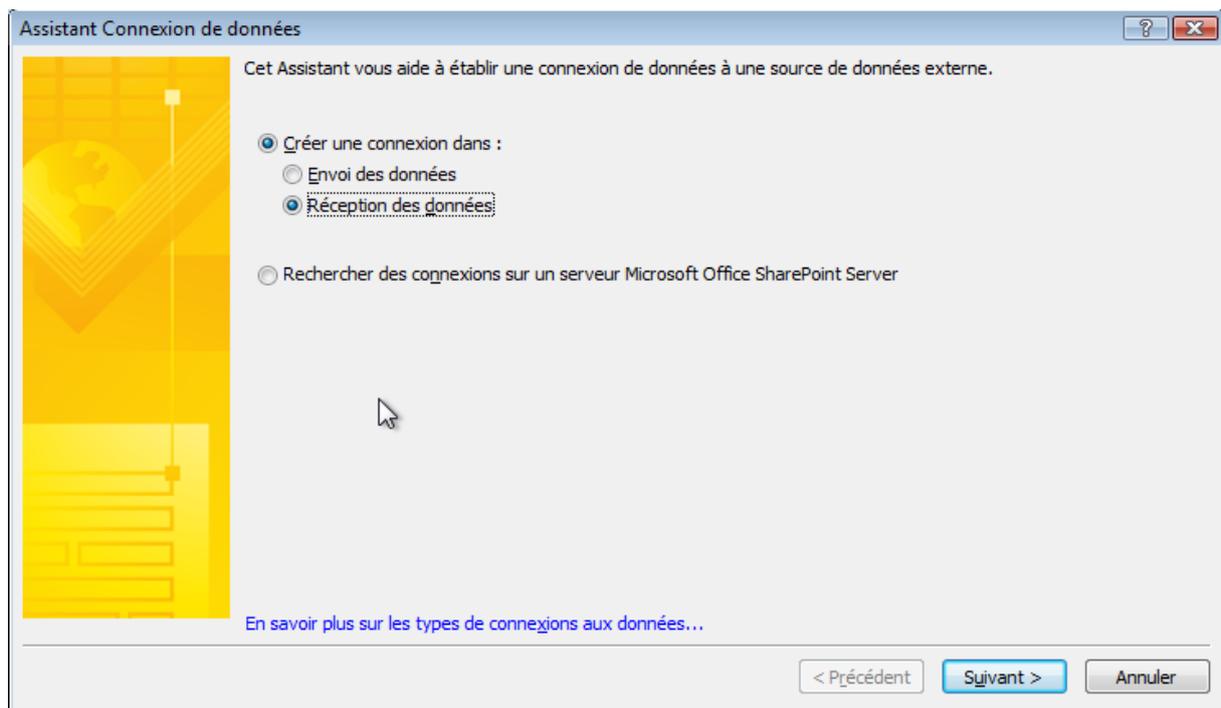
Allez dans *Outils/Connexion de données*:



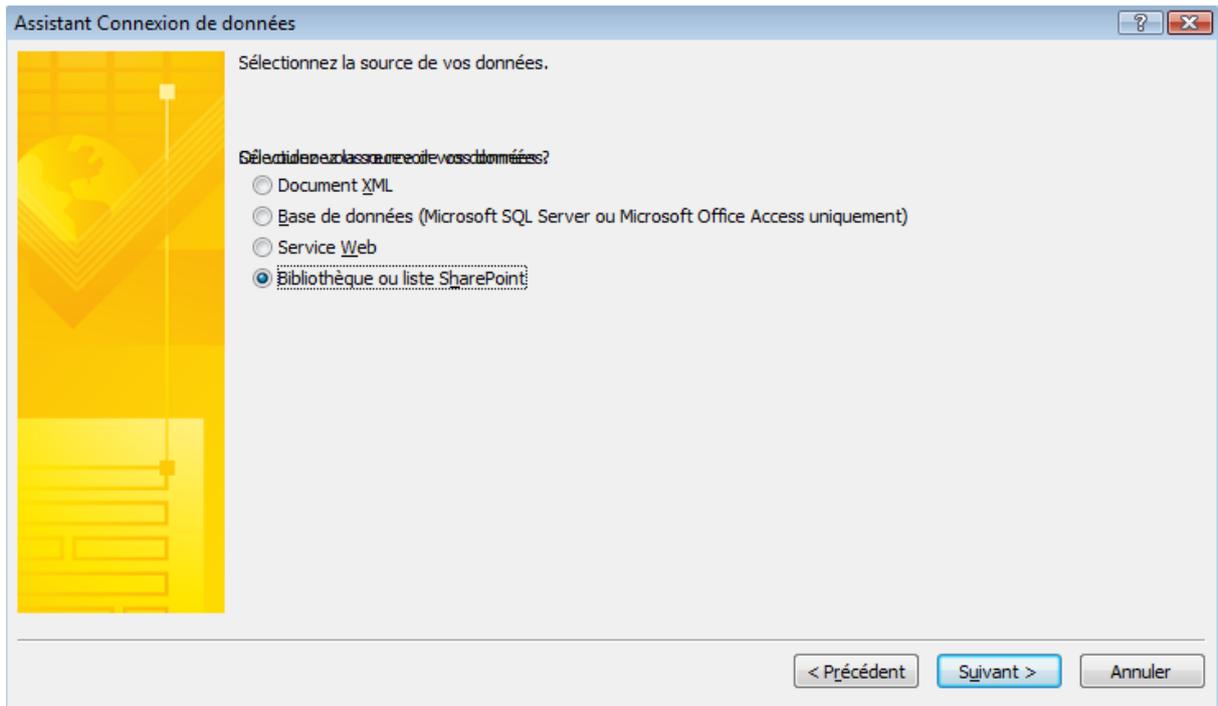
Ensuite cliquez sur *Ajouter*:



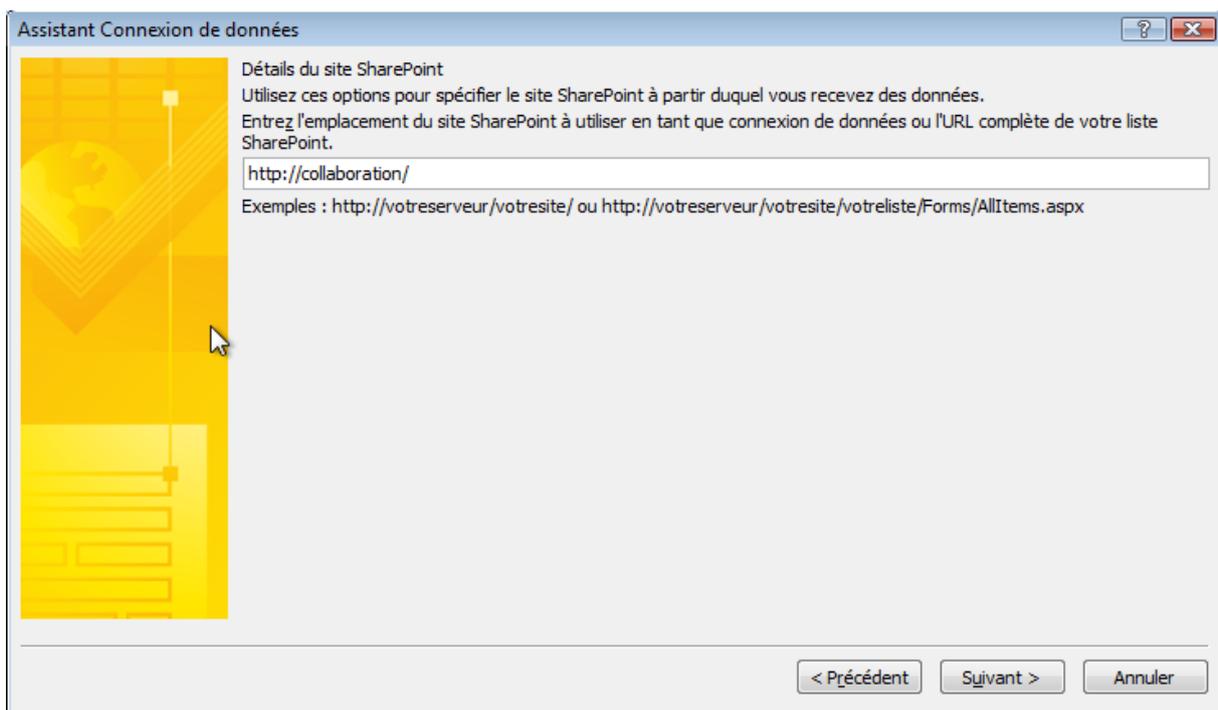
Prenez *Réception des données* et cliquez sur *Suivant*:



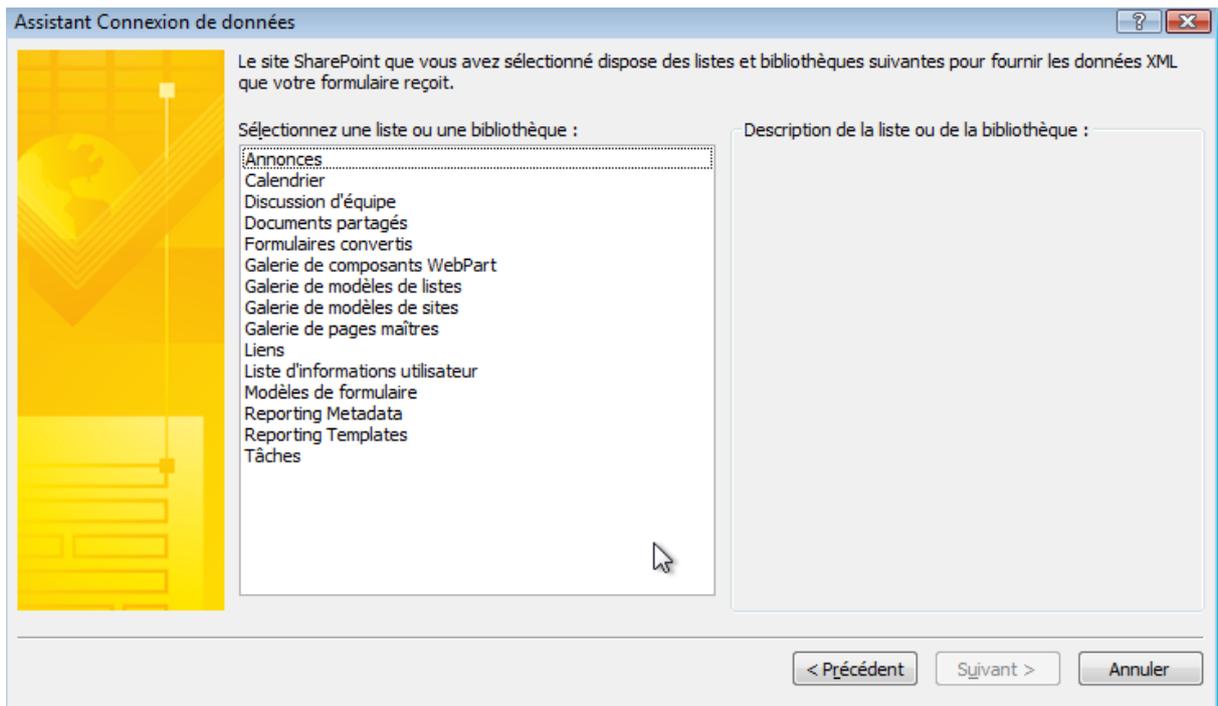
Prenez ensuite *Bibliothèque ou liste SharePoint* et cliquez sur *Suivant*:



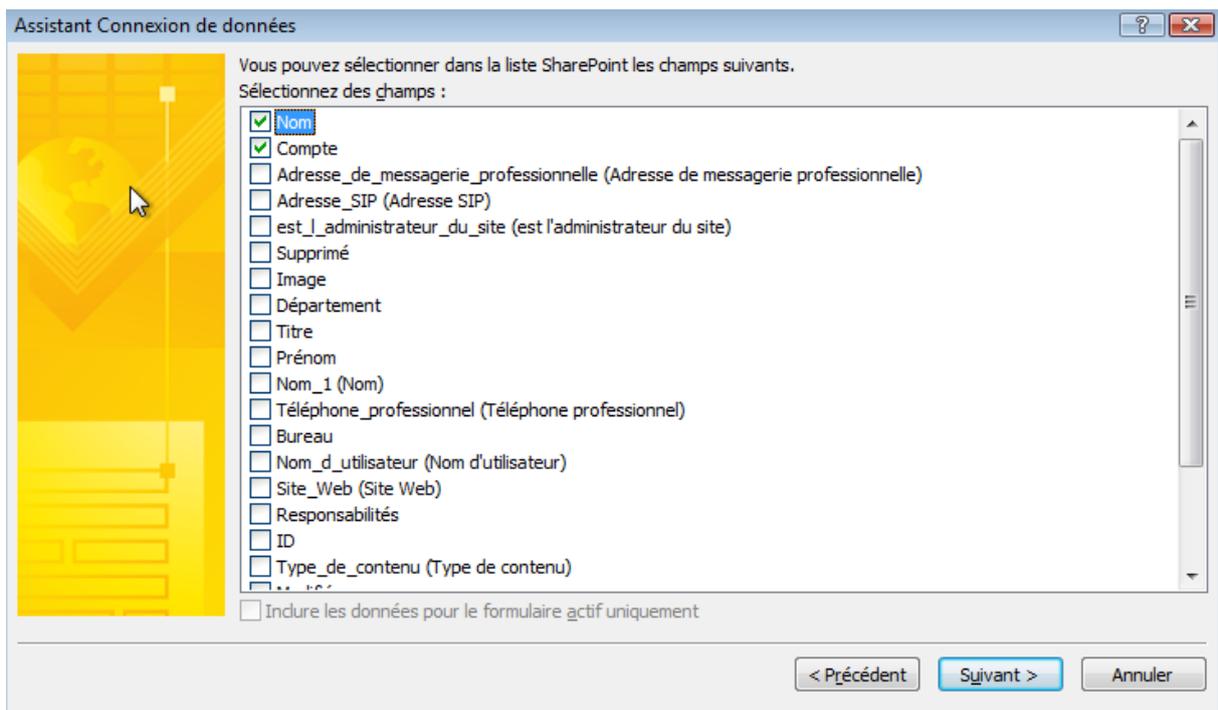
Mettez-y le lien de la racine de votre site SharePoint et cliquez sur *Suivant*:



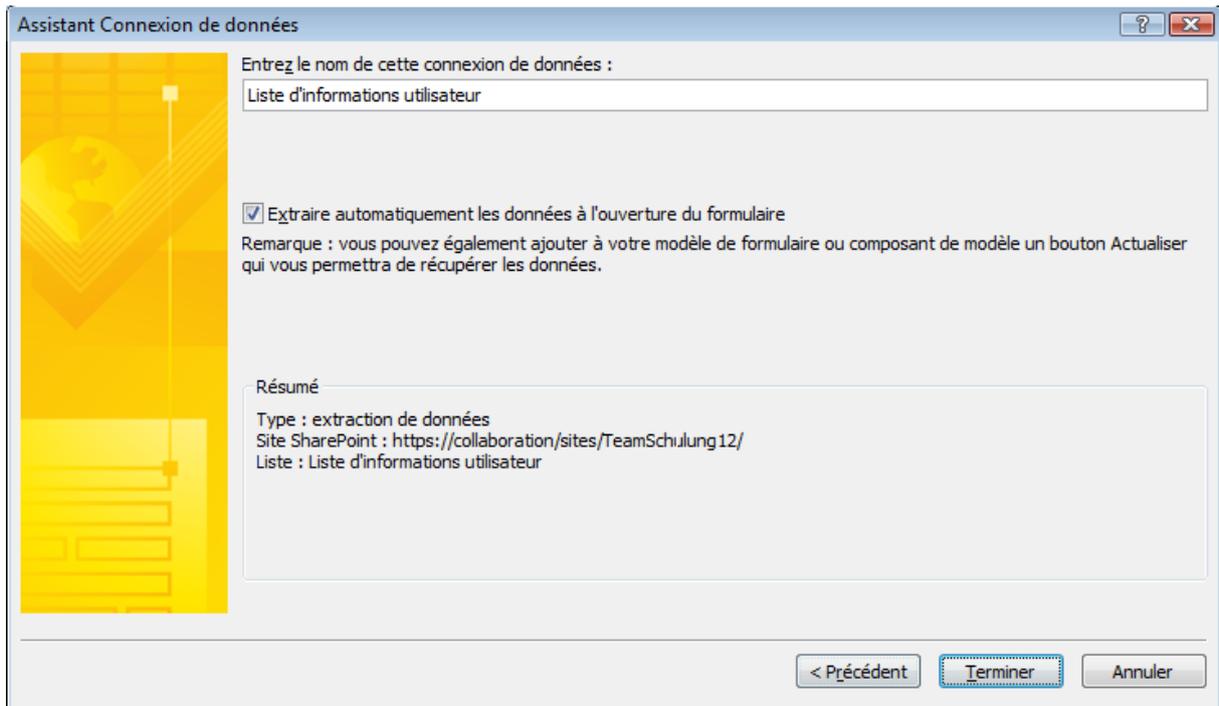
Vous aurez alors:



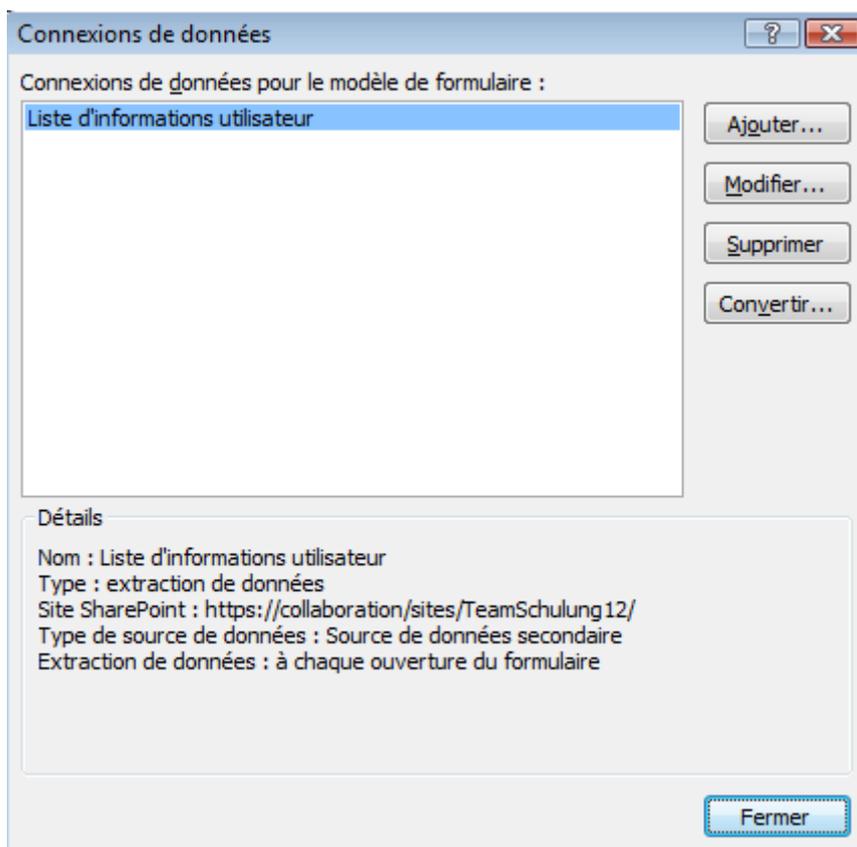
Sélectionnez *Liste d'informations utilisateur* et cliquez sur *Suivant*. Vous aurez alors:



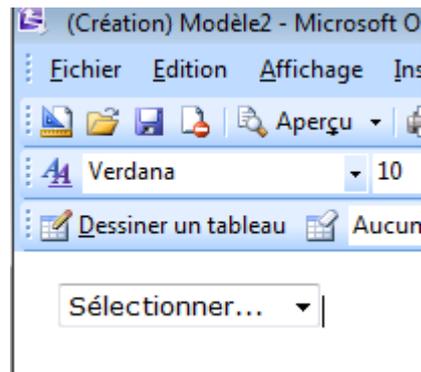
Sélectionnez ce que vous voulez et cliquez deux fois sur *Suivant*:



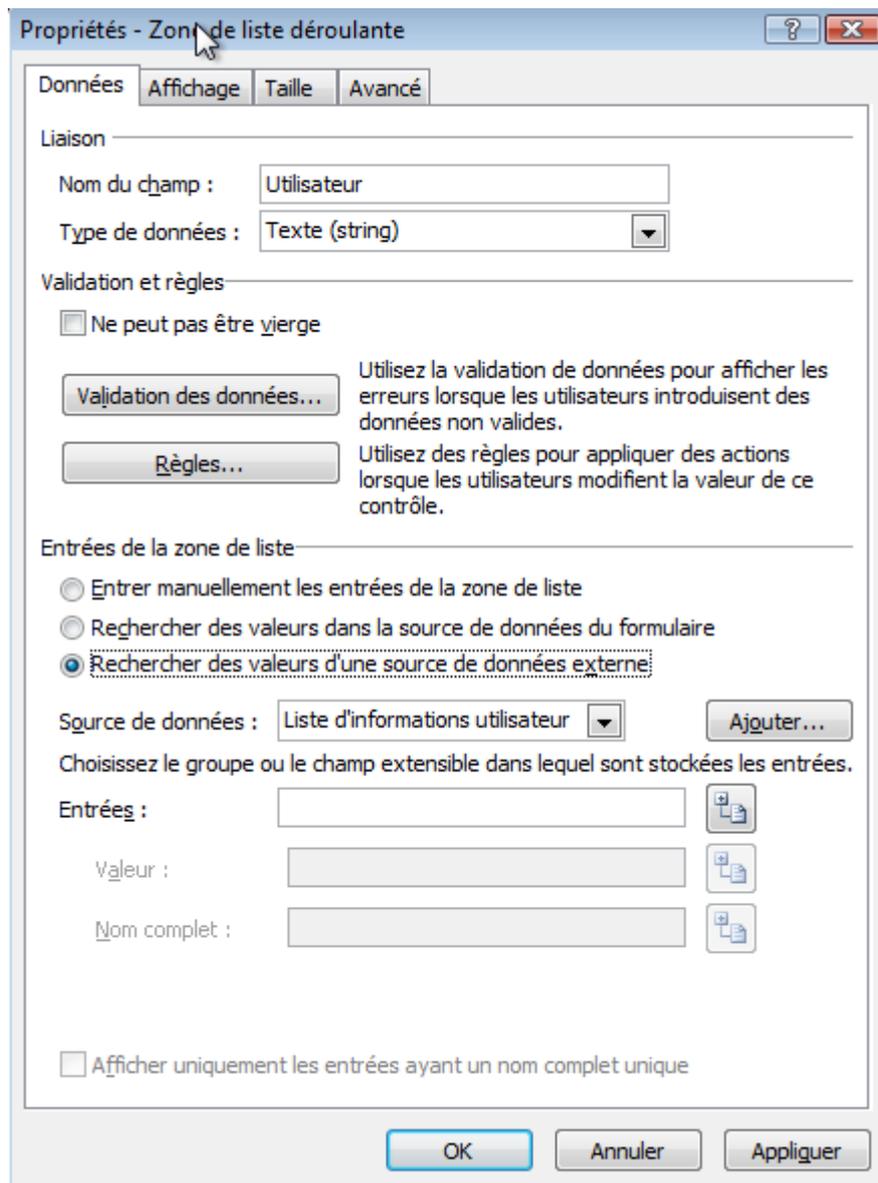
et validez par *Terminer*:



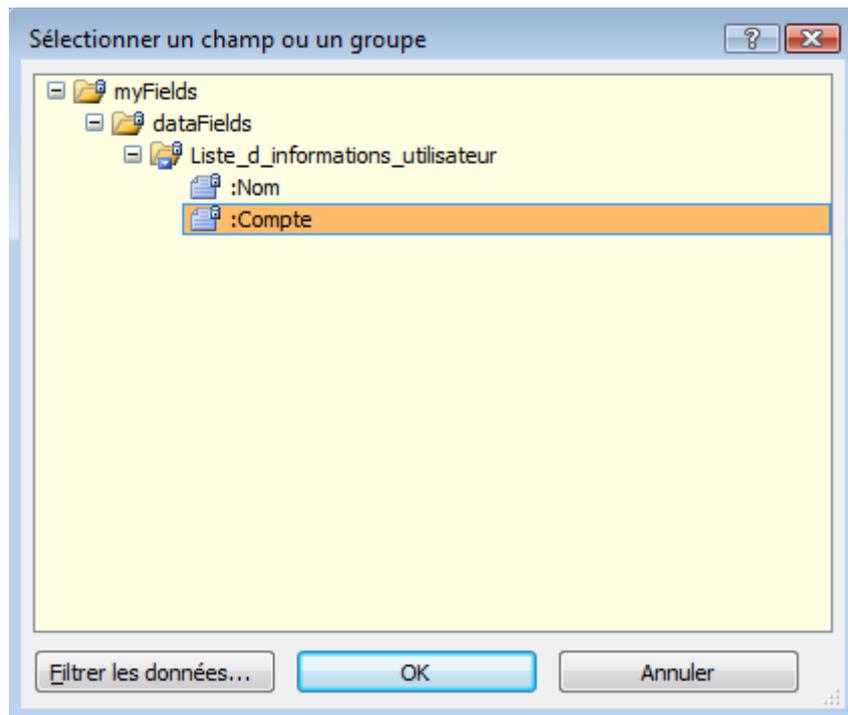
et cliquez sur *Fermer*. Créez une zone de liste déroulante:



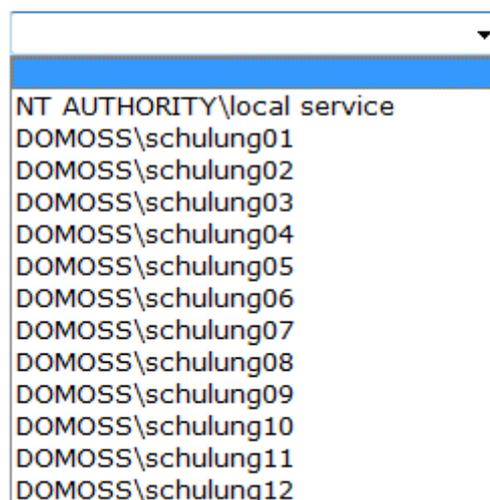
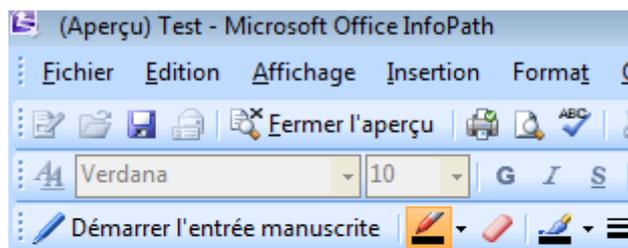
Allez dans les propriétés de la liste déroulante:



Cliquez sur Entrées et sélectionnez par exemple le nom des comptes:

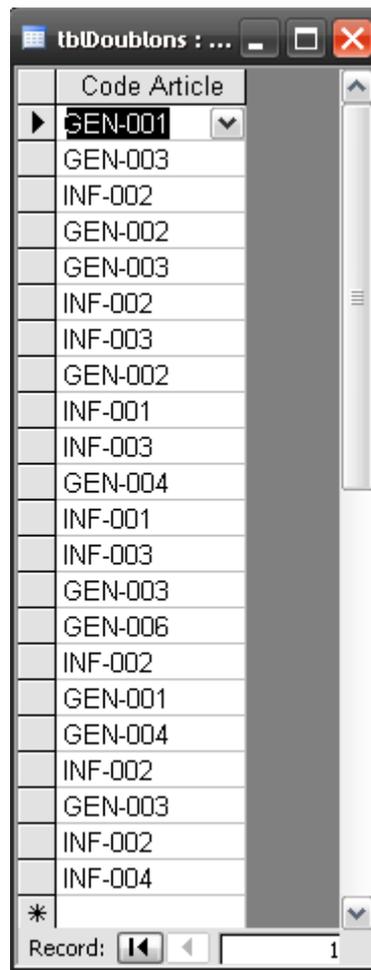


Validez deux fois par OK et voilà le travail qui est fait:

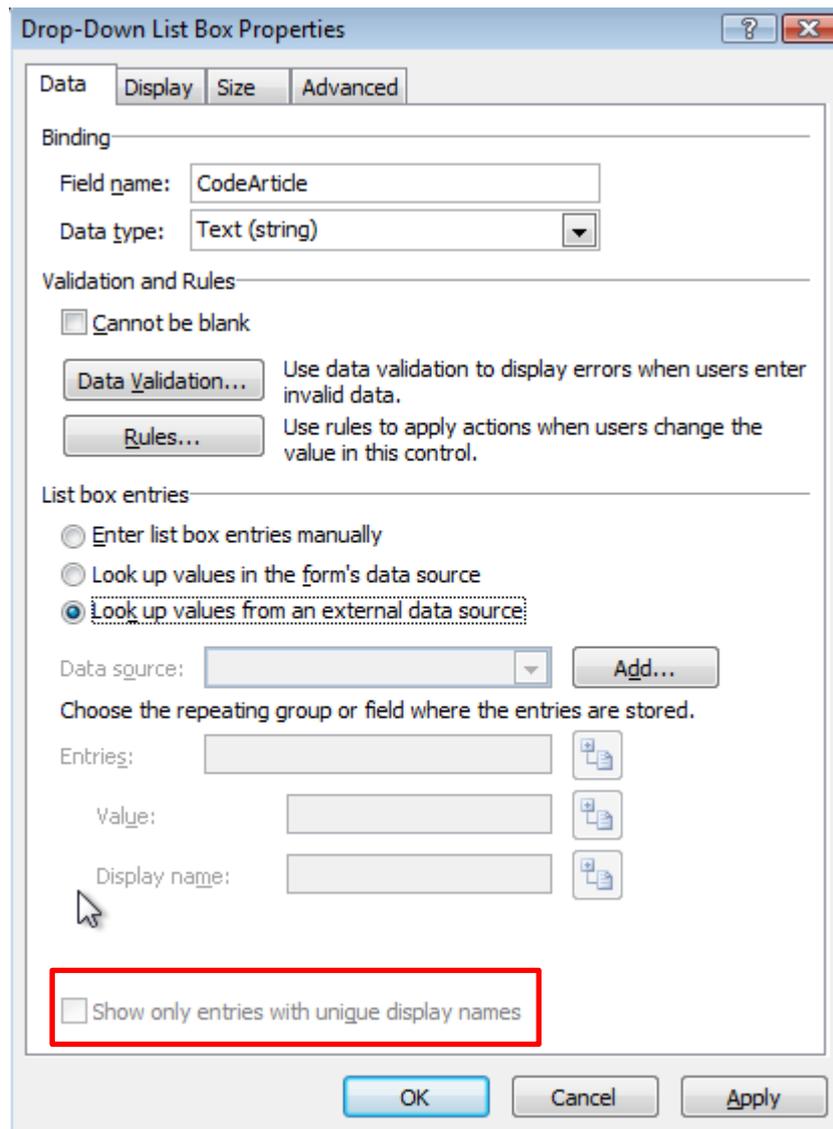


### ***9.5 Suppression des doublons d'une liste déroulante***

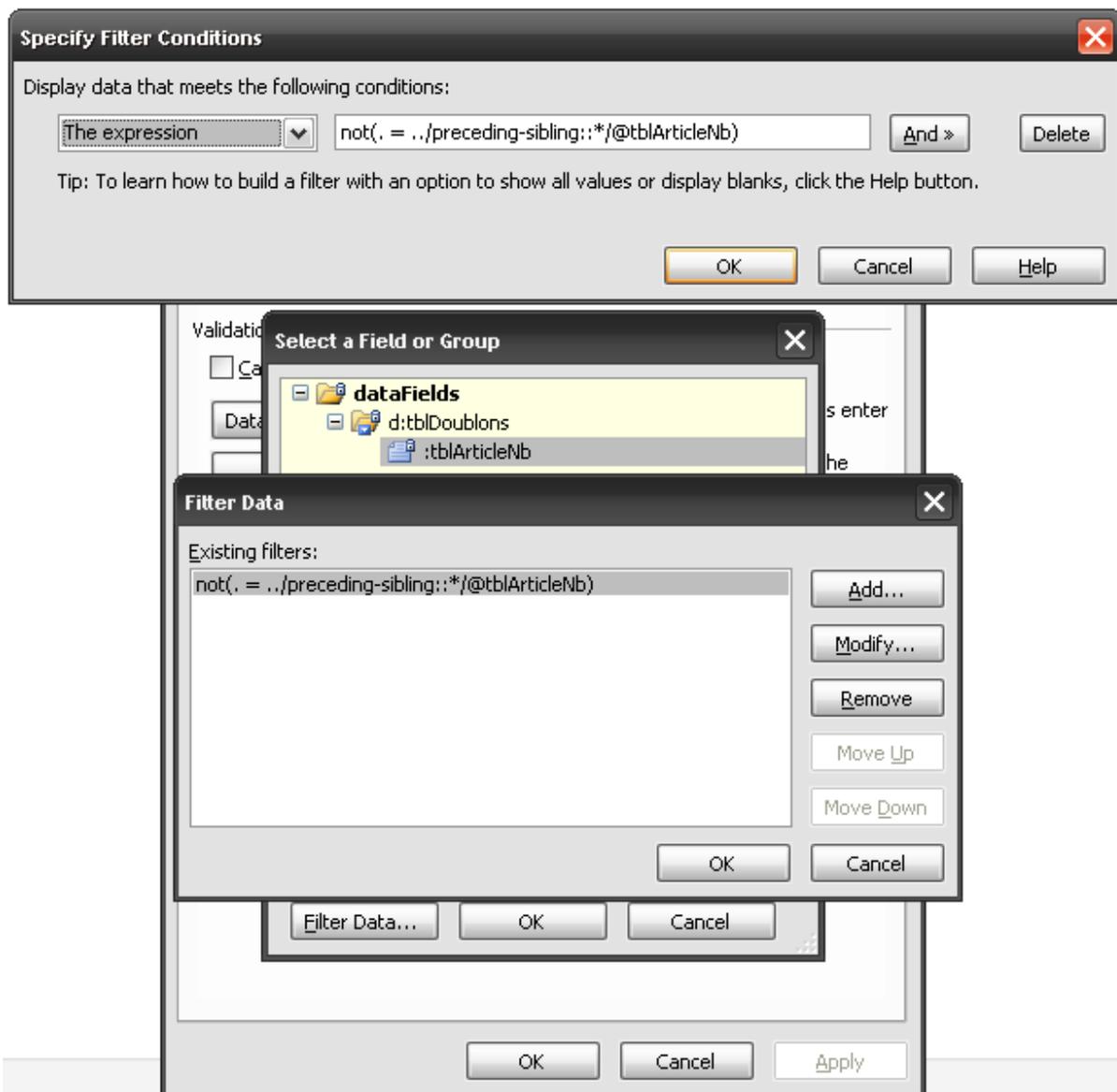
Voici un autre exemple important et souvent demandé sur les forums Internet. Considérons une table source de données (cela peut être du XML, SharePoint, SQL Server ou autre...) avec des doublons comme ci-dessous pour une liste déroulante InfoPath:



Si nous connectons cette table MS Access à une liste déroulante MS InfoPath 2003 nous aurons aussi les doublons. Bien évidemment, pour ceux qui ont InfoPath 2007 et ultérieur cette question ne se pose pas puisqu'ils ont une nouvelle option dans les contrôles de liste déroulante:



Donc pour ceux qui travaillent avec InfoPath 2003, pour palier au problème des doublons, l'astuce consiste à mettre un filtre:



et voilà...

### ***9.6 Filtrage d'une section répétitive sur des sélections antérieures***

Une option intéressante d'InfoPath est de bloquer une table répétitive à des choix se trouvant dans une liste et de limiter la sélection à qu'un seul des éléments de celle-ci.

Un exemple simple pour voir comment faire cela est de prendre un music hall en choisissant la disposition des sièges. Un siège ayant déjà été réservé ne pourra pas être réutilisé une seconde fois.

Nous allons donc créer un formulaire simple qui aura une section répétitive avec une liste déroulante qui sera elle-même peuplée par une autre liste déroulante se trouvant dans le formulaire. Ensuite, un filtre ad hoc sera appliqué à chaque fois à un nouvel élément de la section répétitive afin qu'on ne puisse pas sélectionner un élément qui l'a déjà été au préalable.

Créez d'abord un formulaire de ce type (ce qui ne devrait pas être pas une difficulté maintenant):

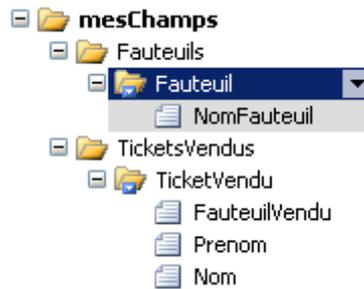
Type de sièges
<input type="text"/>

Tableau extensible

Siège	Nom	Prénom
<input type="text"/>	<input type="text"/>	<input type="text"/>

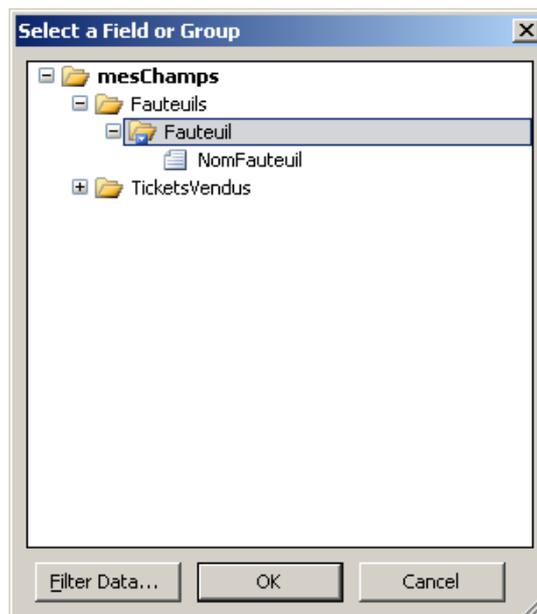
Tableau extensible

et adaptez la structure de données (du moins les noms) de manière à avoir ensuite:



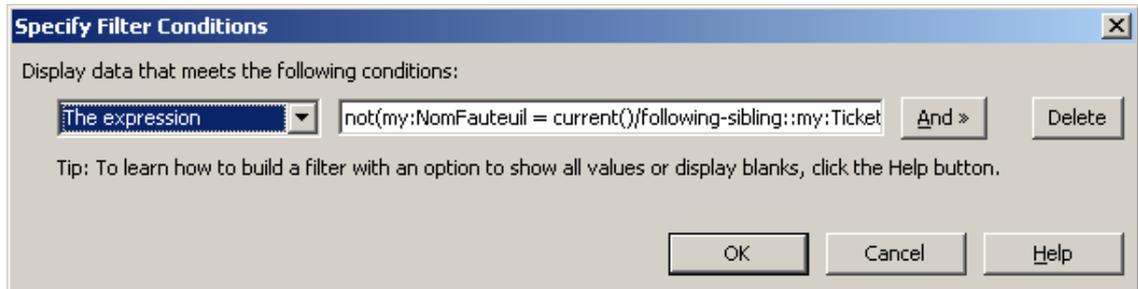
Ensuite:

1. Faites un clic droit sur la liste déroulante sur la première TextBox de la seconde table et choisissez l'option *Change To/Drop Down List Box*
2. Maintenant que nous avons une liste déroulante, double cliquez sur ce contrôle pour ouvrir la boîte de dialogue de ses propriétés
3. Choisissez l'option *Look Up Values In The Form's Data Source* et ensuite cliquez sur le bouton XPath à droit du champ *Entries*
4. De la boîte de dialogue qui apparaît, sélectionnez le nœud *mesChamps/Fauteuil/Fauteuil*



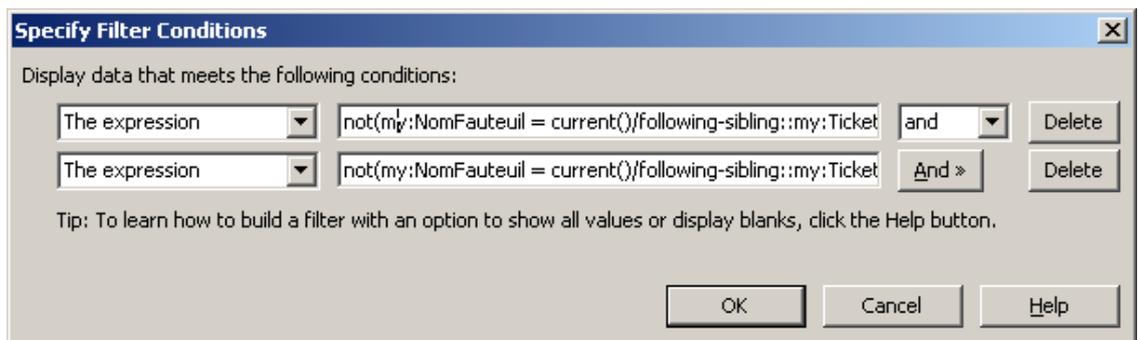
Et ensuite cliquez sur le bouton *Filter Data...*

5. Dans la boîte de dialogue *Filter Data* cliquez sur le bouton *Add*, sélectionnez *The Expression* de la liste déroulante et insérez l'expressions suivante:



```
not(my:NomFauteuil = current()/preceding-sibling::my:TicketVendu/my:FauteuilVendu)
```

6. Cliquez sur le bouton *And* et sélectionnez *Expression* et insérez l'expression suivante:



```
not(my:NomFauteuil = current()/following-sibling::my:TicketsVendus/my:FauteuilVendu)
```

7. Cliquez sur *OK* jusqu'à ce que vous retourniez dans la vue initiale d'InfoPath

Il ne vous reste ensuite plus qu'à tester:

1. Vous activez votre formulaire en mode prévisualisation et saisissez quelques valeurs dans la liste des fauteuils
2. Ensuite, dans la table du dessous, vous sélectionnez un premier fauteuil et ensuite insérez une ligne avec un autre fauteuil
3. Notez que seulement les fauteuils non déjà sélectionnés sont disponibles dans la liste

Type de sièges		
Terrasse		
Parterre		
Balcons		
<input type="checkbox"/> Insérer un élément		

Siège	Nom	Prénom
Terrasse		
Parterre		
Balcons		

### 9.7 Formulaire de saisie connecté (MS Access)

Nous allons maintenant faire un formulaire dynamique lié directement à la table *tblFournisseurs* de notre base MS Access avec des listes déroulantes, etc. etc. L'échange des données entre MS Access et MS InfoPath se faisant par ADO (donc sans faire usage de XML mais bon c'est quand même bien de faire un exemple ne serait-ce que pour voir la facilité d'usage de MS InfoPath).

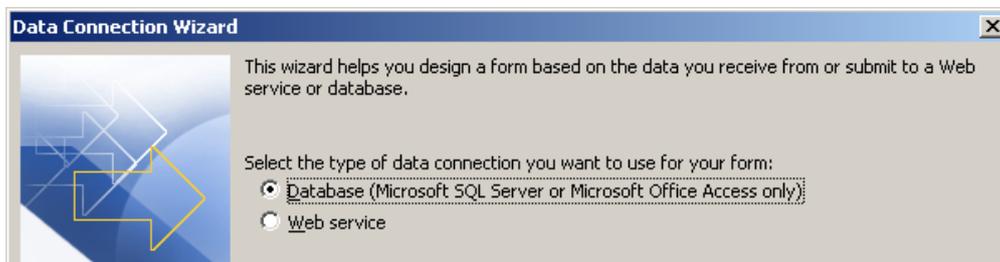
Ouvrez InfoPath (au minimum la version SP1 de 2003 car la version InfoPath sans le SP1 n'avait pas cette fonctionnalité intégrée) et sélectionnez:



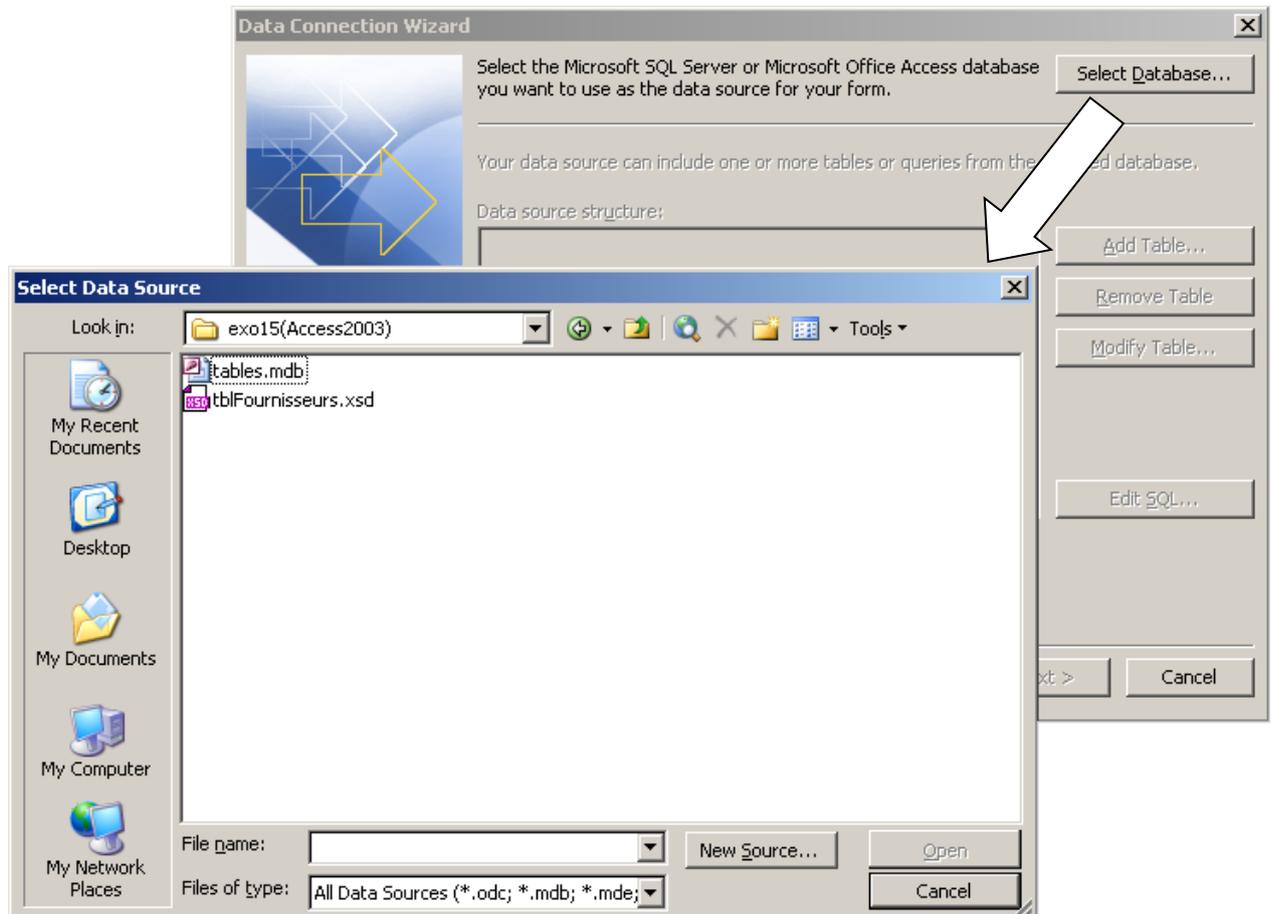
Dans le volet Office cliquez sur:



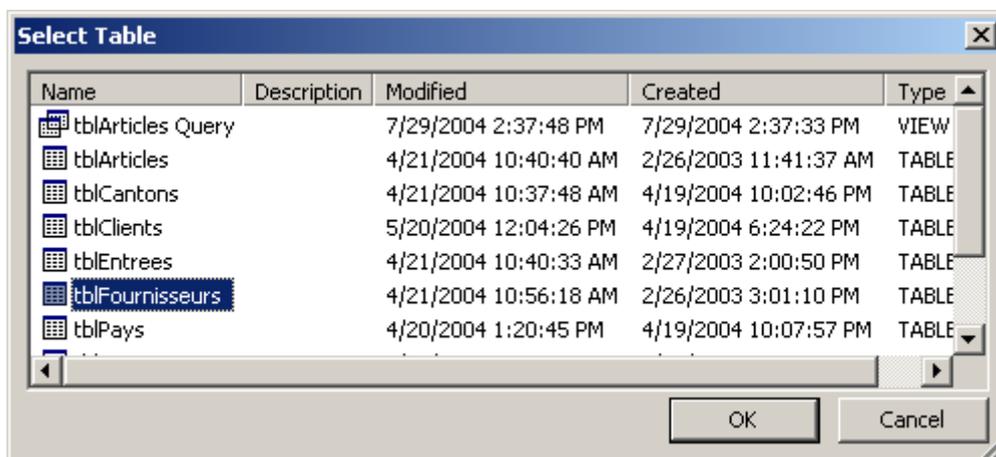
Afin que nous puissions nous connecter à la base MS Access (les Webservices utilisant les flux XML par SOAP ne seront pas traités dans ce document – c'est que pour les spécialistes):



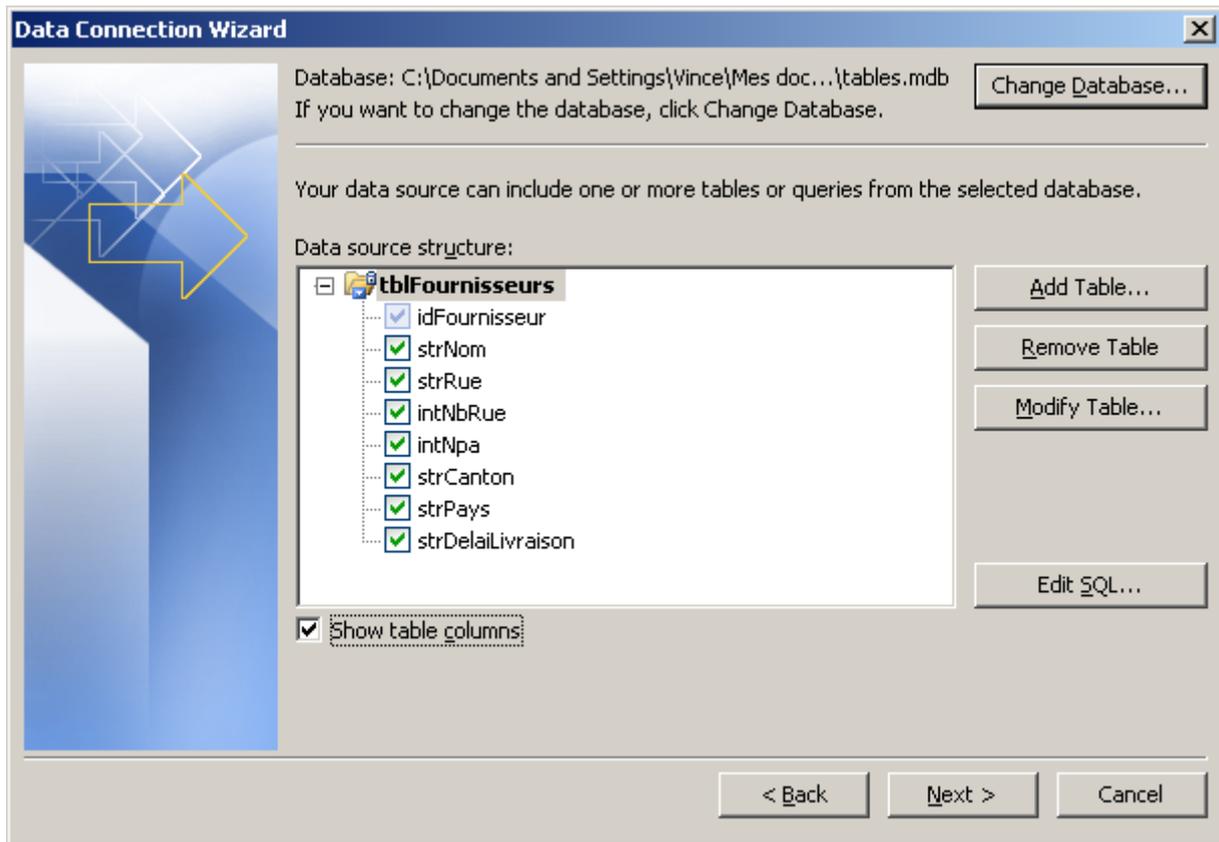
Ensuite la fenêtre suivante apparaît (cliquez sur *Next*):



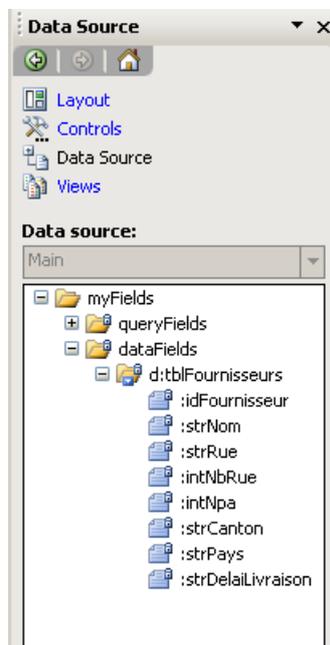
et après avoir sélectionné *tables.mdb* sélectionnez *tblFournisseurs*:



et vous devez avoir:



Cliquez sur *Next*, donnez un nom à votre connexion (*Fournisseurs* par exemple) et sur *Finish*. Vous devrez avoir le résultat suivant dans le volet Office:



A gauche sur la feuille (InfoPath SP1), vous avez une zone *Drag data fields here* (vous pouvez effacer la partie concernant la *Query*). Glissez-y le nœud *d:tblFournisseurs*:

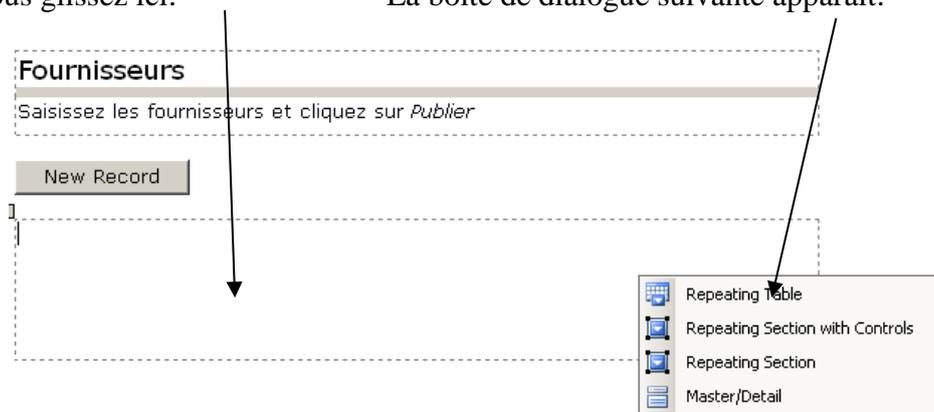
**Fournisseurs**

Saisissez les fournisseurs et cliquez sur *Publier*

New Record

Quand vous glissez ici:

La boîte de dialogue suivante apparaît:



Dans la boîte de dialogue, sélectionnez *Repeating Section with Controls*. Vous aurez:

**Fournisseurs**

Saisissez les fournisseurs et cliquez sur *Publier*

New Record

Id Fournisseur:  tblFournisseurs

Str Nom:

Str Rue:

Int Nb Rue:

Int Npa:

Str Canton:

Str Pays:

Str Delai Livraison:

Repeating Section

Vous pouvez changer le nom des étiquettes et supprimer le champ *id Fournisseur* (puisque c'est une clé primaire *AutoNumber* dans notre base Access):

**Fournisseurs**

Saisissez les fournisseurs et cliquez sur *Publier*

New Record

Nom:

Rue:

Rue:

Npa:

Canton:

Pays:

Délai Livraison:

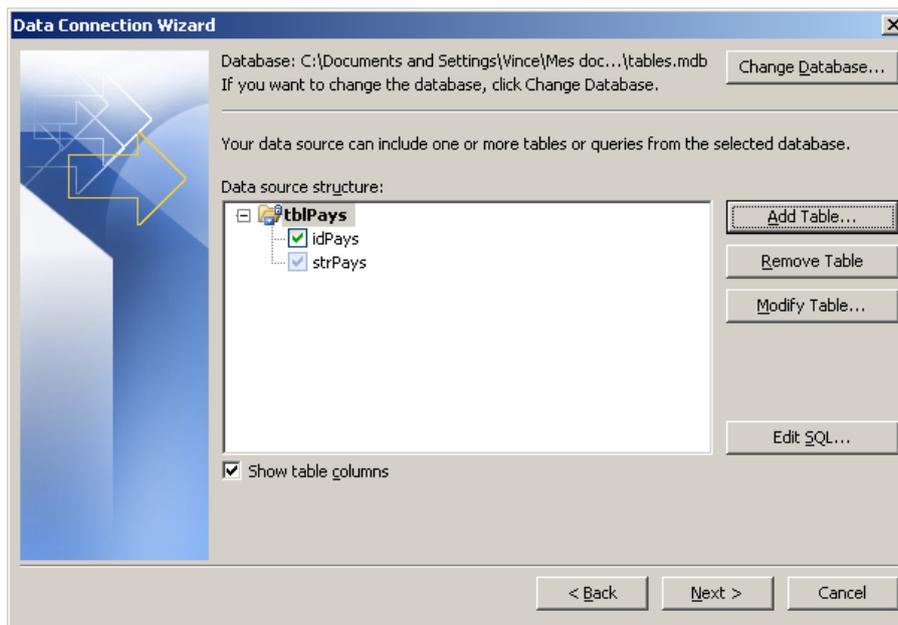
Repeating Section

Maintenant nous aimerions que lorsque l'utilisateur clique sur *Canton* ou *Pays* que la liste des cantons ou pays disponibles dans la base Access (dans les tables *tblCantons* et *tblPays*) s'affiche à l'écran (pour que l'utilisateur n'ait pas à les saisir).

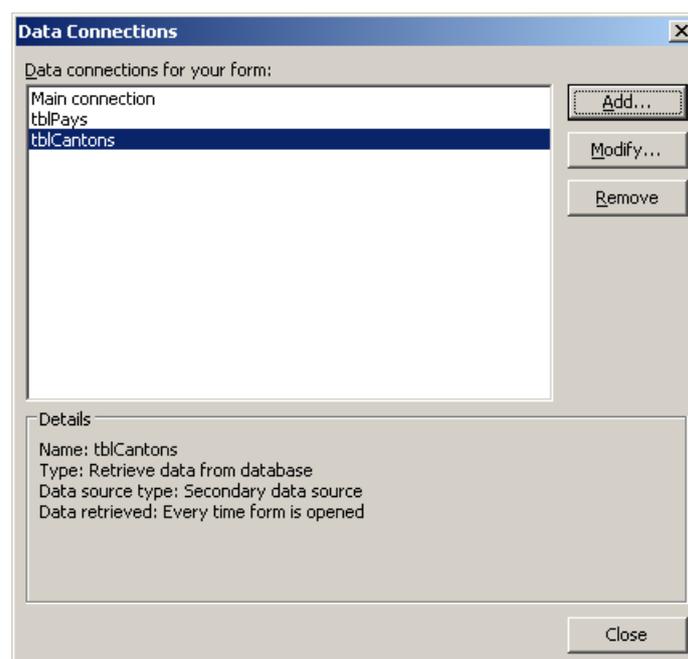
Pour ce faire, il faut aller dans le menu *Tools/Data Connection*:



*Add...* ensuite *Receive Data* ensuite *Database (Microsoft SQL Server or Microsoft Access only)* ensuite *Select Database* et allez chercher la base *tables.mdb* et la table *tblPays*:



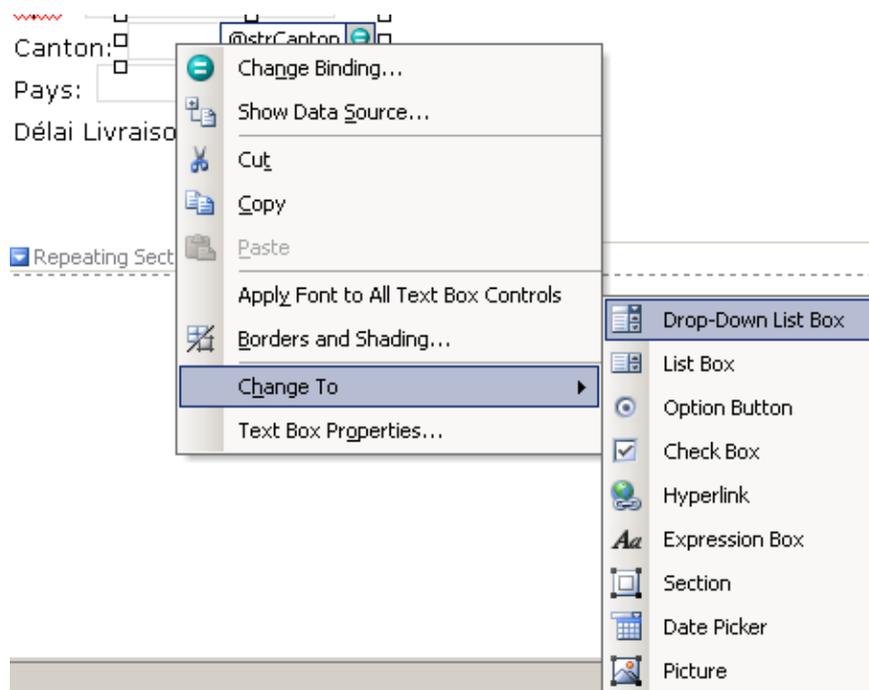
Cliquez sur *Next*, donnez un nom à la connexion et recommencez l'opération pour la table des cantons. Vous aurez alors dans la fenêtre des connexions (*Tools/Data connections*) trois connexions:



Cliquez sur *Close*.

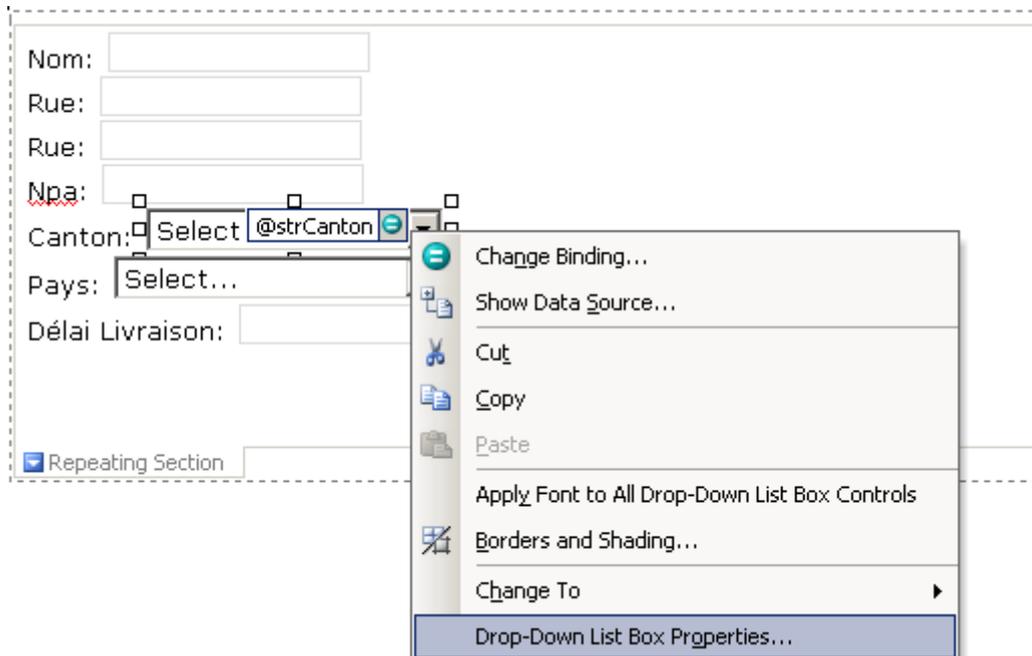
Pour relier les champs *Cantons* et *Pays* du formulaire InfoPath aux tables de la base Access et les transformer en listbox voici comment il faut procéder:

Il faut d'abord transformer les champs texte en Drop Down Listbox. Pour ce faire, bouton droit et:

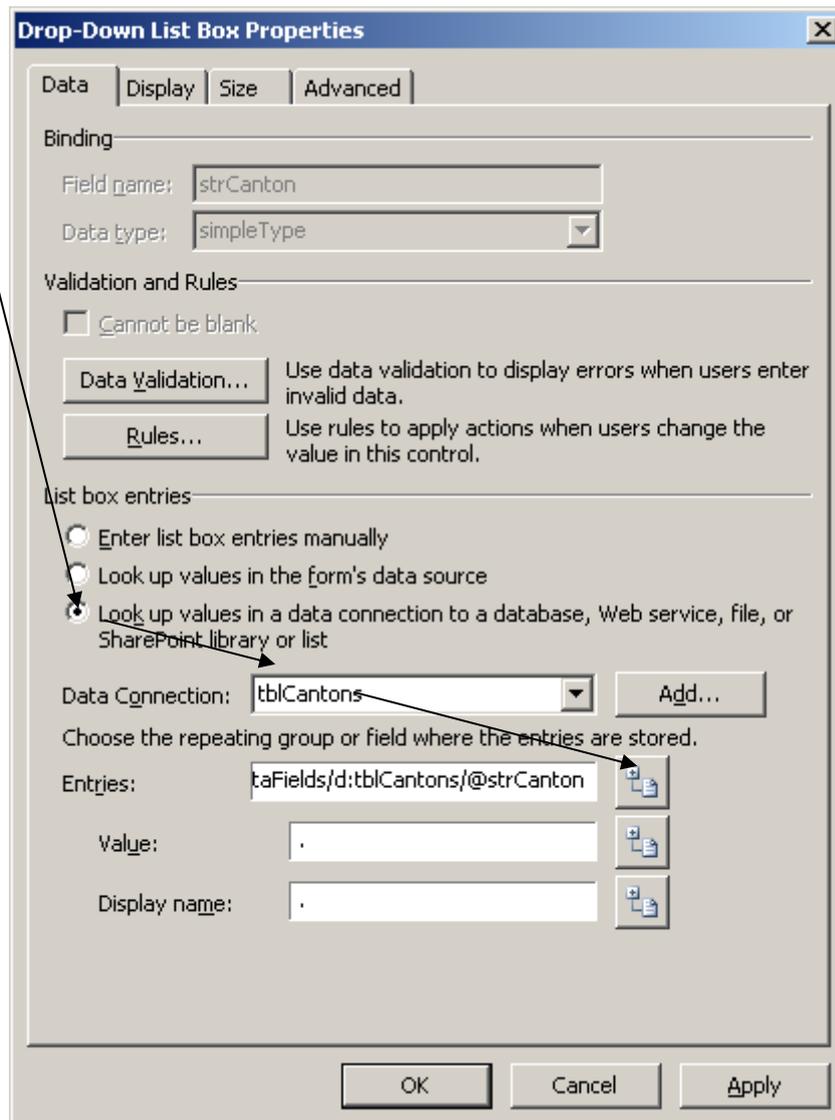


et idem pour les pays.

Maintenant il faut "peupler" les ListBox avec le contenu des tables de la base Access. Pour ce, clique droit et:



Ensuite:



Vous cliquez sur *OK* et vous faites idem pour les pays.

Vous enregistrez votre formulaire (\*.xsn) et l'ouvrez par un double clique depuis votre raccourci sur le bureau ou votre explorateur Windows (si lorsque vous l'ouvrez il vous dit que les données se trouve sur un autre domaine vous validez par *Oui*):

**Fournisseurs**

---

Saisissez les fournisseurs et cliquez sur *Publier*

Nom:   
Rue:   
Rue:   
Npa:   
Canton:   
Pays:   
Délai Livraison:

Repeating Section

Cliquez sur une des ListBox et oh miracle:

**Fournisseurs**

---

Saisissez les fournisseurs et cliquez sur *Publier*

Nom:   
Rue:   
Rue:   
Npa:   
Canton:   
Pays:   
Délai Livr:

Insert item

et maintenant pour la saisie de données ? Eh ben c'est très simple, on saisit le premier *Record* et le deuxième (cliquez sur *Insert Item*) etc...:

## Fournisseurs

Saisissez les fournisseurs et cliquez sur *Publier*

Nom:

Rue:

N° Rue:

NPA:

Canton:

Pays:

Délai Livraison:

Nom:

Rue:

N° Rue:

NPA:

Canton:

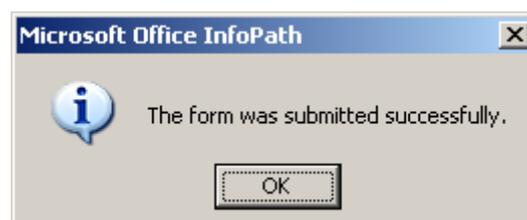
Pays:

Délai Livraison:

et quand c'est fini, on clique sur:



on a:



☺ on est forts hein...

Vincent ISOZ

Et si nous allons ouvrir notre table Access *tblFournisseurs*, nous avons bien:

	idFournisseur	Nom Société	Rue	N°	N.P.A	Commune	Pays
+	1	Duplibureau	Ruelle du Soleil	0	2003	Neuchâtel	
+	2	La maison du papier	Rue des Anges	17	1010	Lausanne	
+	3	Tartanpion	Grand-Rue	115	1205	Genève	
+	4	RSR	Ch. des Vernet	22	1024	Genève	
+	5	SSR	Burgstrasse	225	8300	Neuchâtel	Allemagne
*	(AutoNumber)						

Remarque: Pour supprimer le bouton d'édition de MS InfoPath sur les postes clients allez dans *Tools/Options* et activez *Enable protection*.

Voilà pour les bases de MS InfoPath et XML.

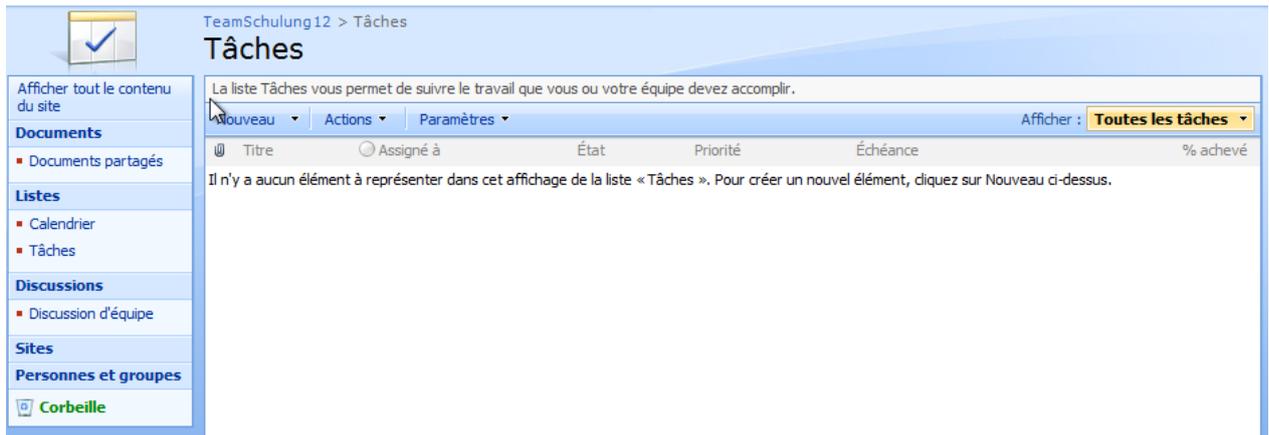
Bien sûr il faut avoir en tête que tout cela communique avec l'ensemble des produits de la gamme MS Office 2003.

### ***9.8 Formulaire de saisie connecté (MS Access/SharePoint)***

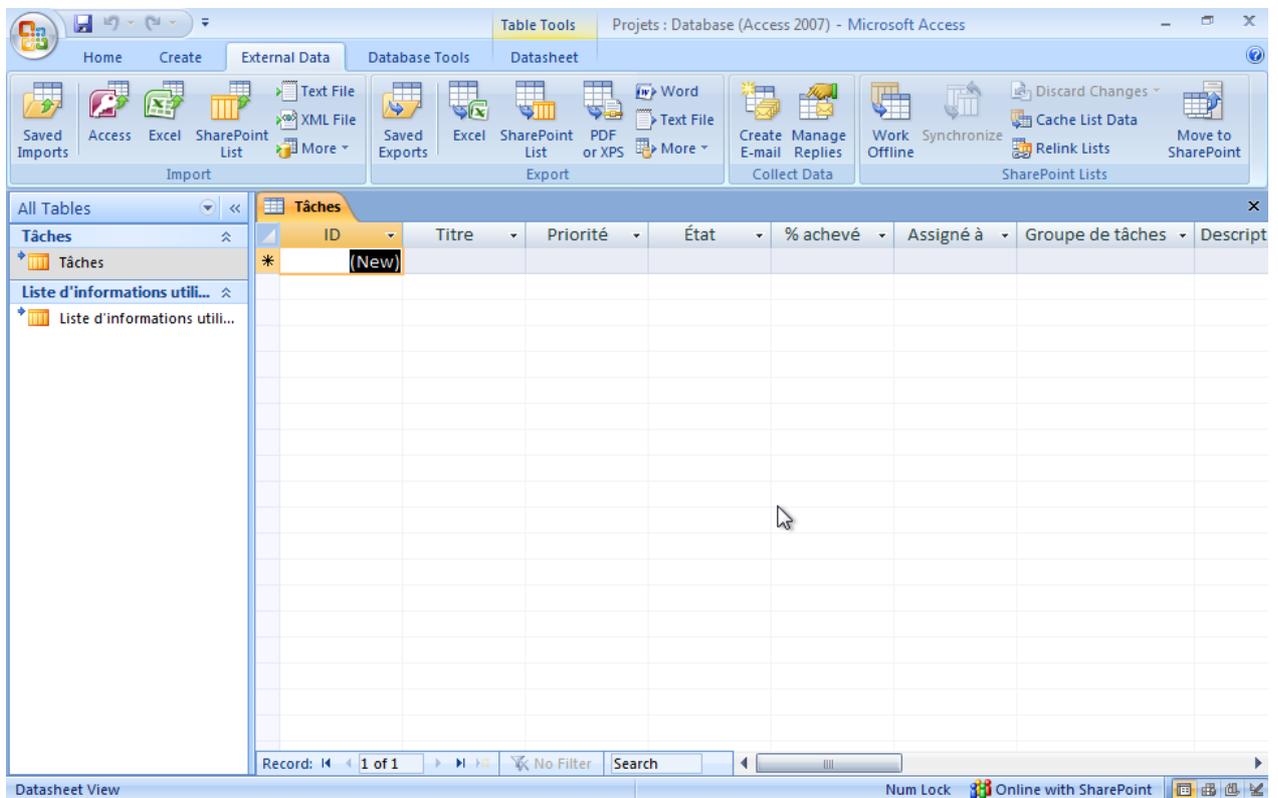
L'idée va être la suivante:

1. Vérifier si nous pouvons créer un formulaire InfoPath d'une table Access à l'origine liée à une table SharePoint (comme avant mais donc avec une petite différence quant à la source)
2. Vérifier si ce formulaire peut se substituer de façon pertinente à la méthode complexe de création d'un formulaire InfoPath d'une liste SharePoint 2003 ou 2007 (sachant qu'avec 2010 la question ne se pose plus).
3. Vérifier si nous pouvons distribuer ce formulaire par e-mail et que la saisie se retrouve dans SharePoint.

Nous partons pour l'exemple d'une liste de tâches SharePoint:

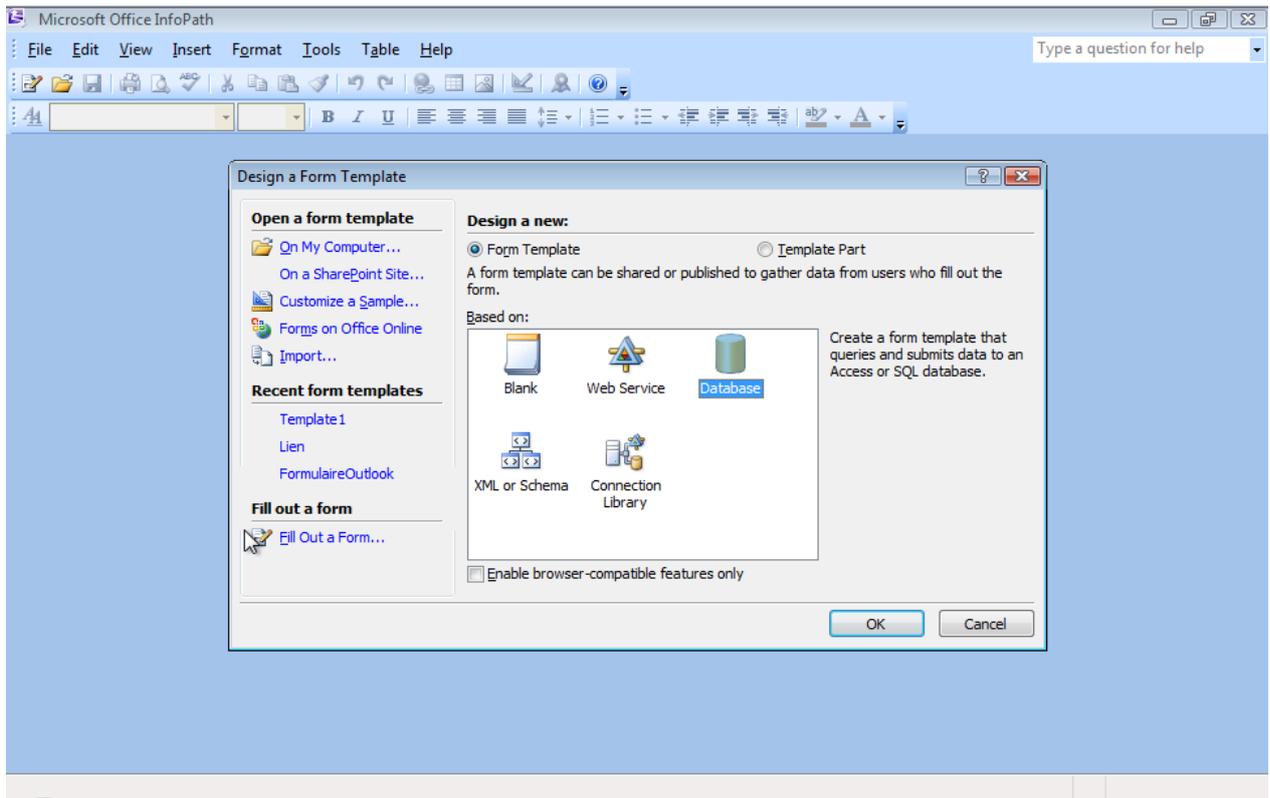


et nous l'avons déjà liée à une base MS Access 2007 (pour savoir comme lier une liste SharePoint à Access, se référer à mon livre électrique sur MS Access):

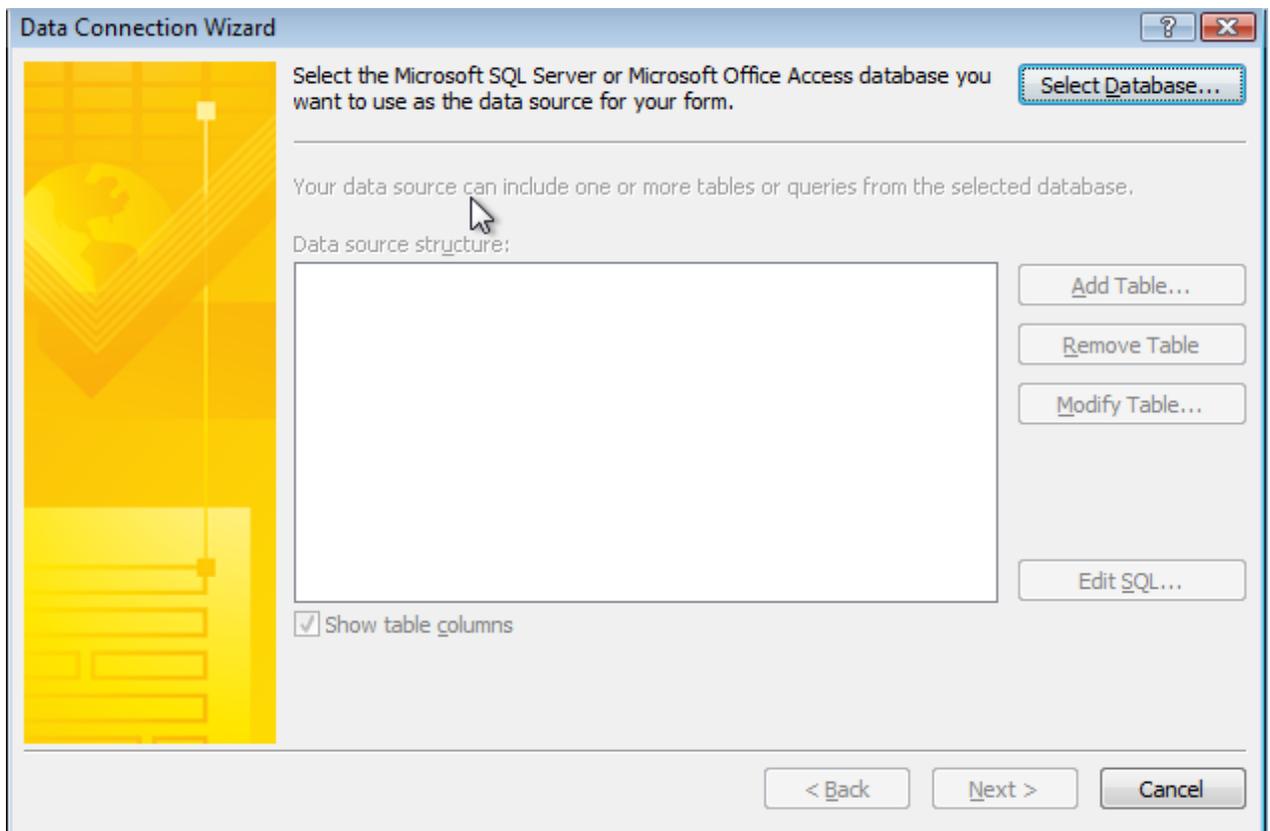


Vérification du point (1):

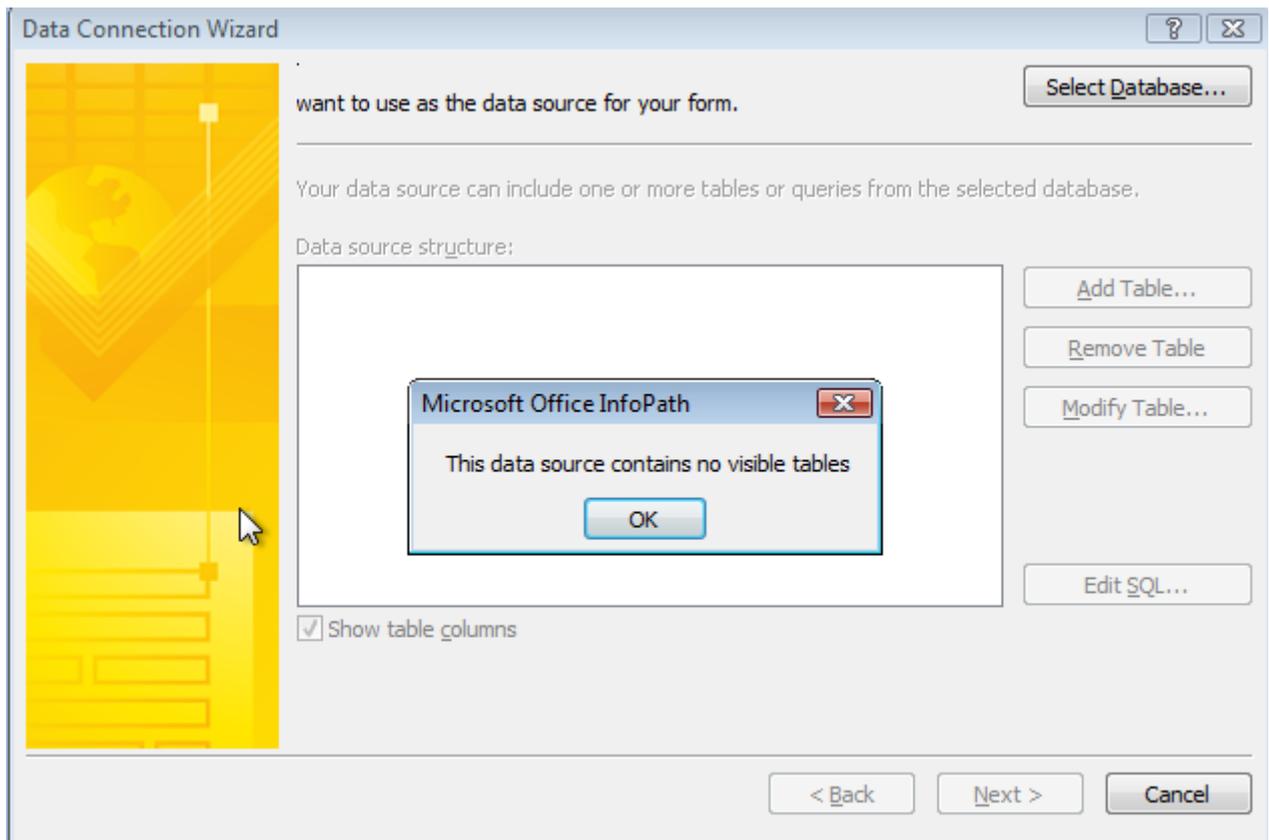
Nous ouvrons InfoPath et créons un formulaire à partir d'une base de données:



nous validons par **OK**:



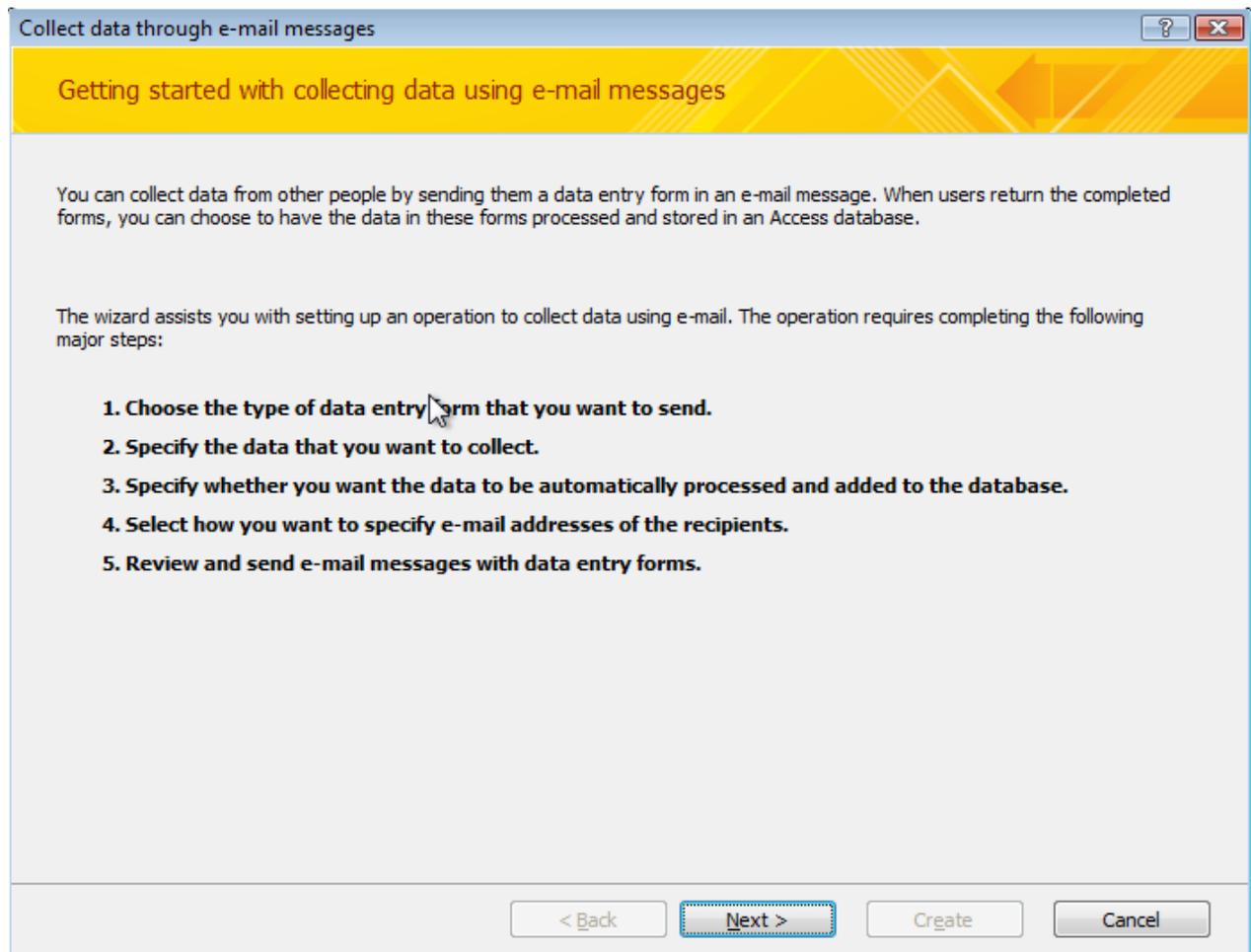
Nous cliquons sur Select Dabase en en validant le choix de notre base Access, nous avons:



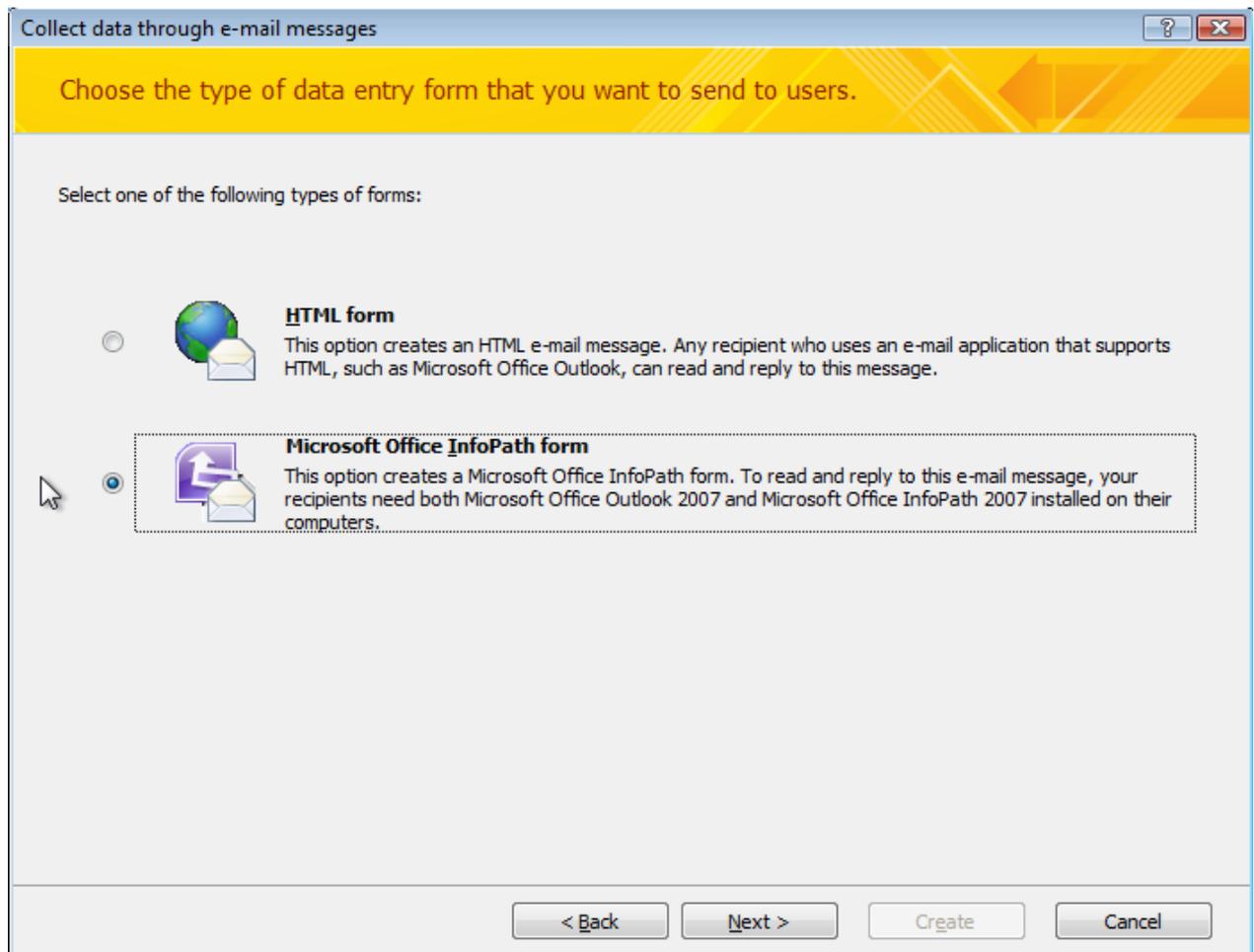
ce qui répond à la première question. Mais essayons d'une autre manière... en passant directement pas MS Access:



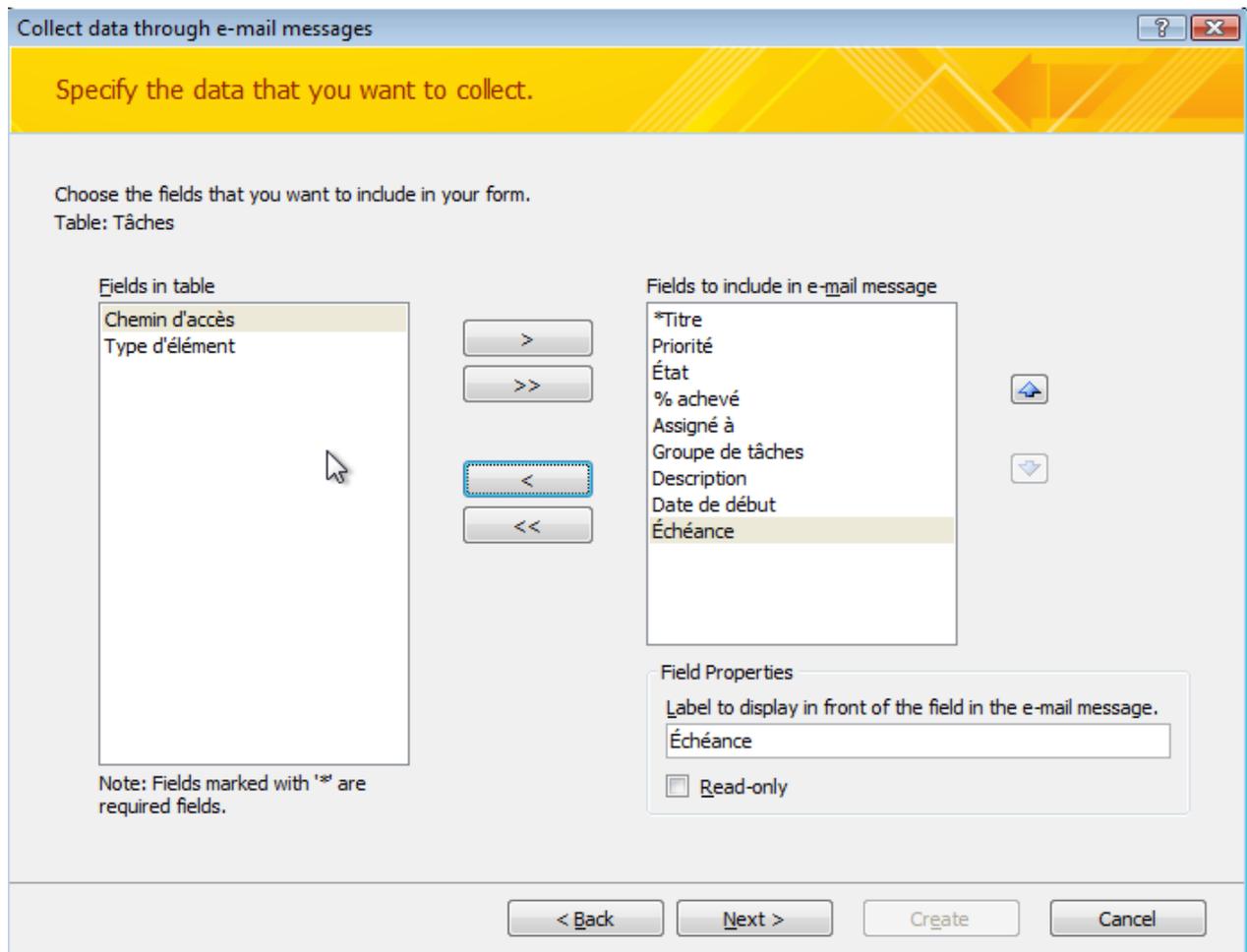
Nous sélectionnons la table *Tâches* et cliquons sur **Create E-Mail** dans le groupe **Collect Data** du ruban **External Data**:



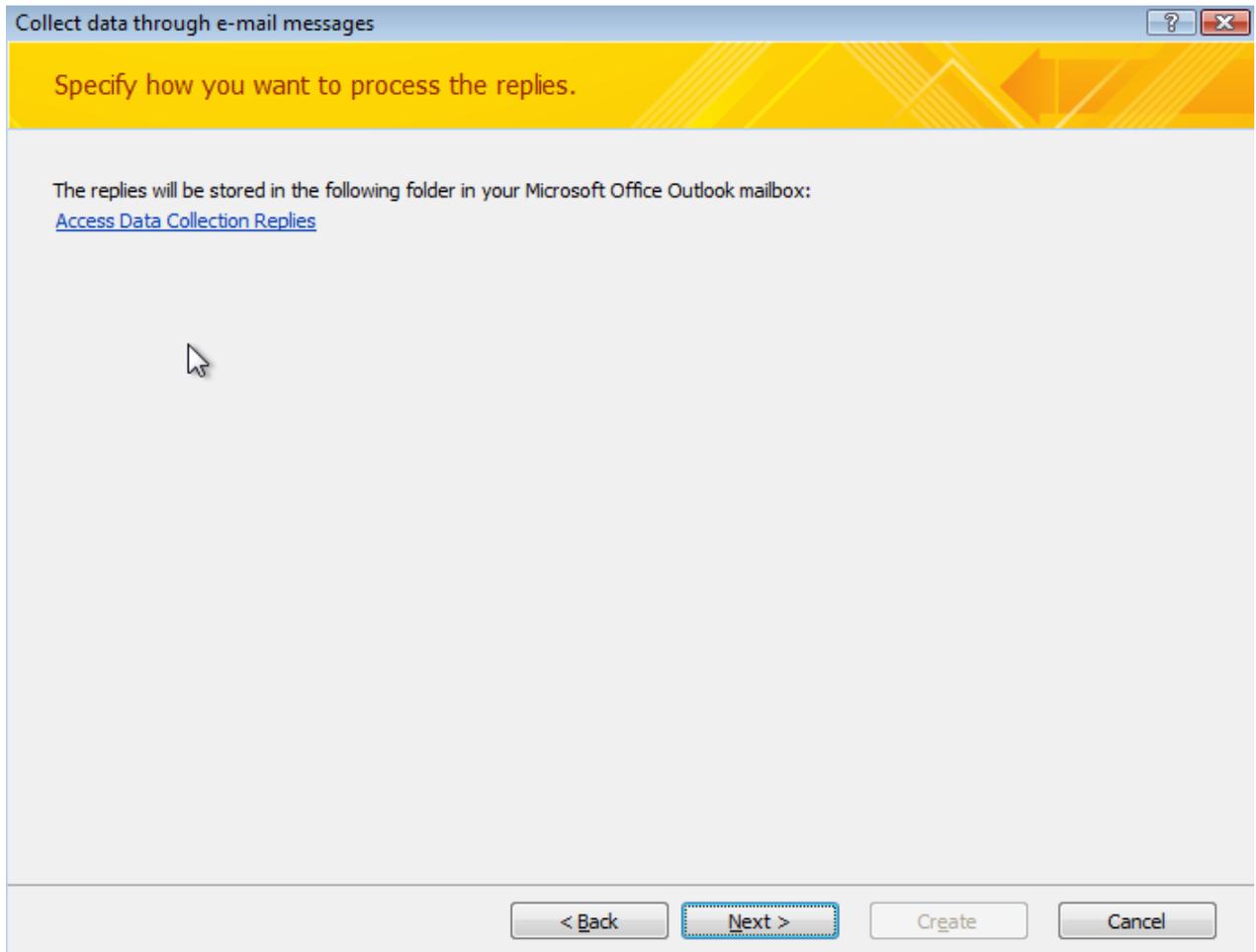
Nous cliquons sur **Next**:



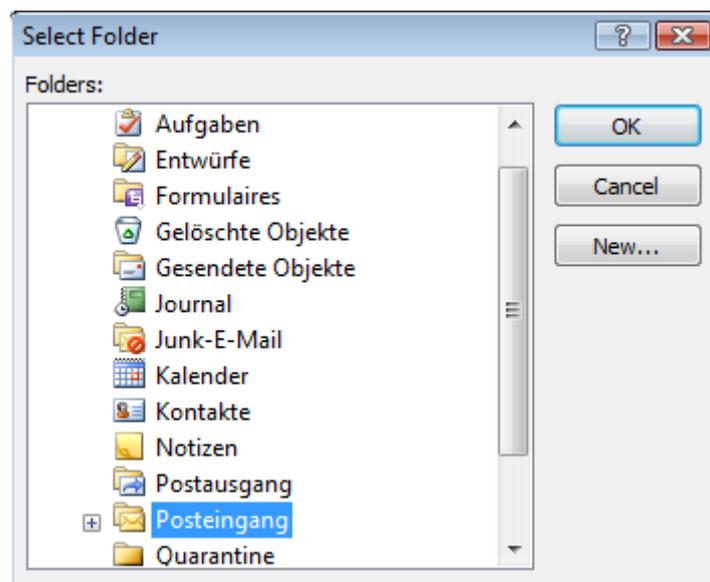
et nous sélectionnons bien évidemment **Microsoft Office InfoPath form** et nous cliquons sur **Next**:



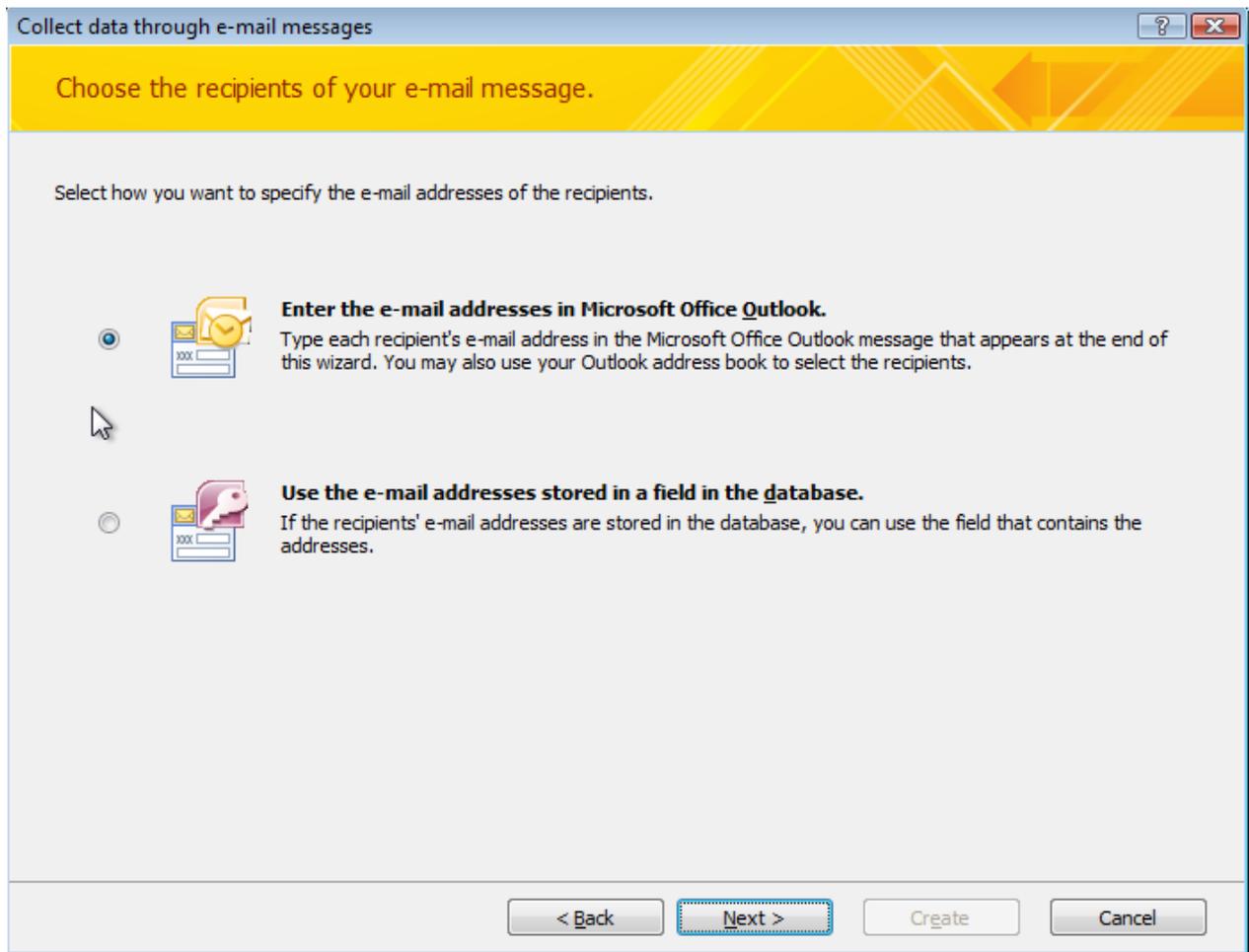
Nous avons pris les champs qui nous intéressaient. Nous cliquons sur **Next**:



si vous cliquez sur le lien Access Data Collection Replies, vous pouvez choisir où les réponses seront stockées dans Outlook:



Nous allons laisser la boîte de réception par défaut et nous cliquons sur **Next**:



Nous cliquons sur **Next**:

Collect data through e-mail messages

Customize the e-mail message

Review the subject and introduction of the e-mail message and make necessary changes.

Subject

Add Tâches Table Form

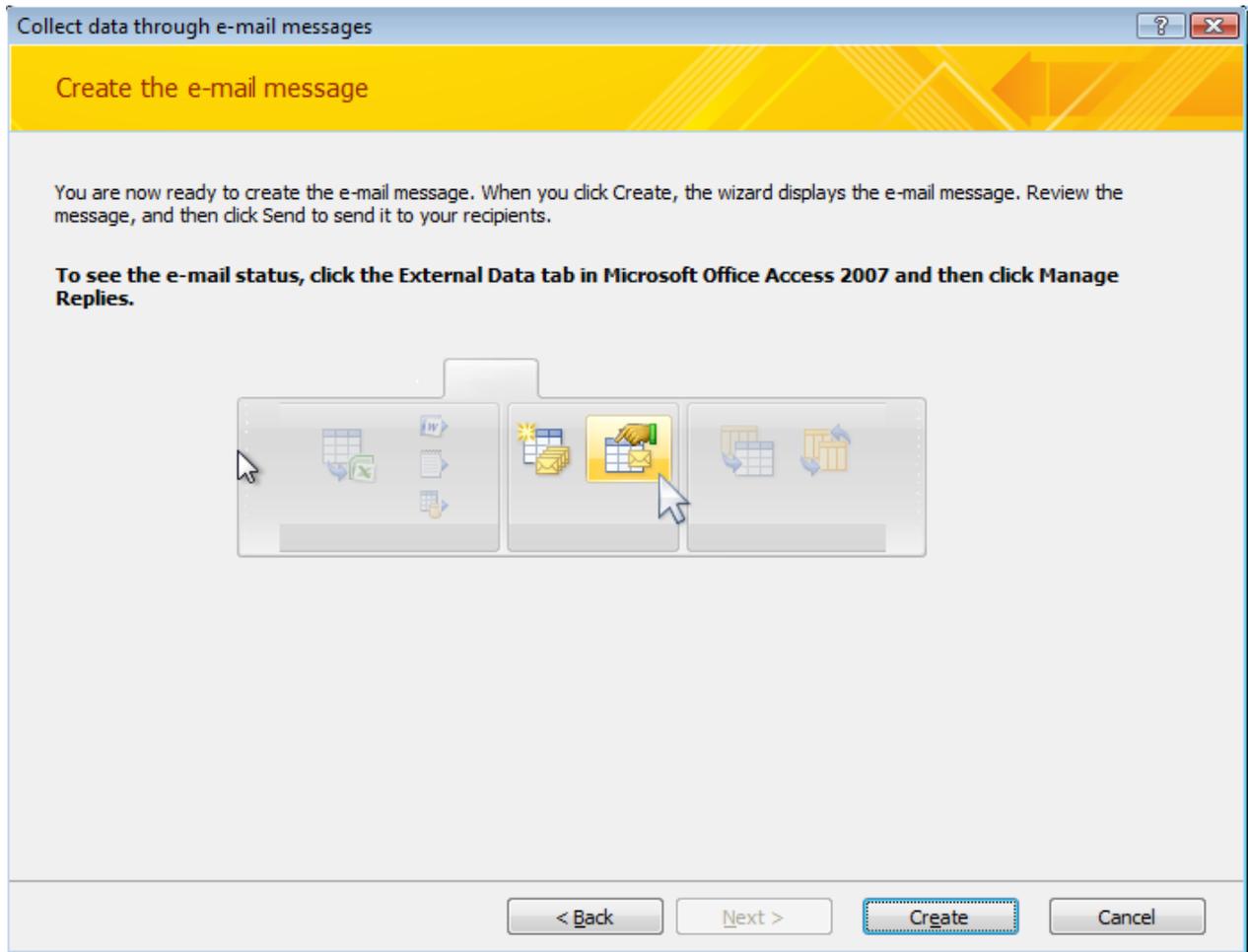
Introduction

Fill out the form included in this message and send it back to me.

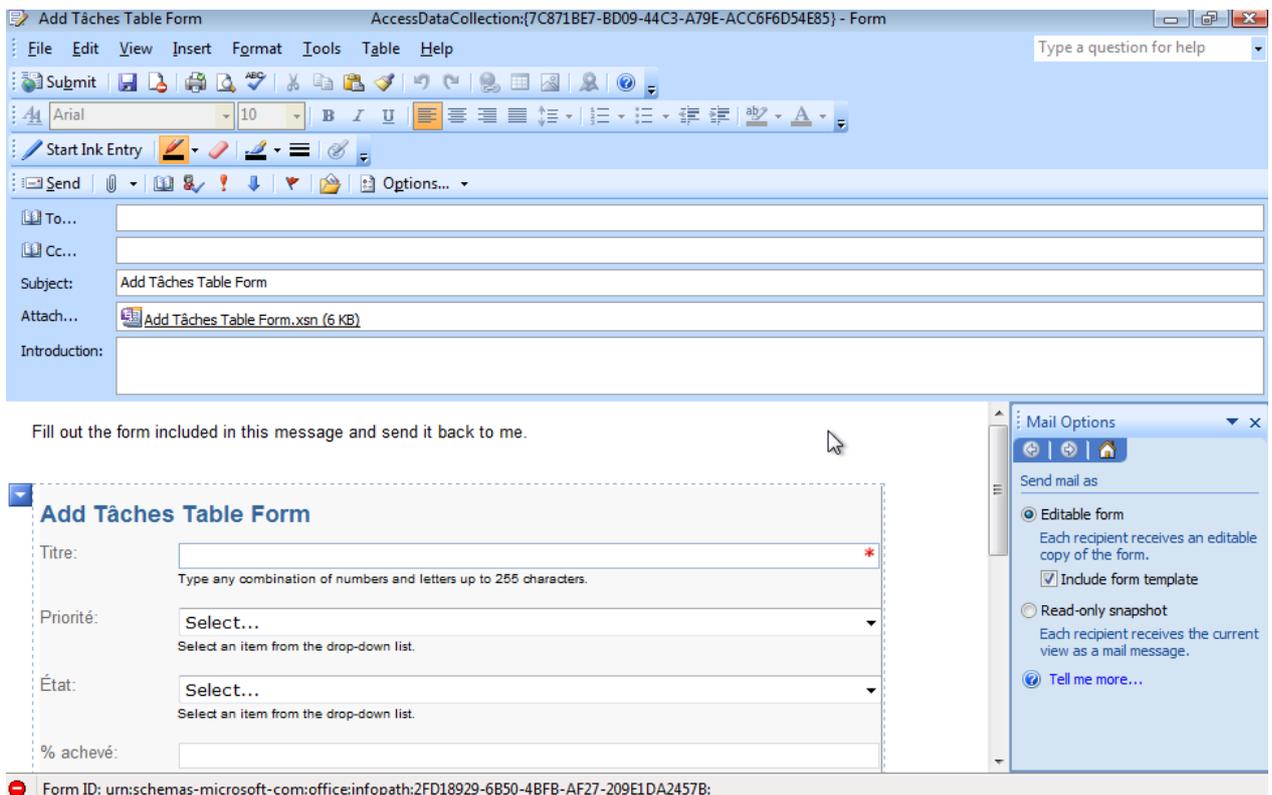
Review and finalize the introductory text in the e-mail message

< Back   Next >   Create   Cancel

Nous cliquons sur **Next**:



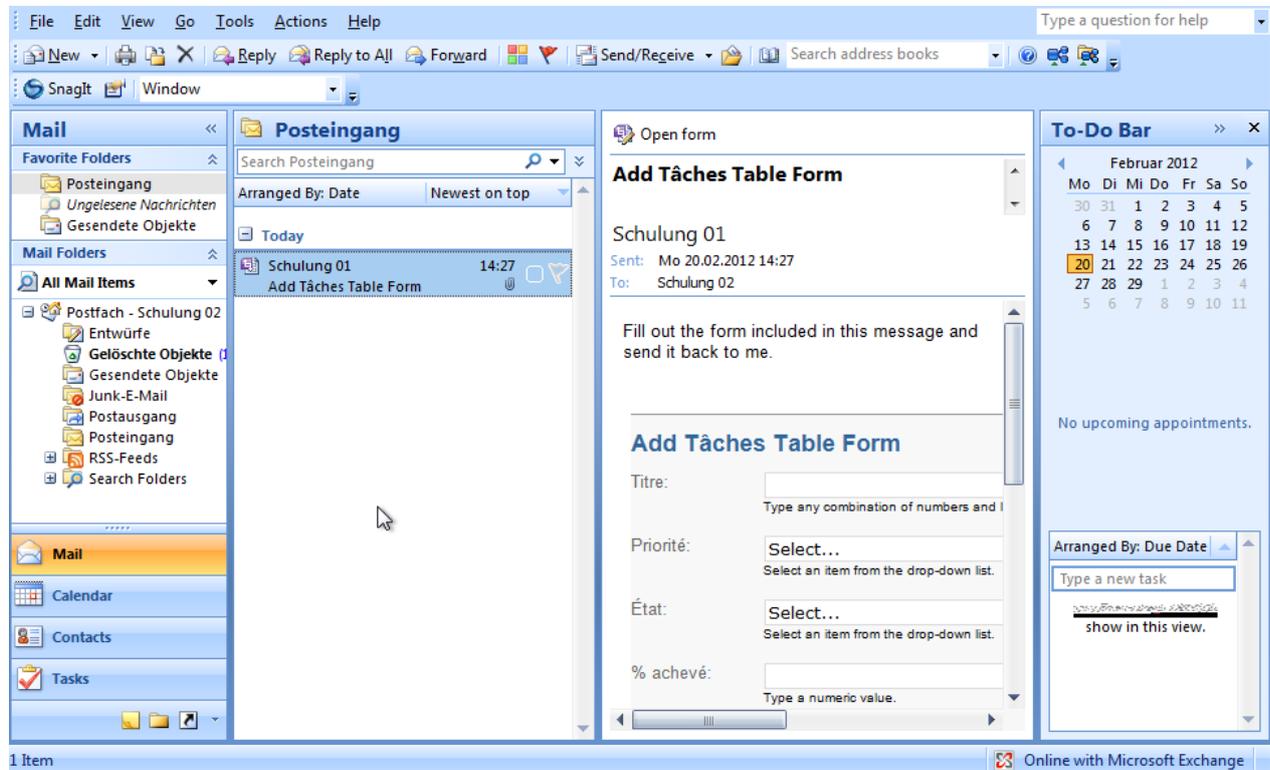
Nous cliquons sur **Create** et un mail va se créer:



Vincent ISOZ

**Remarque:** La limite de cette technique est que vous ne pouvez pas modifier le formulaire généré par MS Access dans InfoPath Designer sinon quoi vous n'aurez plus la fonctionnalité d'import vers MS Access que nous allons découvrir un peu plus bas.

Nous choisissons des destinataires et leur envoyons le formulaire. Chez le destinataire cela donnera:



S'il ouvre le message:

## Vincent ISOZ

The screenshot shows an Outlook message window titled "Add Tâches Table Form" with the following details:

- From: Schulung 01
- To: Schulung 02
- Subject: Add Tâches Table Form
- Sent: 20.02.2012 14:27

The message body contains the text: "Fill out the form included in this message and send it back to me."

The form, titled "Add Tâches Table Form", includes the following fields:

- Titre:  (with a red asterisk and a note: "Type any combination of numbers and letters up to 255 characters.")
- Priorité:  (with a note: "Select an item from the drop-down list.")
- État:  (with a note: "Select an item from the drop-down list.")
- % achevé:  (with a note: "Type a numeric value.")
- Assigné à:  (with a note: "Select an item from the drop-down list.")
- Groupe de:  (with a note: "Select an item from the drop-down list.")

At the bottom of the form, there is a red circle icon and the text: "Form ID: urn:schemas-microsoft-com:office:infopath:2FD18929-6B50-48FB-AF27-209E1DA2457B:"

Nous remplissons et cliquons sur **Submit**. Apparaît alors la fenêtre suivante:

The dialog box is titled "Add Tâches Table Form AccessDataCollection:{7C871BE7-BD - Message" and contains the following text: "To submit your form to the following recipients from your default Microsoft Office Outlook e-mail account, click Send. To go back to editing, click Cancel."

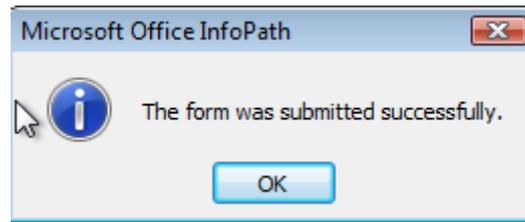
The dialog box includes the following fields:

- To:
- Cc:
- Bcc:
- Subject:

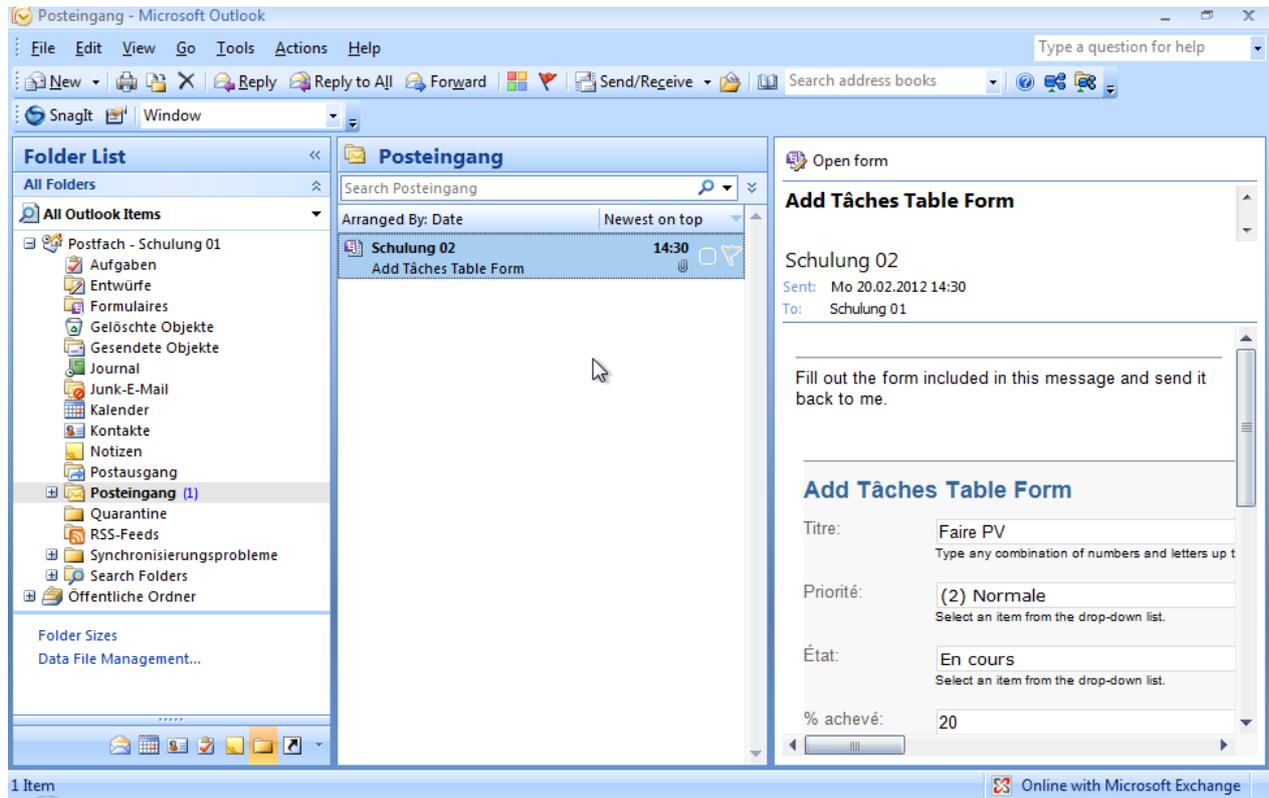
At the bottom of the dialog box, there are two buttons: "Send" and "Cancel".

Nous validons par **Send**. Viens alors:

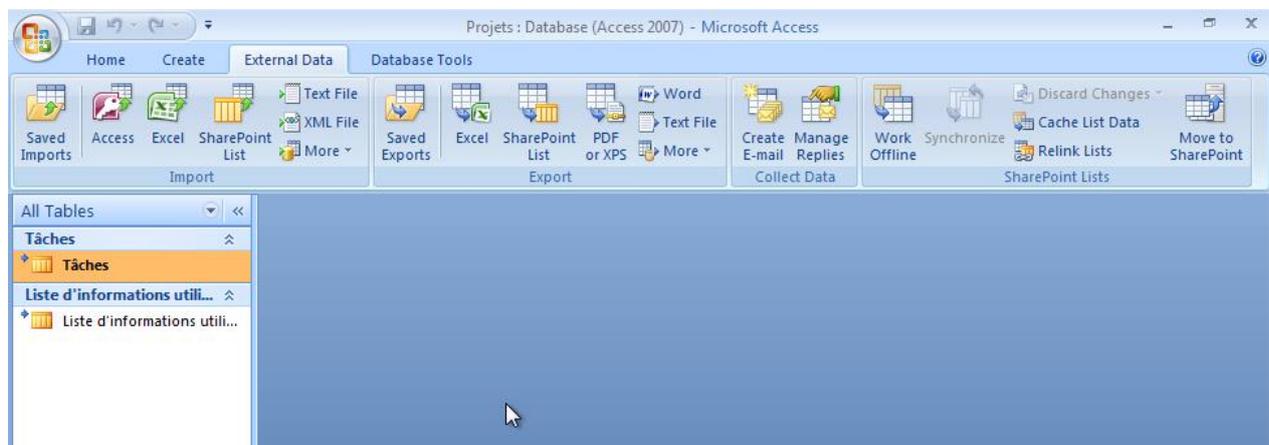
Vincent ISOZ



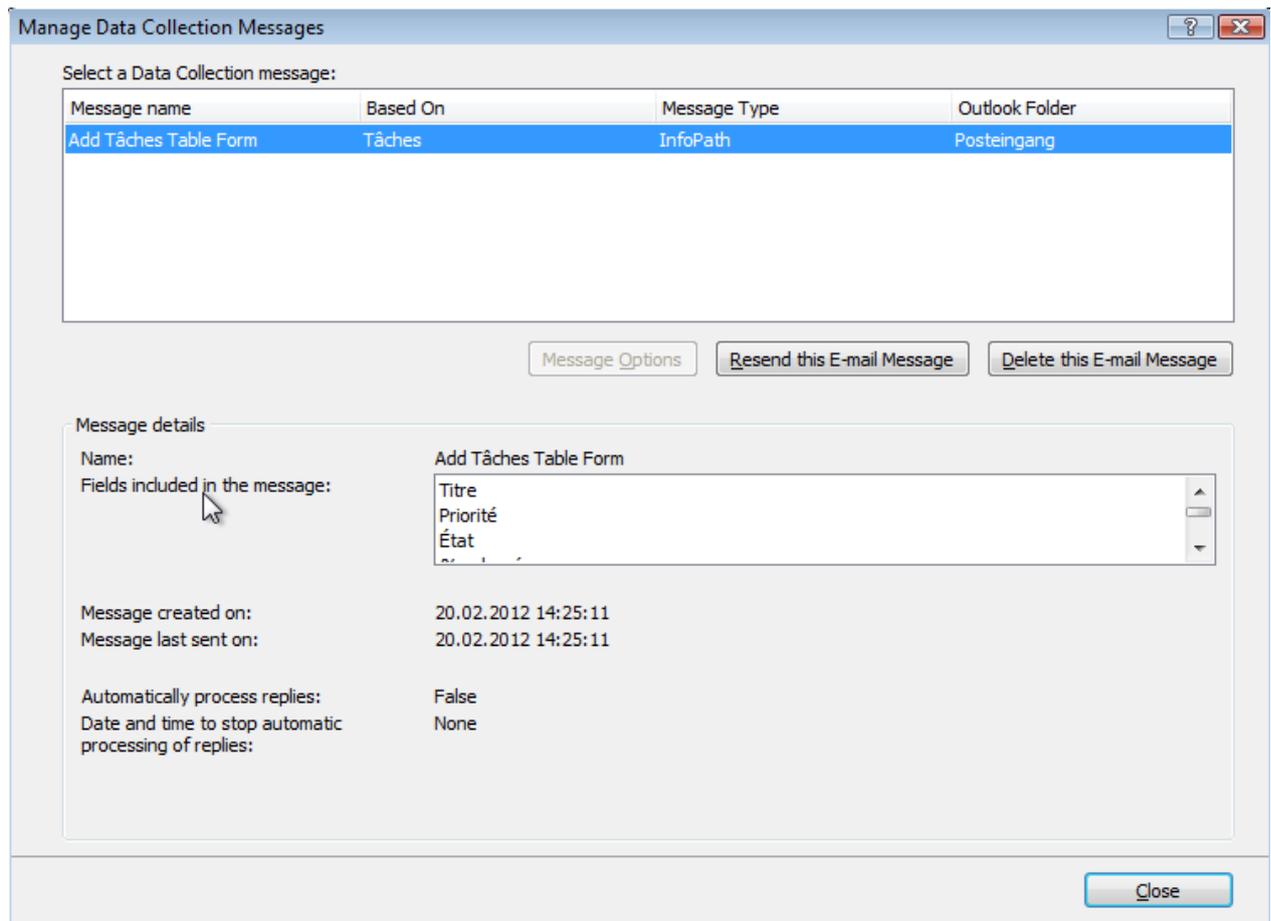
Chez l'expéditeur d'origine, nous avons alors:



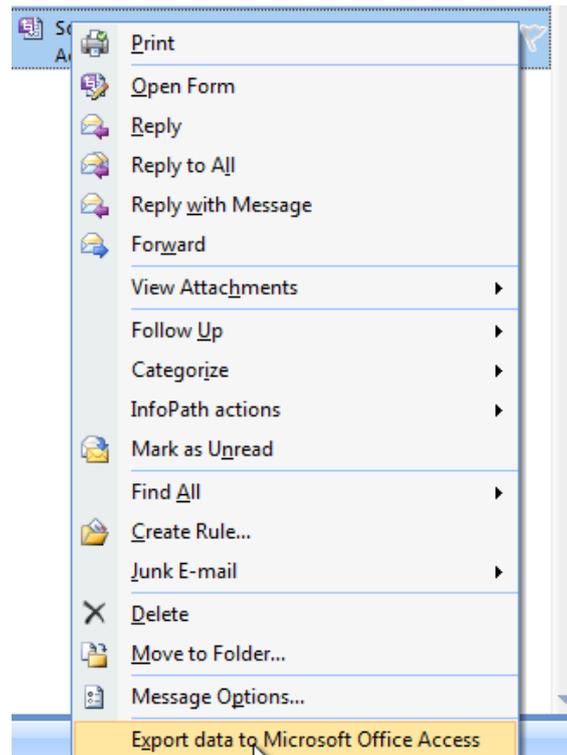
Pour importer dans MS Access, nous ouvrons la base et cliquons sur le bouton **Manage Replies** du groupe **Collect Data** du ruban **External Data**. En ayant pris soin bien évidemment de sélectionner la table *Tâches* au préalable:



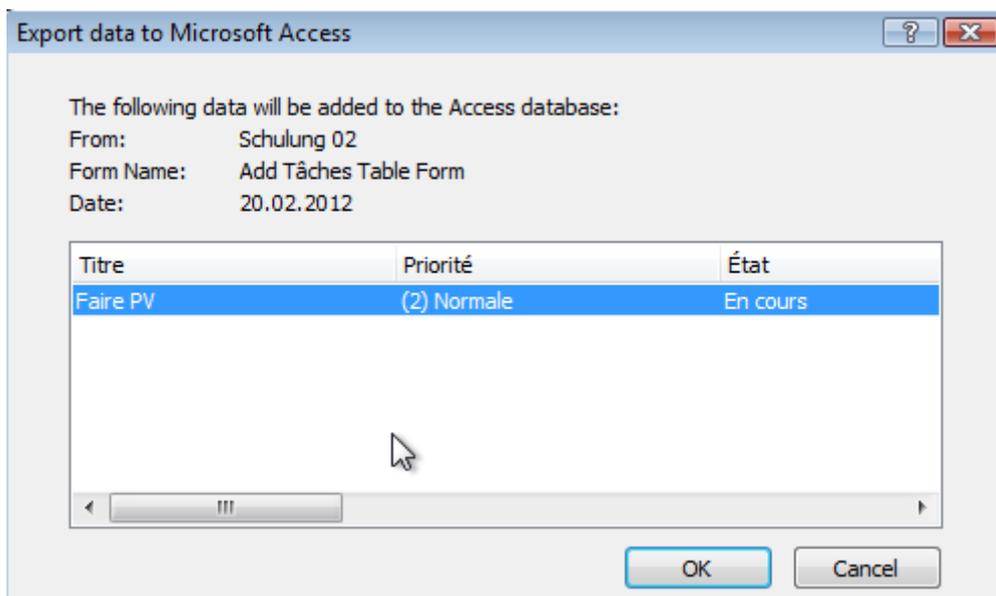
Il apparaît alors:



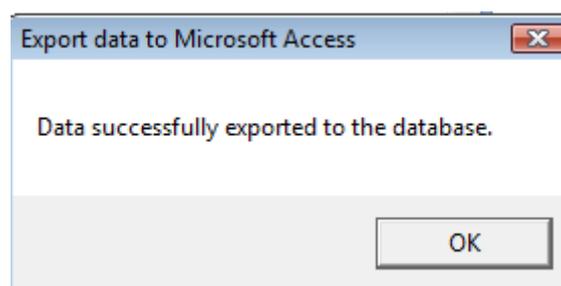
Nous voyons que contrairement au format HTML pour l'envoi mails, InfoPath n'est pas capable d'être traité en envoyé automatiquement de MS Outlook à Access. Il va donc falloir faire l'opération d'import à la main. Pour cela, dans MS Outlook, nous faisons un clic droit sur le mail que nous avons reçu:



et nous cliquons sur **Export data to Microsoft Office Access**. Nous avons alors:

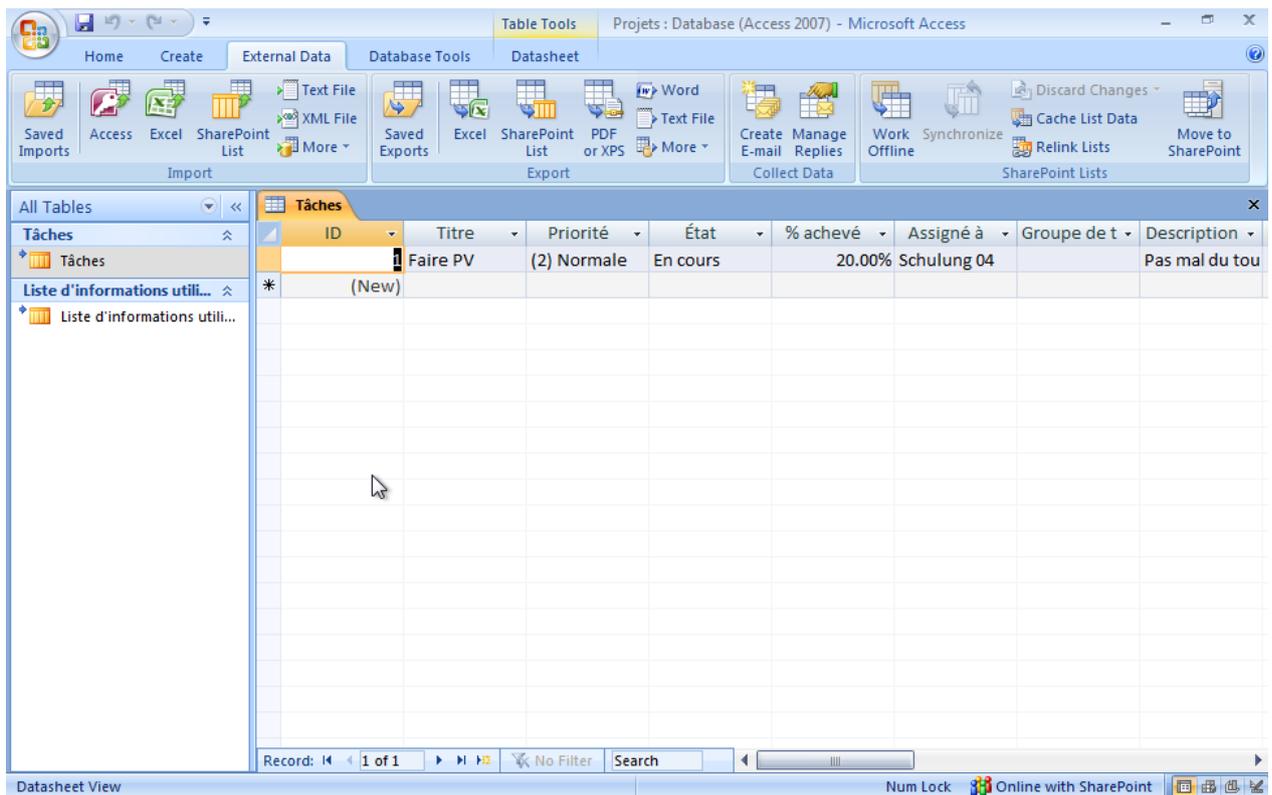


Nous validons par **OK**. Et si toutes les conditions sont remplies, nous avons:

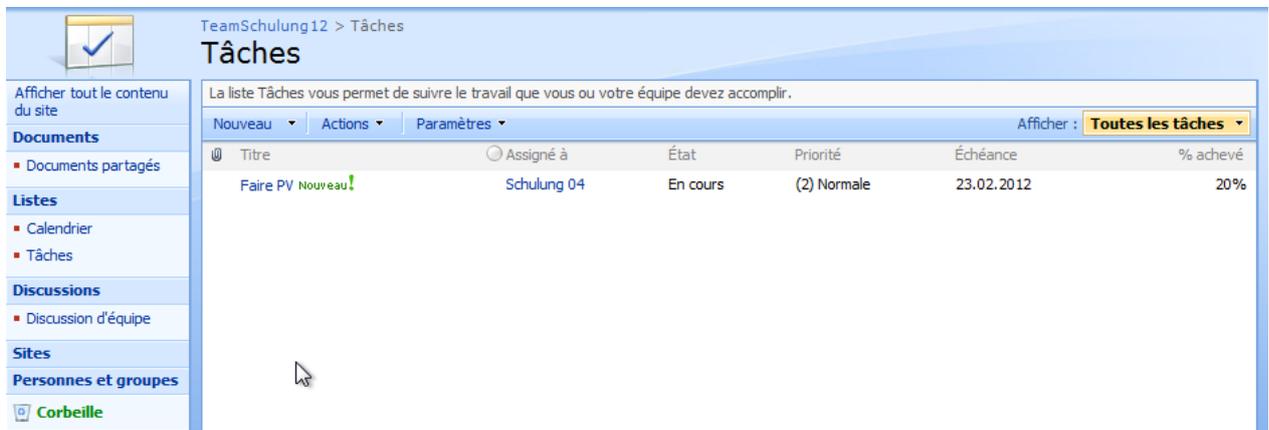


Vincent ISOZ

Allons vérifier dans MS Access:



Pas mal du tout! Et finissons en contrôlant dans SharePoint:

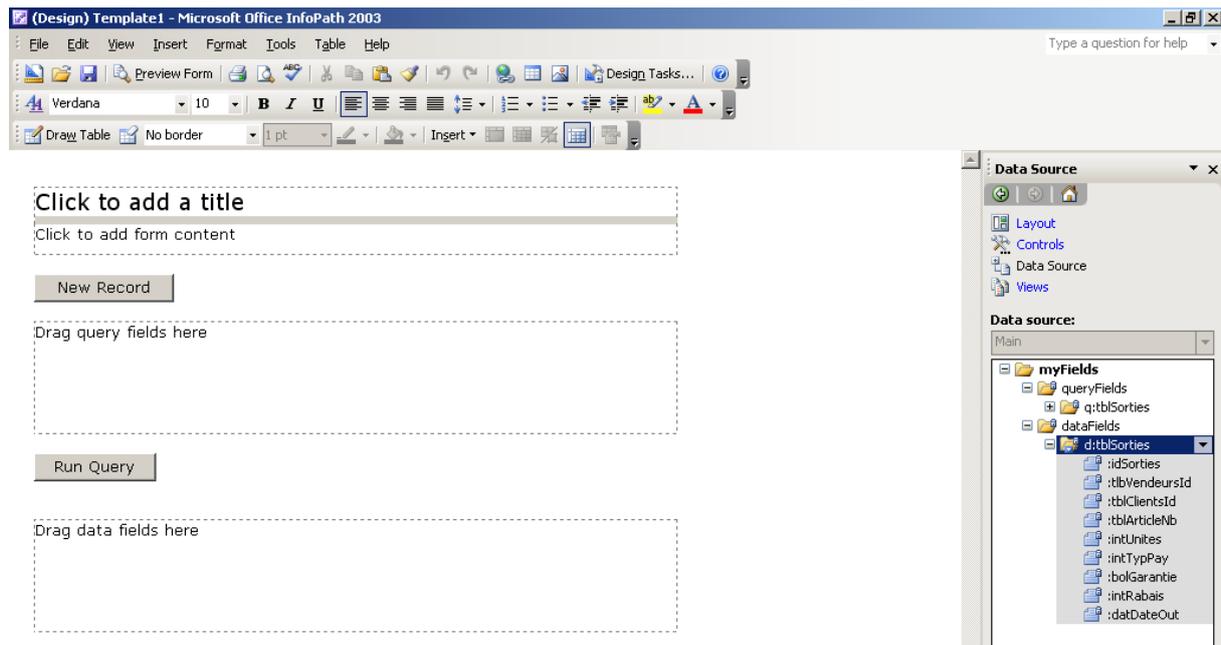


Vraiment pas mal du tout!

## 9.9 Formulaire de requête connecté (MS Access)

Nous allons dans cet exemple repartir d'un formulaire vierge, connecté à la même base de données mais cette fois-ci à la table *tblSorties* pour faire des requêtes dans le formulaires (bien que ce soit stupide puisqu'on peut faire beaucoup mieux directement depuis MS Access et de toute façon les connexions à MS Access ne fonctionnent pas avec InfoPath Services)

Pour faire ceci, le début est identique au début de l'exemple précédant pour arriver au résultat suivant (attention!!! l'esthétique ne nous intéresse pas dans cet exemple):



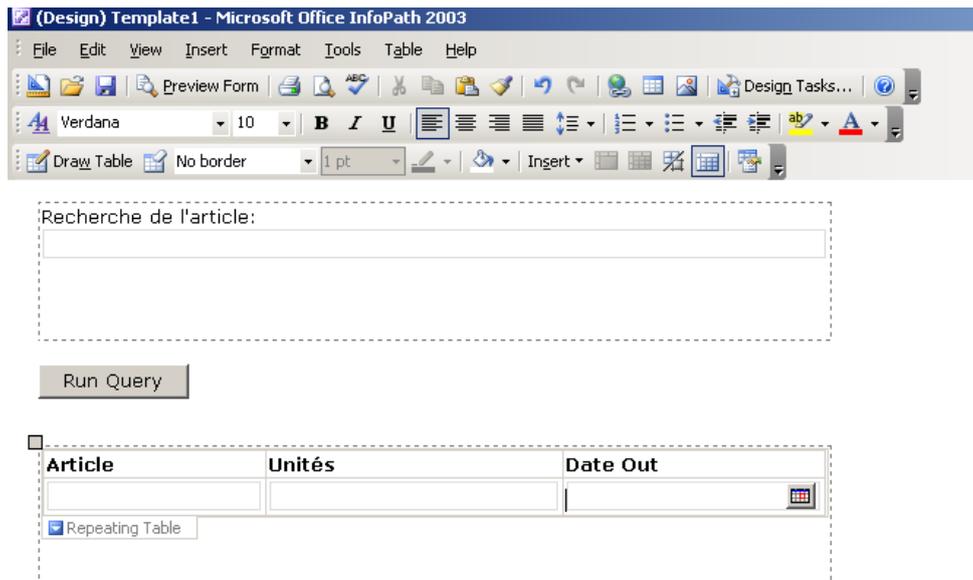
Maintenant:

1. Supprimez le bouton *New Record* ainsi que la zone de titres
2. Glissez et déplacez les champs *d:tblSorties* dans la zone *Drag data fields here*
3. Insérez les données en tant que *Repeating table*
4. Enlevez les données insérées précédemment qui ne sont pas pertinentes relativement à vos besoins
5. Insérez dans la zone *Drag query fields here* le champ *tblArticleNb* de la datasource *queryFields/q:tblSorties*

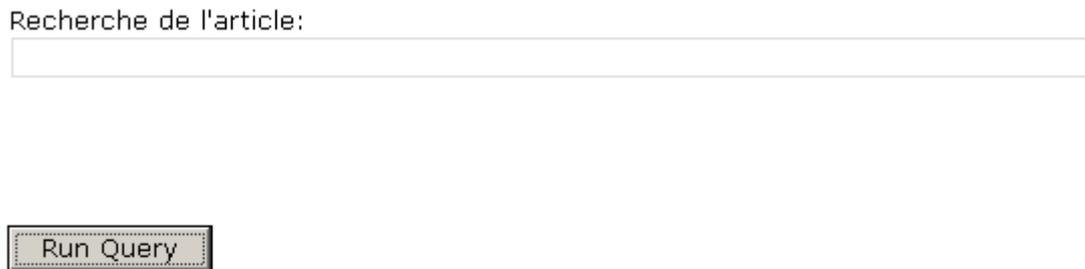
Remarque: Si vous le souhaitez, vous pouvez à la place de champ, insérer un contrôle de type *ListBox*. Ensuite, MS InfoPath vous demandera automatiquement à quel *queryFields* il doit être lié. Ne reste ensuite plus qu'à peupler la *ListBox* avec la source de données souhaitée.

6. Renommez et formatez les champs comme il vous plaît

Résultat à obtenir:

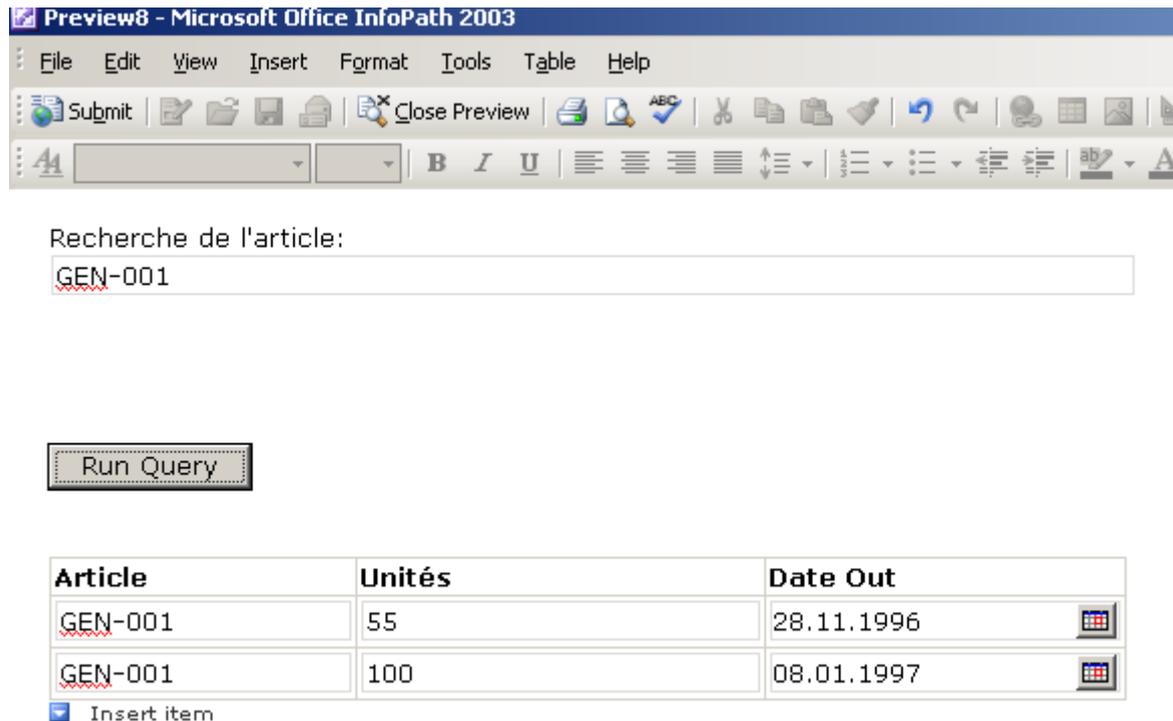


Allez ensuite en mode *Preview Form* et cliquez sur *Run Query*. Vous aurez alors:

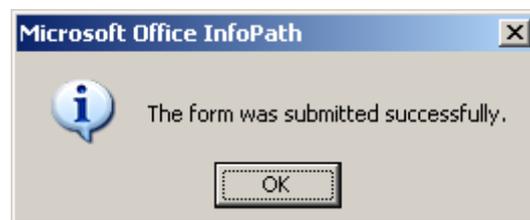


Article	Unités	Date Out
GEN-001	55	28.11.1996
GEN-003	10	10.12.1996
INF-002	1'500	10.12.1996
GEN-002	100	11.12.1996
GEN-003	20	11.12.1996
INF-002	500	11.12.1996
INF-003	30	11.12.1996
GEN-002	50	18.12.1996
INF-001	5	18.12.1996
INF-003	10	18.12.1996
GEN-004	80	30.12.1996
INF-001	15	30.12.1996

et si vous saisissez un valeur dans le champ de recherche et cliquez ensuite sur *Run Query* vous aurez par exemple (InfoPath filtre automatiquement la requête):



Si maintenant vous changez la quantité de GEN-001 de la valeur 100 à 110 et cliquez sur le bouton *Submit* dans la barre d'outils InfoPath vous aurez:



et la mise à jour aura été effectué dans MS Access.

**Remarque:** Tous les champs que vous ajouterez dans la zone supérieure du formulaire sont reliés par un ET en ne peuvent contenir de relations d'ordre (>, <, <>, =, LIKE, ...). Pour utiliser ce genre de possibilités il vous faudra passer par du code.

### 9.10 Désactivation conditionnelle de boutons

Souvent les personnes cherchent à avoir un bouton de soumission de formulaire qui apparaît à l'écran que lorsque tous les champs du formulaire sont remplis. Ceci peut être fait en utilisant le formatage conditionnel.

Dans cet exemple nous allons créer un formulaire simple qui requière que tous les champs soient remplis pour que le bouton de soumission apparaisse à l'écran.

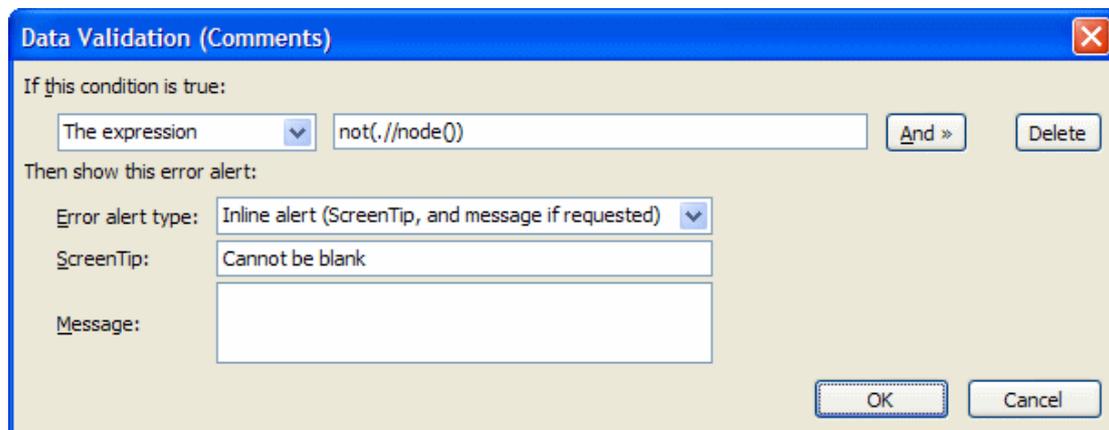
Pour cela, dans un formulaire vide nous ajoutons un champ de texte nommé *Name* avec la propriété *Cannot be blank*, deux champs de textes nommées *Address* et *Phone* et un champs de texte nommé *Email* avec la propriété *Cannot be blank*. Nous ajoutons un zone de texte enrichie que nous nommerons *Comments*:

Name:   
Adress:   
Phone :   
E-mail:   
Comments:

Nous ne pouvons pas spécifier sur un champ de type texte enrichi la propriété *Cannot be blank*. Nous devons effectuer cette manipulation manuellement en utilisant l'option *Data validation*.

Pour ce faire:

1. Dans le panneau de source de données sélectionnez *Comments*
2. Sur l'onglet *Validation And Event Handlers* cliquez sur *Field Or Group Properties* et cliquez sur *Add*
3. Sélectionnez *The Expression* dans la première liste déroulante et saisissez l'expression *XPath* suivante *not(//node())*. Vous aurez alors:



Validez par *OK*.

Ensuite, sélectionnez l'ensemble des champs et étiquettes et après dans le panneau des sources de données faites un clic droit sur *myFields* et choisissez *Controls In Layout Tabel*

Vous aurez alors quelque chose de plus structuré:

Name	<input type="text"/>
Adress	<input type="text"/>
Phone	<input type="text"/>
Email	<input type="text"/>
Comments	<input type="text"/>

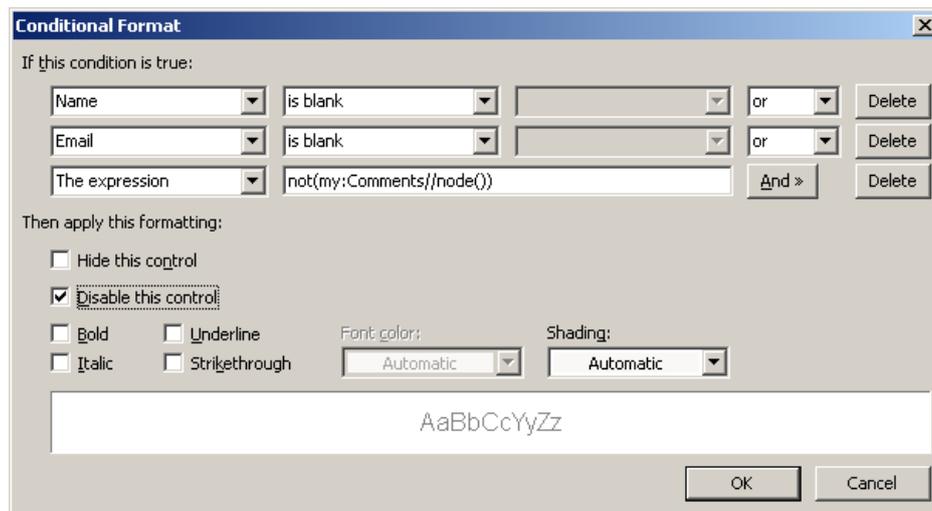
Ensuite, ajoutez un contrôle de type *Button* dans le formulaire et double cliquez sur le bouton.

Pour le *Label* saisissez *Submit* et ensuite changez l'*ID* à *btnSubmit*.

Cliquez ensuite sur *Rules* et cliquez sur *Add*. Nommez la règle *Submit* et cliquez sur *Add Action*. Ensuite sélection *Show A Dialog Box Message* de la liste déroulante. Saisissez *Form Submitted* et cliquez trois fois sur *OK*.

Dans l'onglet *Display* des propriétés du bouton sélectionnez *Conditional Formatting* et cliquez sur *Add*.

Et faites en sorte d'avoir:



Validez par *OK* et testez votre formulaire.

Le bouton *Submit* ne devra s'activer que si *Name*, *E-Mail* et *Comment* ont un contenu!

### 9.11 Assignation d'une numérotation automatique à un champ

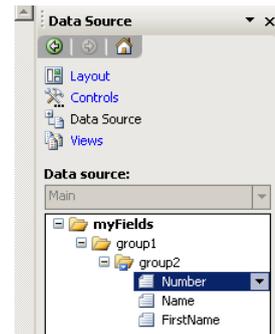
Si vous souhaitez afficher un numéro de ligne dans une table répétitive vous pouvez utiliser la boîte d'expression avec la commande *XPath position()*.

Il s'agit d'une manière élégante d'afficher le numéro de ligne d'une table répétitive.

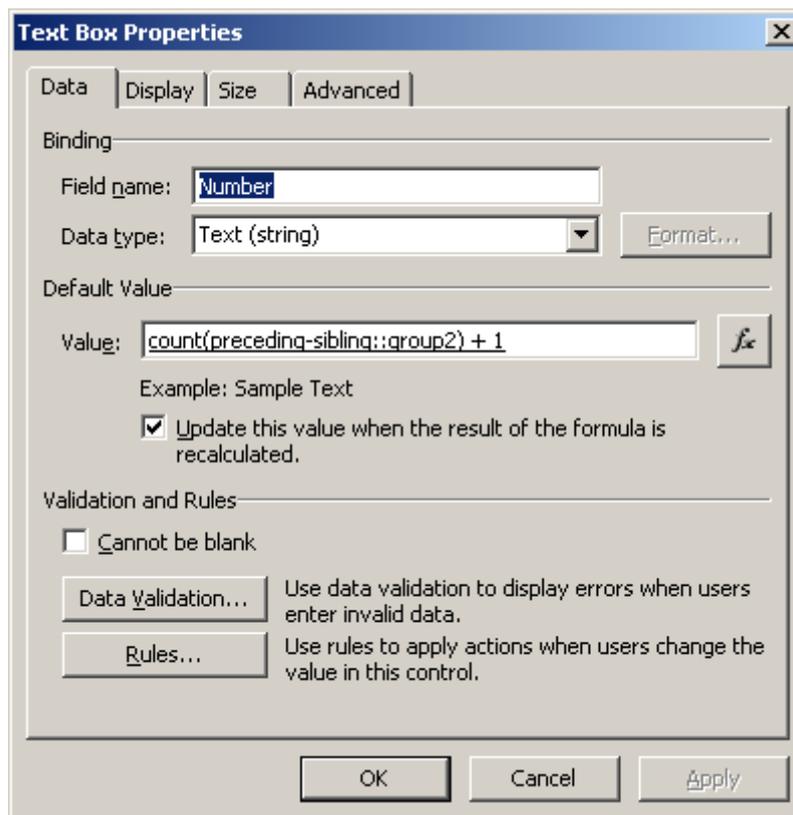
Pour cet exemple, nous allons créer la petite structure suivante comportant une table répétitive avec trois champs:

N°	FirstName	Name

Repeating Table



Ensuite, dans les propriétés du champ *Number* nous allons saisir:

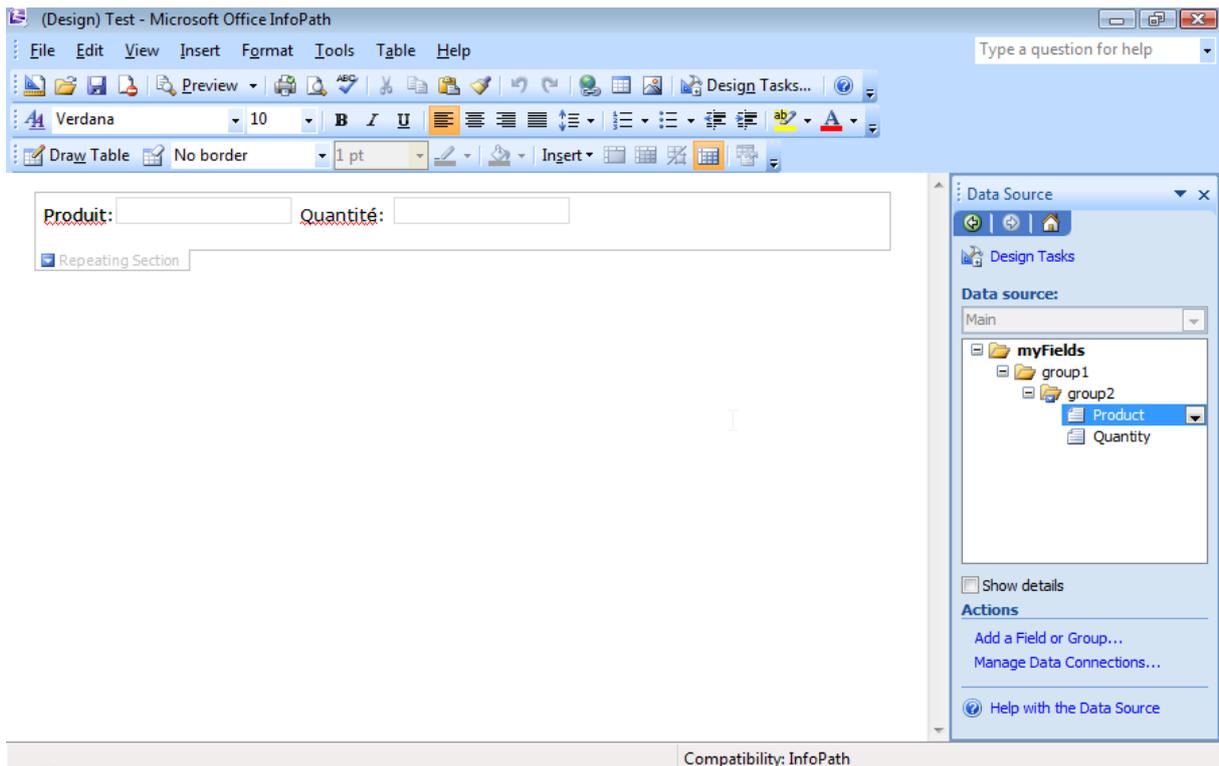


Maintenant testez votre formulaire et vous verrez que le champ *N°* s'incrémente automatiquement!

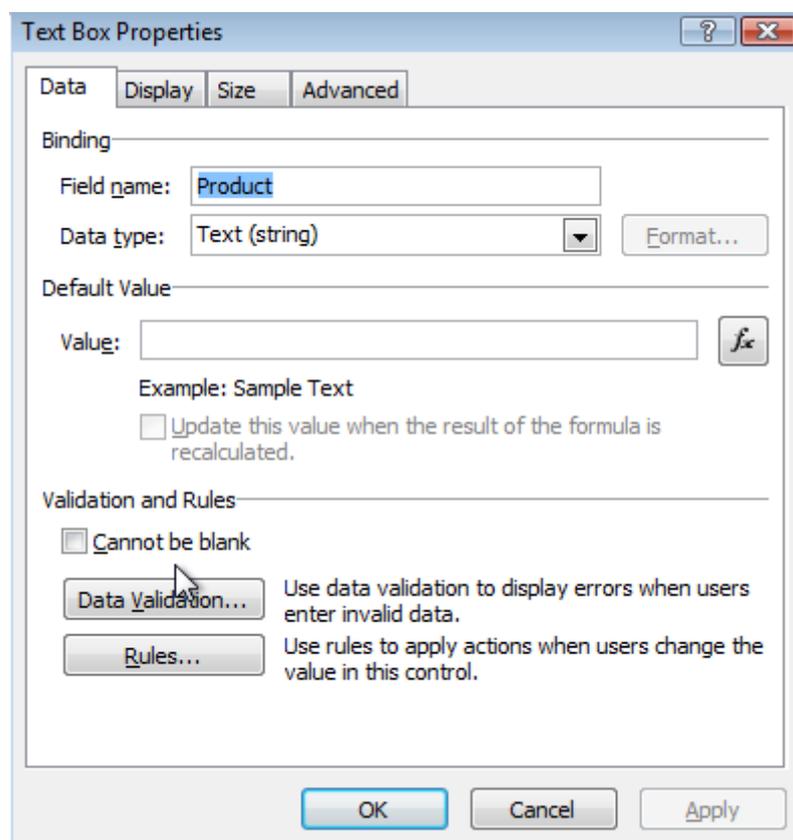
### ***9.12 Interdire la saisie de doublons sur un champ***

Dans des formulaires de commande ou d'inscription à des cours, il y a très souvent des sections répétitives. Une demande fréquente est d'interdire à l'utilisateur la possibilité d'ajouter deux fois une même ligne dans une section répétitive basé sur le contrôle de l'unicité d'un champ.

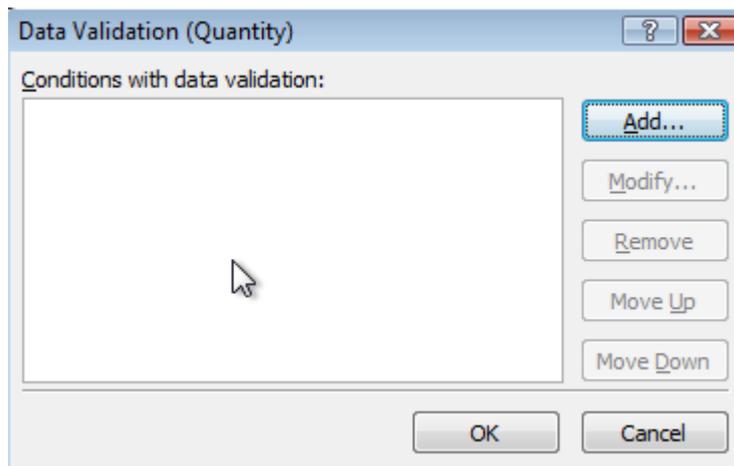
Imaginons pour faire simple le cas suivant:



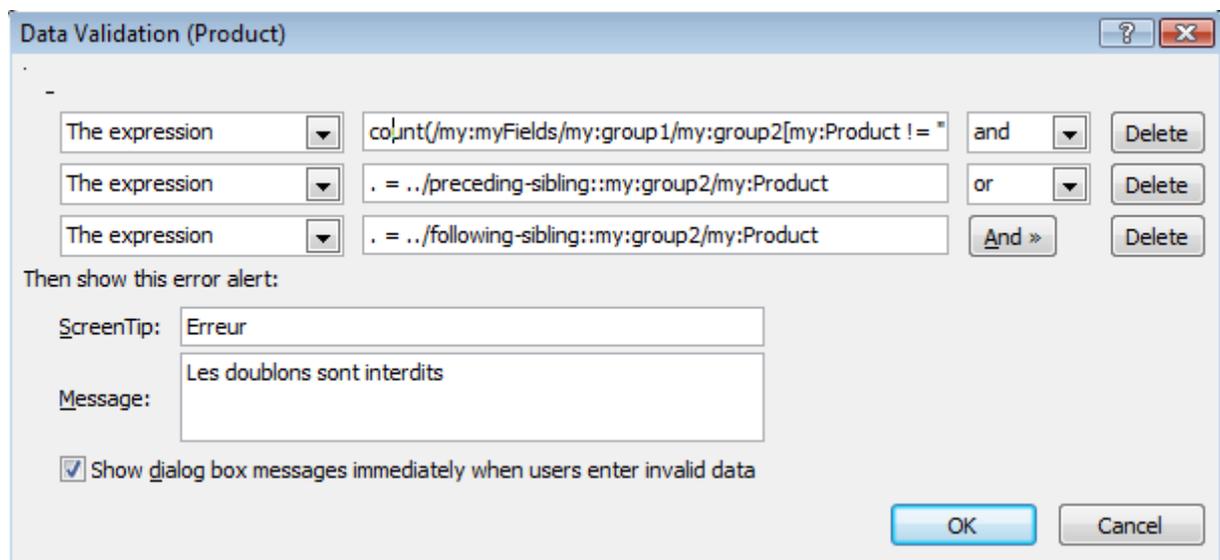
Nous souhaiterions donc interdire le fait d'ajouter deux fois le même produit. Pour cela, nous double cliquons sur le champ *Produit*:



et nous cliquons sur *Data Validation...*:



Nous cliquons sur *Add...* pour mettre les requêtes XPath suivantes:



où à la première ligne nous avons en entier:

```
count(/my:myFields/my:group1/my:group2[my:Product != ""]) > 1
```

### 9.13 Utilisation des rôles

Nous avons créé au début de ce cours (dont le support est la base) un formulaire de notes de frais.

Ce formulaire a deux vues, une pour les employés et l'autre pour les gestionnaires.

Pour associer la bonne vue avec les bons utilisateurs il faut créer des rôles (*Roles*) dans le formulaire *InfoPath*.

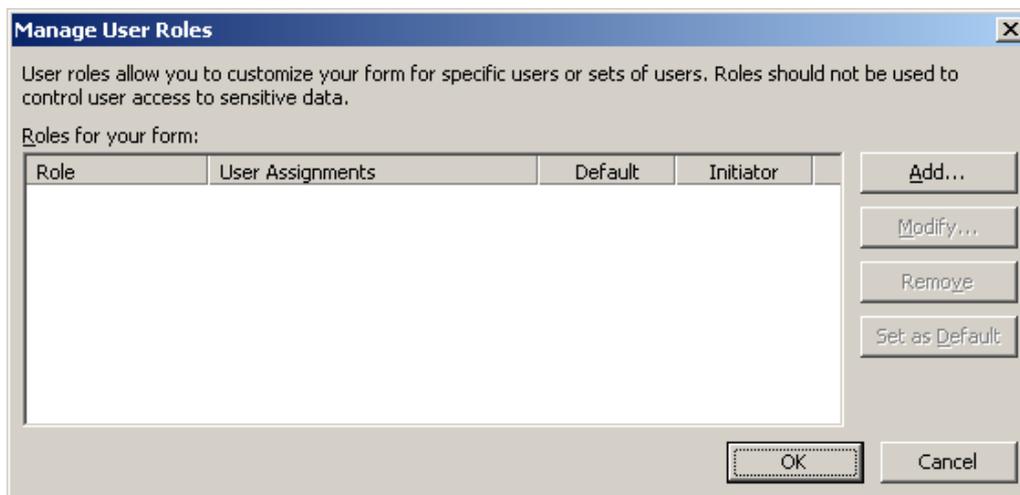
Remarque: Pour utiliser cette fonctionnalité, les groupes doivent être créés dans Active Directory (AD) et le développeur des formulaires d'InfoPath connecté au domaine.

**Attention!!! Cette fonctionnalité ne marche pas avec InfoPath Form Services!!!**

Vincent ISOZ

Voici la procédure à suivre pour que chaque groupe ait sa vue respective:

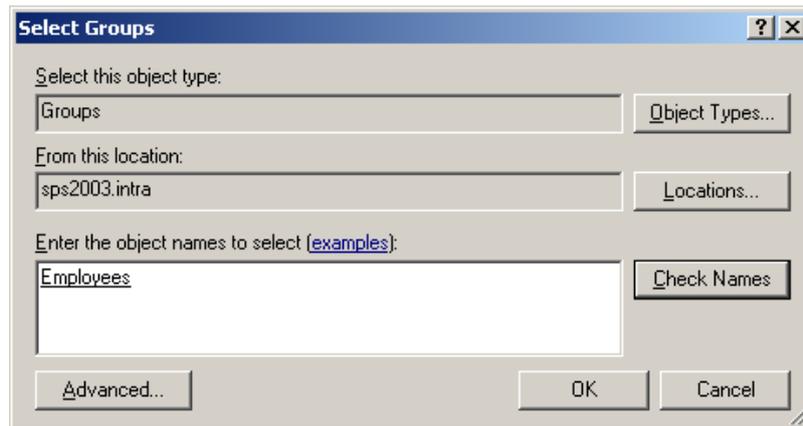
1. Le formulaire en mode *Design* allez dans *Tools/User Roles*



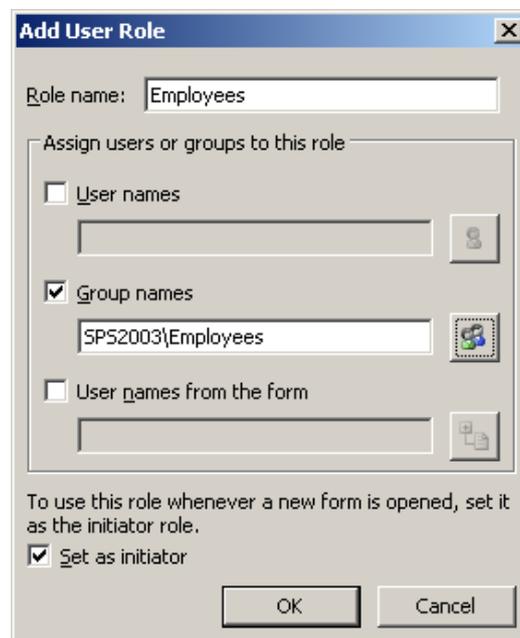
2. Dans la boîte de dialogue *Manage User Roles* cliquez sur *Add*



3. Dans la boîte de dialogue *Role name* saisissez *Employee* et ensuite cochez *Group names* ainsi que *Set as initiator* et cliquez sur le bouton avec les deux personnages puis faites une recherche de votre group dans l'AD:

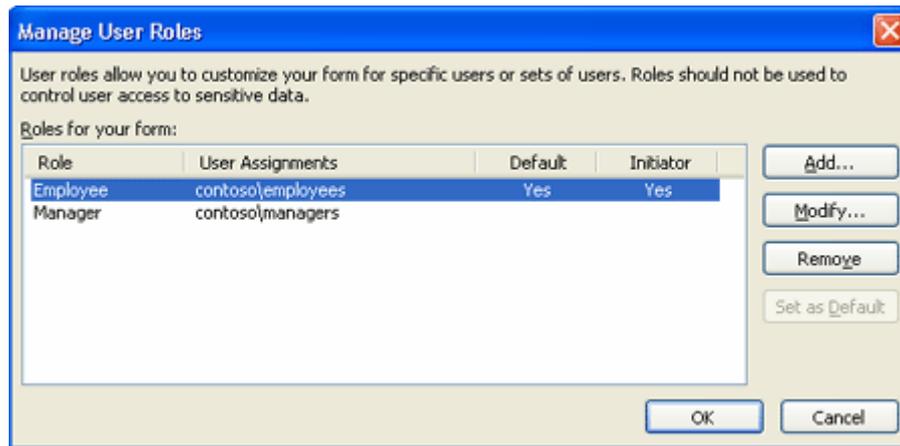


4. Cliquez sur *OK*



5. Ensuite, faites de même pour les *Managers* sans activer *Set as initiator*

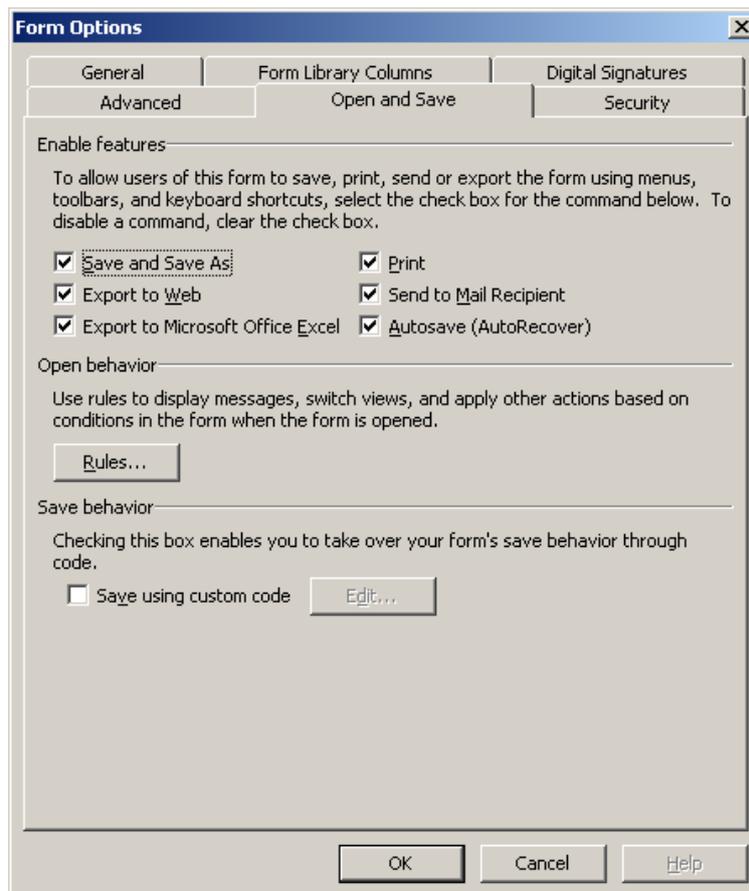
Vous devriez avoir le résultat suivant dans la boîte de dialogue *Manager User Roles* (nom de domaines et groupes à part bien entendu...):



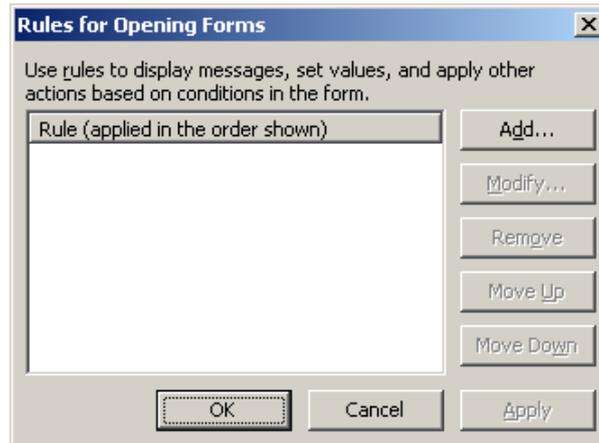
Maintenant que nous avons créé les rôles nous allons dire à InfoPath ce qu'il doit faire avec.

Pour utiliser les rôles dans le formulaire effectuez les étapes suivantes:

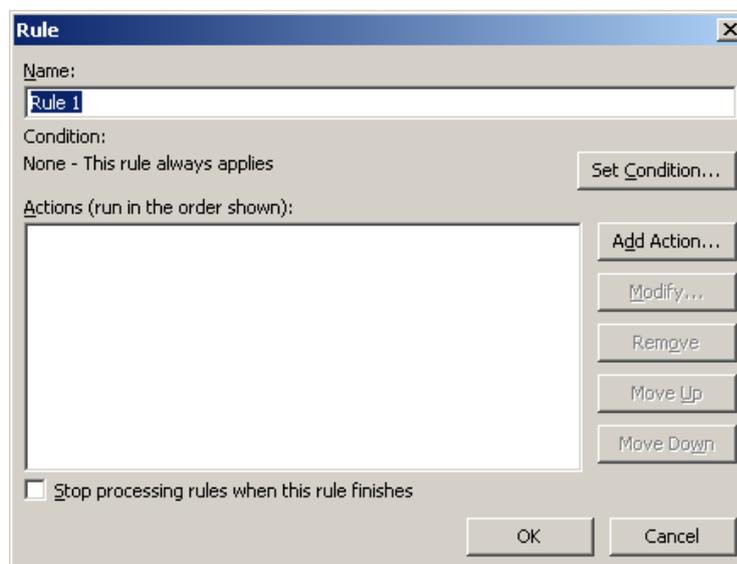
1. Allez dans *Tools/Form Options*
2. Dans la boîte de dialogue *Form Options*, cliquez sur *Open and Save* et ensuite cliquez sur *Rules*.



3. Dans la boîte de dialogue *Rules for Opening Forms* cliquez sur *Add*



4. Dans la boîte de dialogue *Rule* cliquez sur *Set Condition*

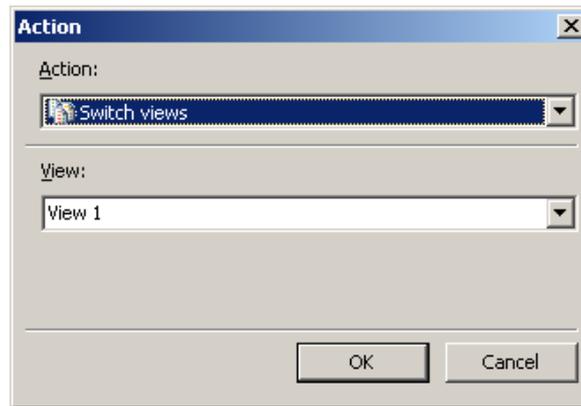


5. Dans la boîte de dialogue *Condition* faites en sorte d'avoir:



et ensuite cliquez sur *OK*.

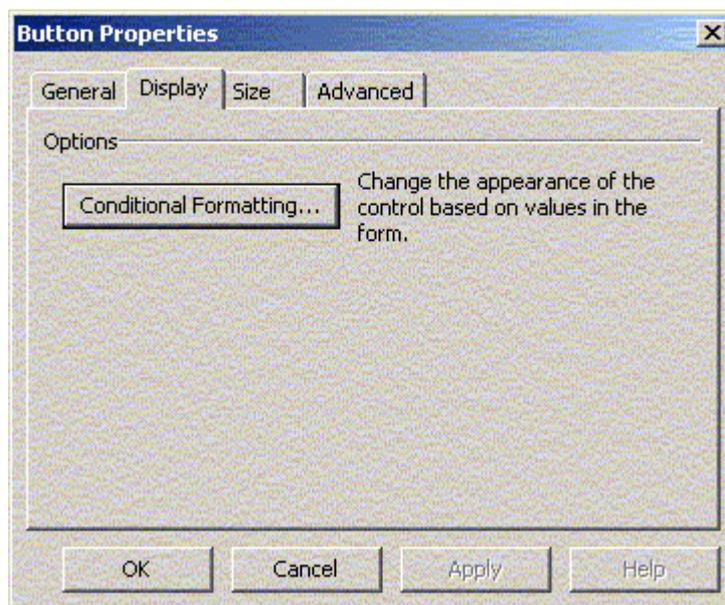
6. De retour dans la boîte de dialogue *Rule*, cliquez sur *Add Action* et choisissez l'action *Switch Views* et le nom de la vue qui devra être activée lorsque que les employés ouvriront le formulaire.



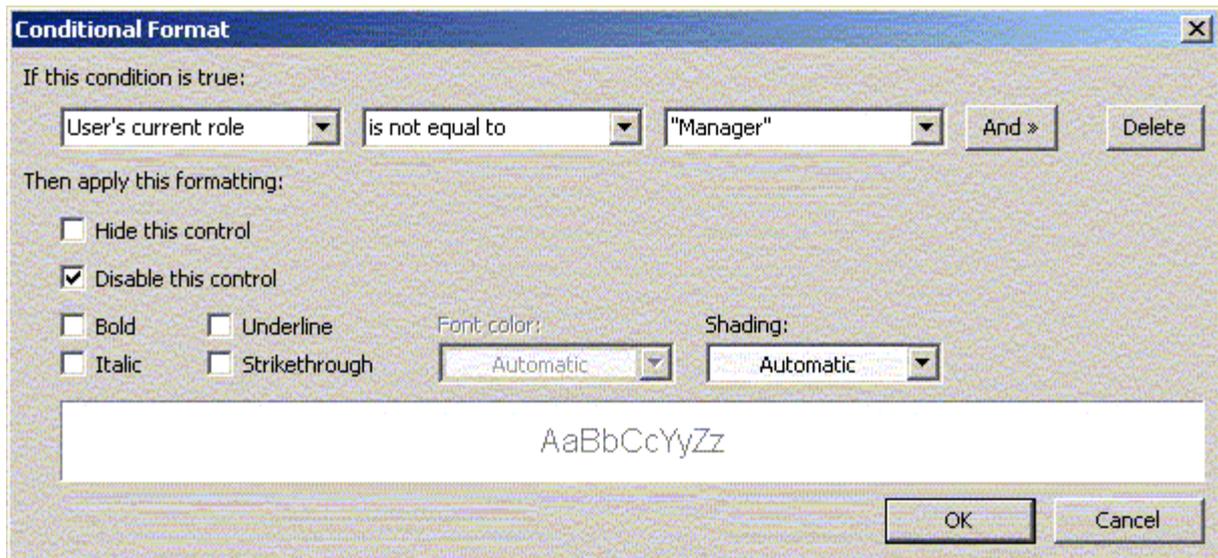
et ensuite validez par *OK*.

7. Pour ajouter une règle pour les *Managers* il suffit de procéder de manière similaire.

Autre exemple, si nous créons un bouton "Approuver" dans un formulaire, ce bouton peut être rendu visible pour les managers uniquement. Nous utilisons pour cela la mise en forme conditionnelle (bouton *Conditional Formatting* de l'onglet *Display* de la page de propriétés du bouton):



Puis nous exprimons que le bouton est désactivé pour les non-managers:

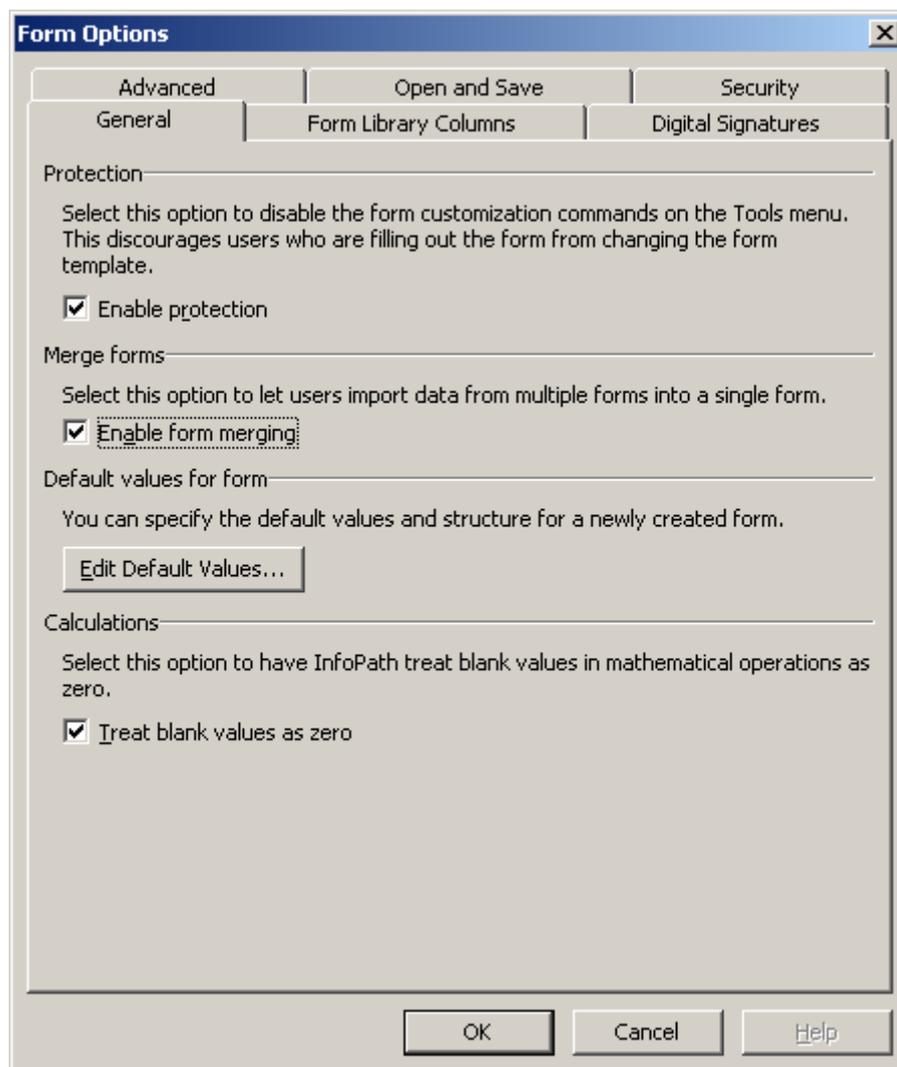


Remarquez le nombre de possibilités offertes par cette fenêtre ; on aurait tout aussi bien pu cacher purement et simplement le bouton.

## 9.14 Protection contre la modification

Pour prévenir les utilisateurs de la modification involontaire du design du formulaire, vous pouvez le protéger.

Pour cela il suffit d'aller dans *Tools/Form Options* et dans l'onglet *General* sous la zone *Protection* d'activer *Enable protection*

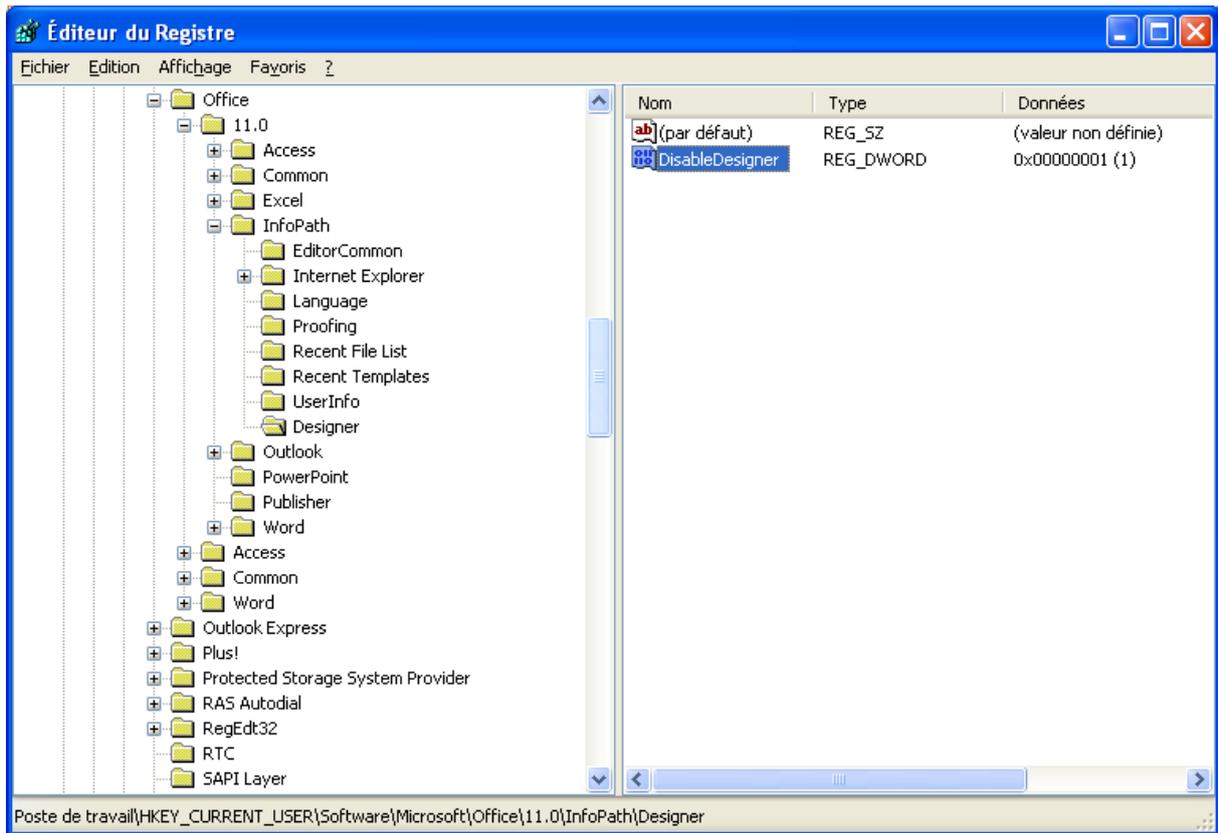


Le problème avec cette méthode est que l'utilisateur arrivera toujours à ouvrir le formulaire en mode *Design* par un clic droit de la souris sur le fichier *xsn*.

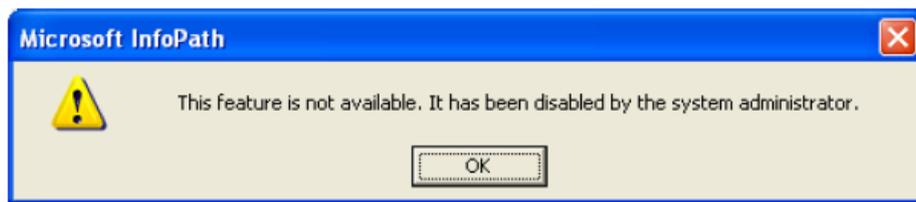
Pour bloquer cela il faudra aller dans la base de registres du profil utilisateur et sous:

HKEY\_CURRENT\_USER\SOFTWARE\MICROSOFT\OFFICE\11.0\InfoPat

et créer une *sub-key* nommée *Designer* et ensuite une entrée *DWORD* nommée *DisableDesigner* en mettant sa valeur à 1.



Ensuite, si l'utilisateur essaie d'ouvrir le formulaire en mode *Design*, il aura à l'écran:

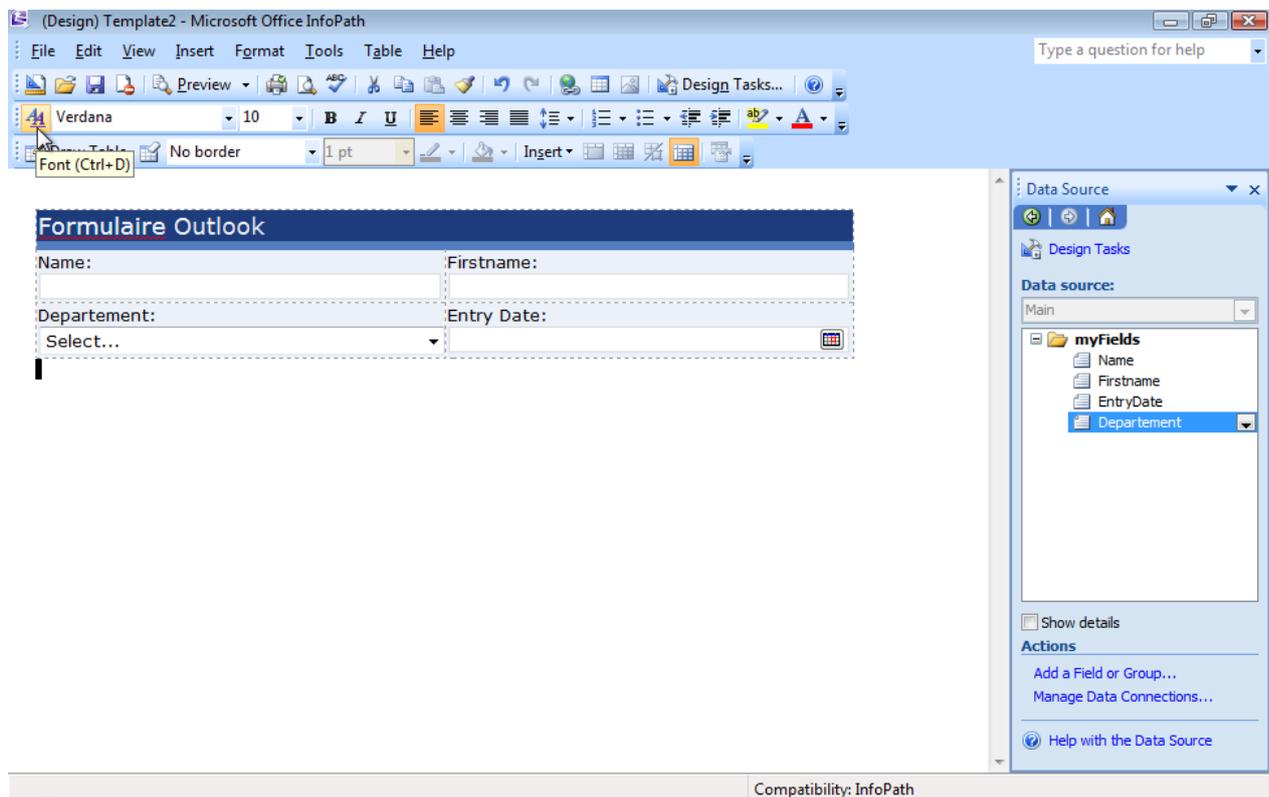


Vincent ISOZ

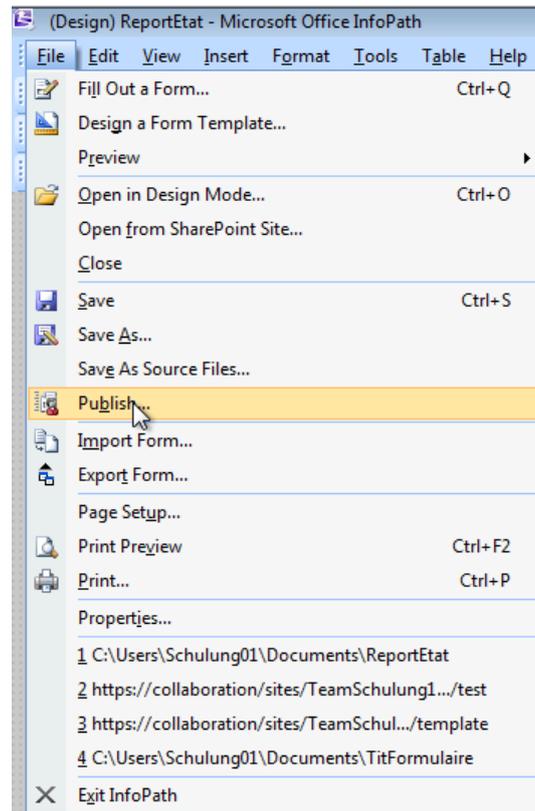
## 9.15 Utilisation d'InfoPath avec Outlook

Si votre entreprise possède MS Outlook 2003/2007 ou 2010 et ultérieur, vous pouvez partager un formulaire avec d'autres utilisateurs en l'envoyant par e-mail.

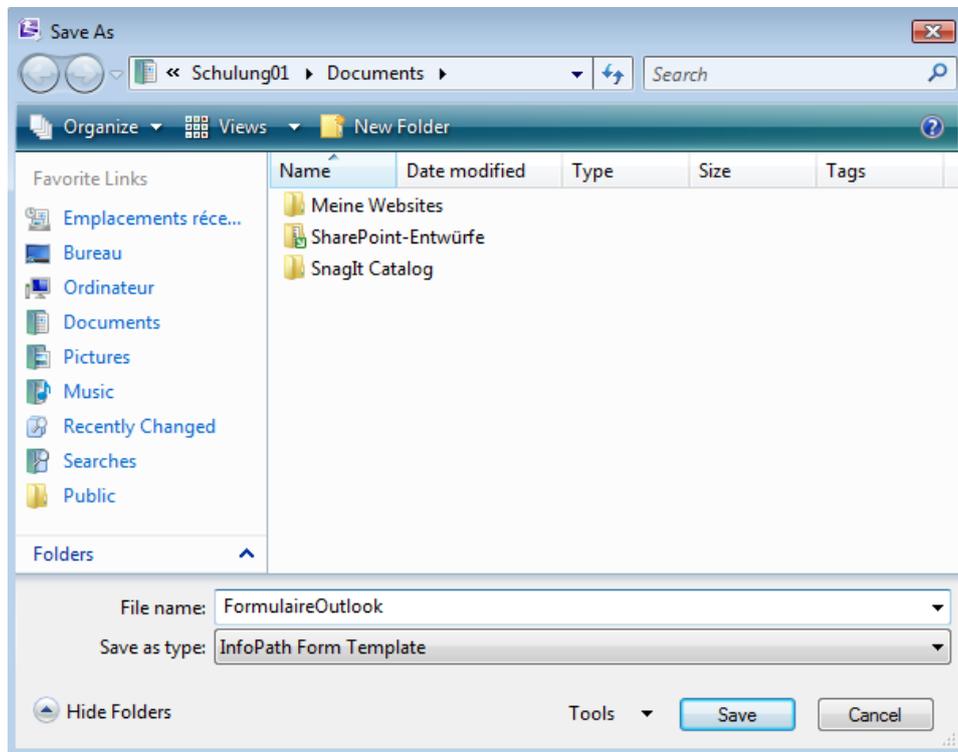
L'utilisateur possédant InfoPath peut même remplir son formulaire directement dans le message Outlook et l'envoyer pour lecture dans le message avec en pièce jointe, le fichier XML contenant les données. Voyons cela en créant un formulaire simple du type suivant:



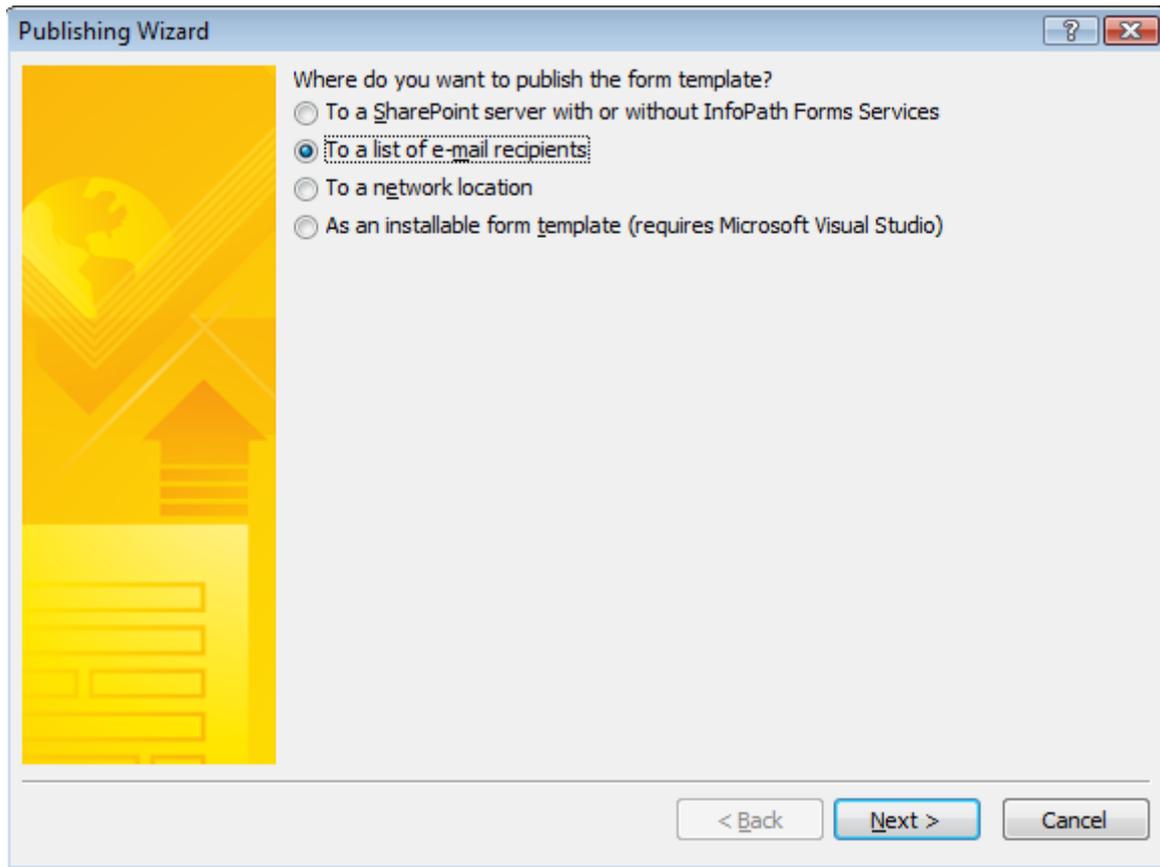
Nous le publions:



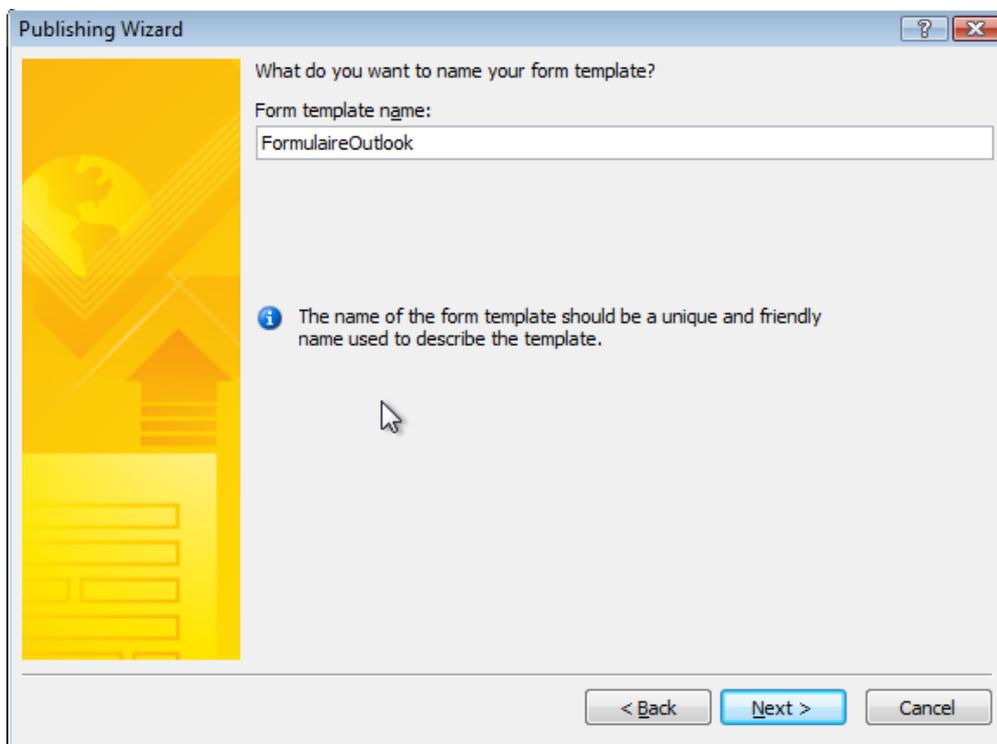
Le logiciel va nous demander d'enregistrer, nous nous exécutons:



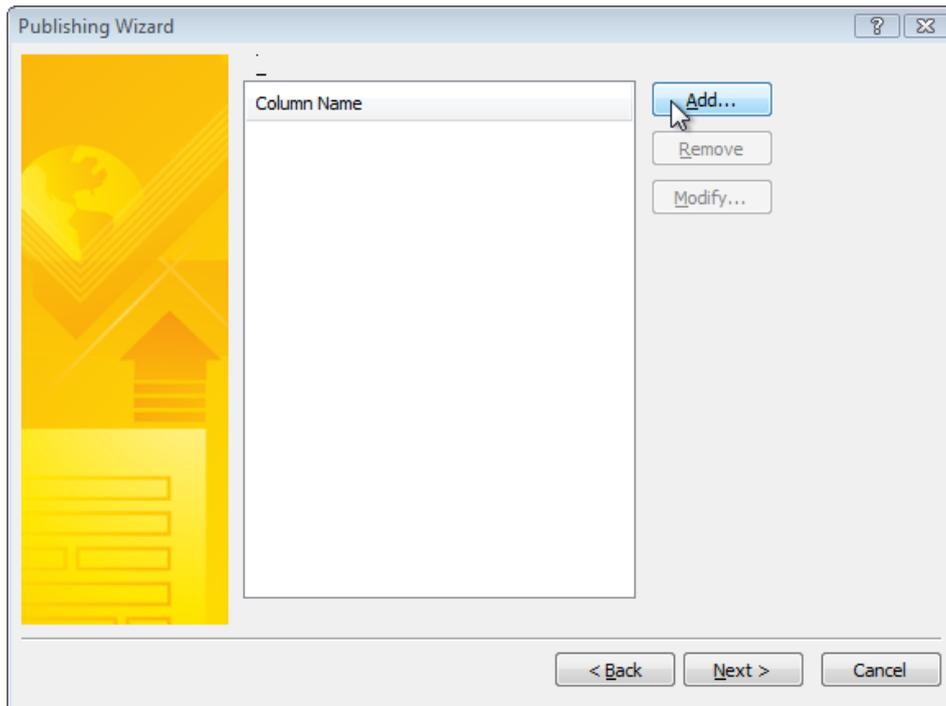
Une fois l'enregistrement validé, l'assistant continue:



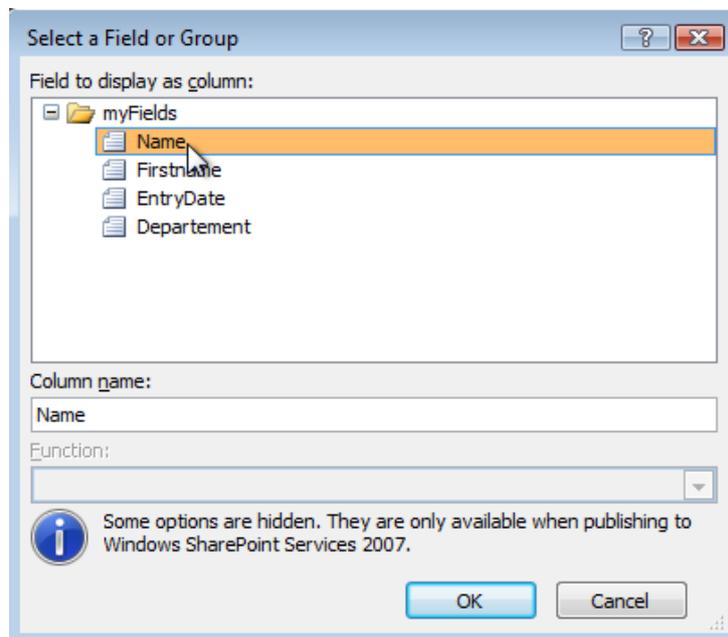
Nous choisissons bien évidemment **To a list of e-mail recipients** et cliquons sur **Next**:



Nous tapons un nom pour le formulaire et cliquons sur **Next**:

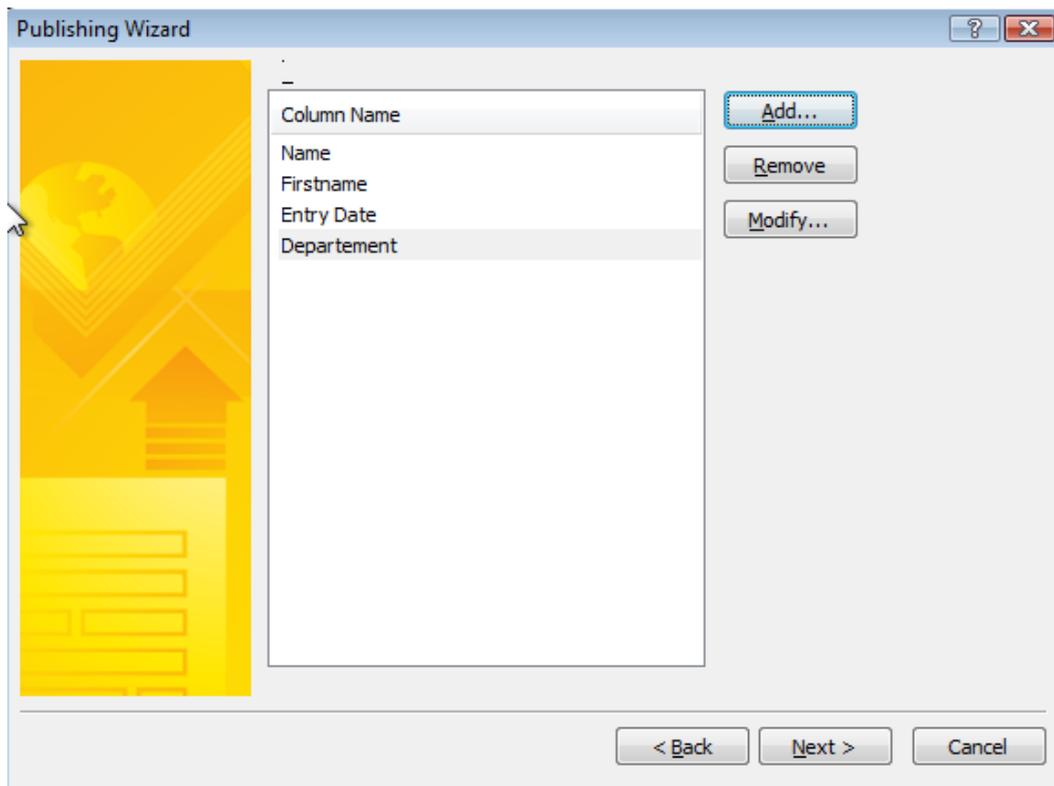


Nous ajoutons tous les champs un par un pour les avoir dans la vue Outlook (quand cela marche...):

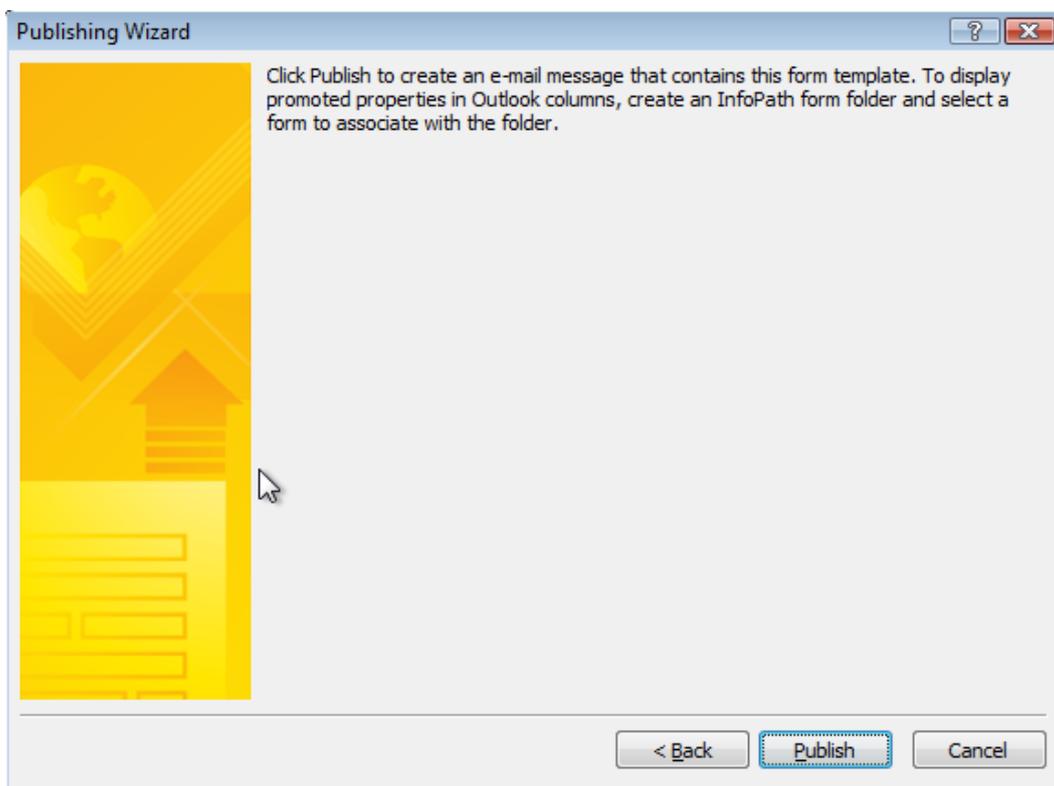


ce qui donnera au final:

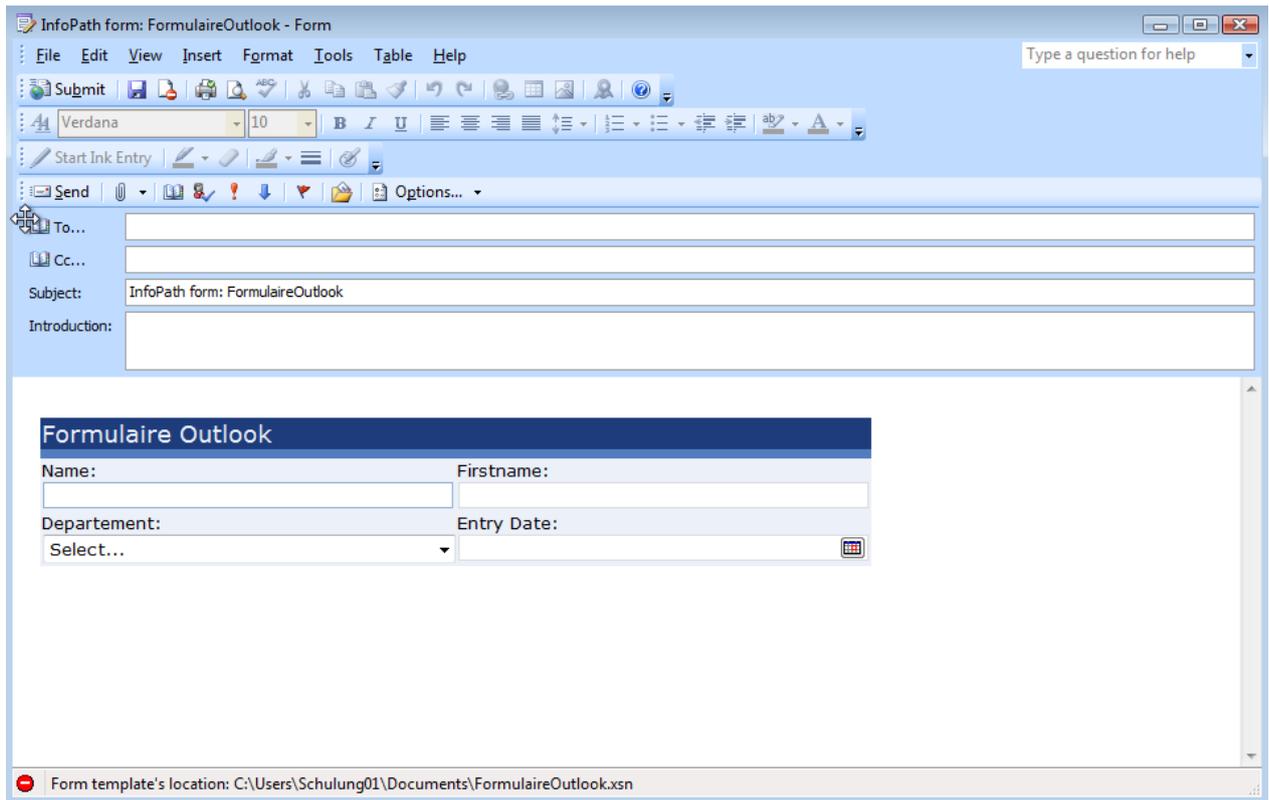
Vincent ISOZ



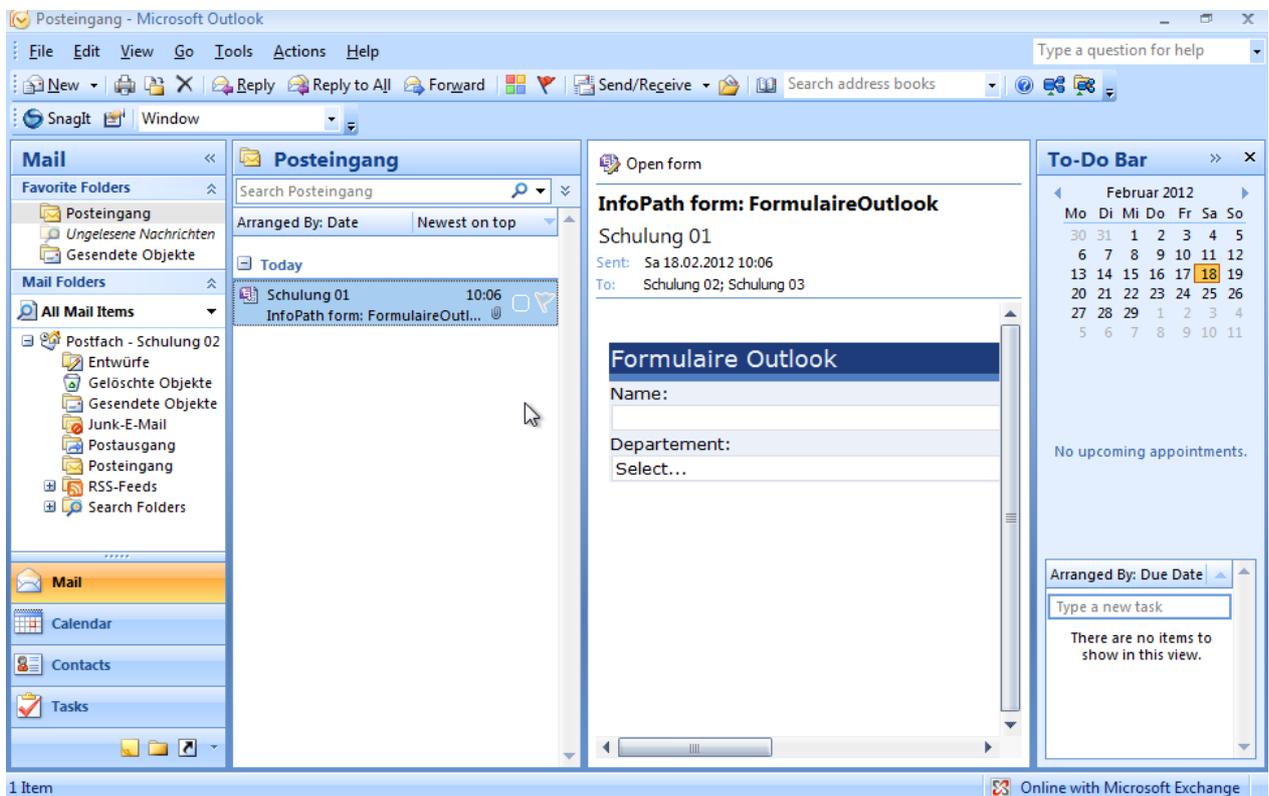
Nous cliquons sur **Next**:



et sur **Publish**. S'ouvre alors un e-mail Outlook avec tout ce qu'il faut:

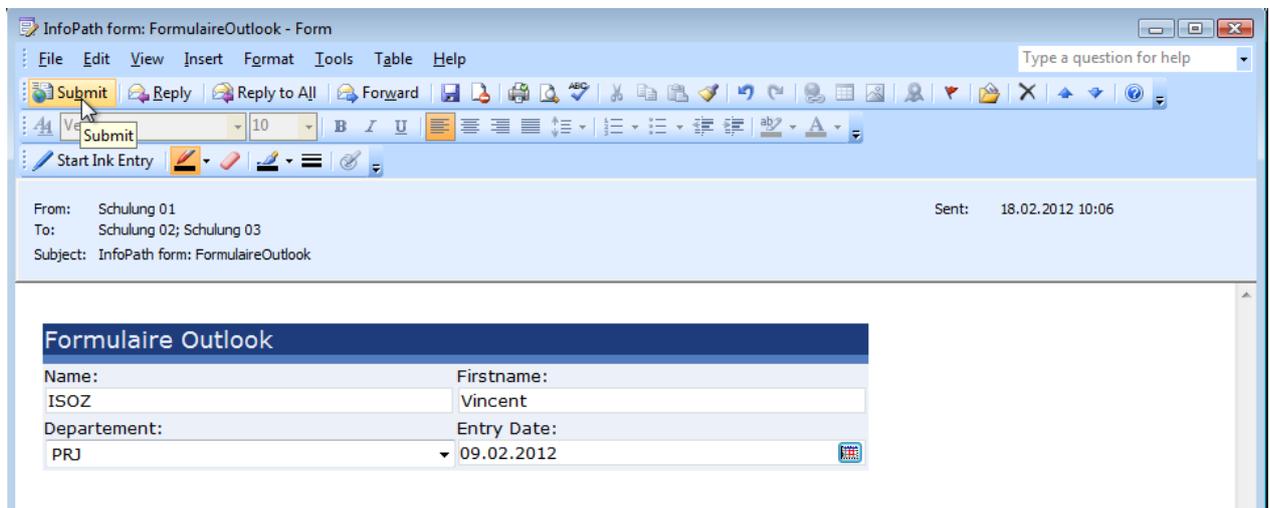


Nous l'envoyons aux destinataires de notre choix qui auront la vue suivante dans leur boîte de réception avec Outlook 2007:

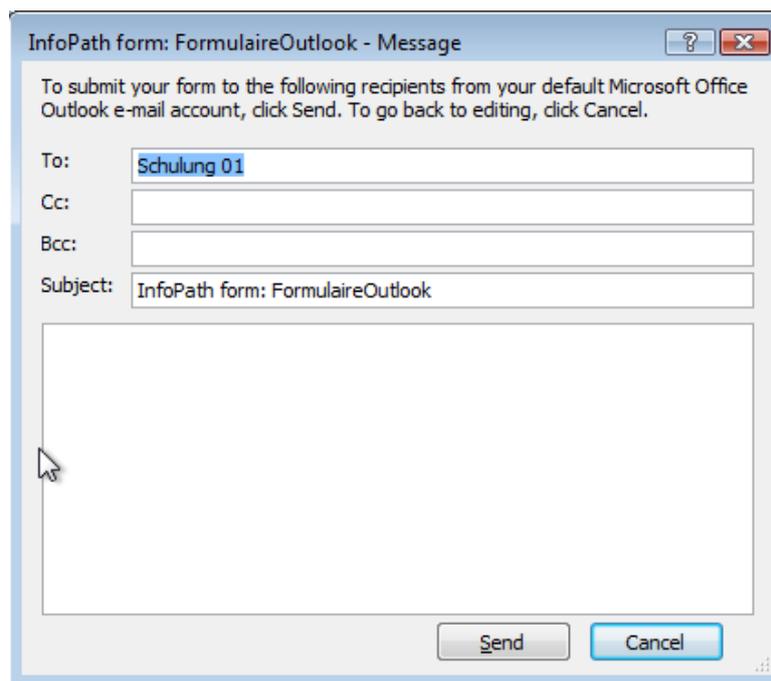


Vincent ISOZ

Il faut double cliquer et non pas faire un **Reply**. Une fois ceci fait, il suffit de remplir le formulaire et de cliquer en haut à gauche sur **Submit** (à moins que le designer ait créé un bouton *Submit* directement dans le message bien évidemment...):

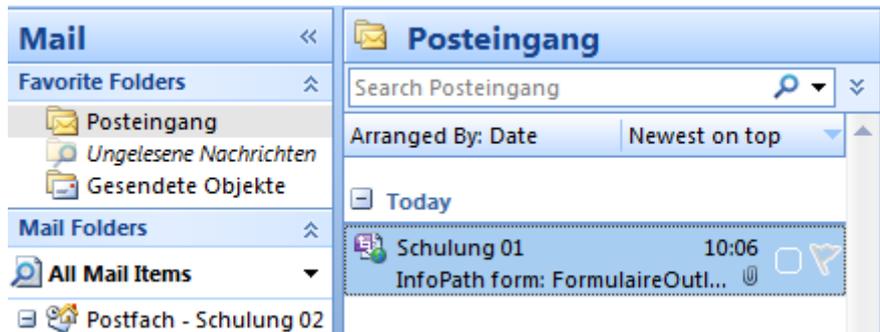


Quand le destinataire aura cliqué sur le bouton Submit, la boîte de dialogue suivante apparaît:

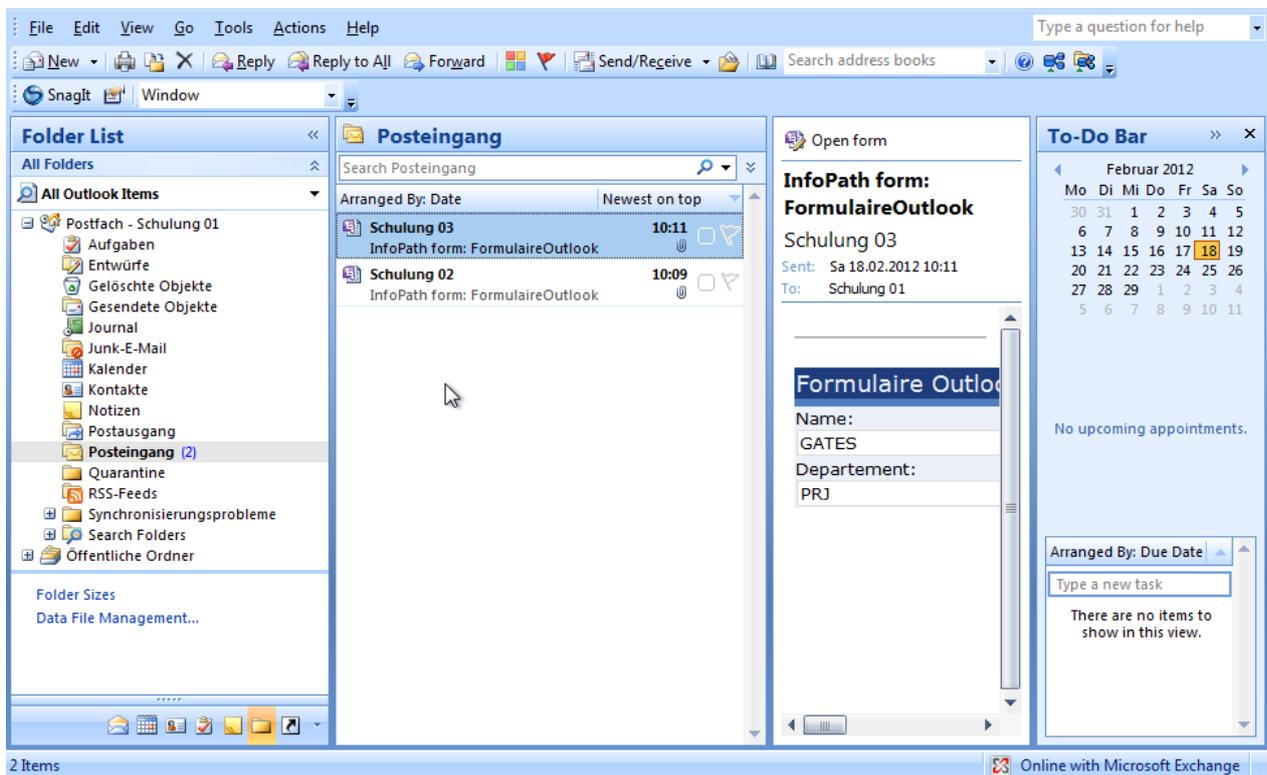


Il remplit et clique sur **Send**. Dès lors, dans sa boîte de réception, l'icône du mail initialement reçu va changer un tout petit peu pour montrer qu'il a déjà répondu:

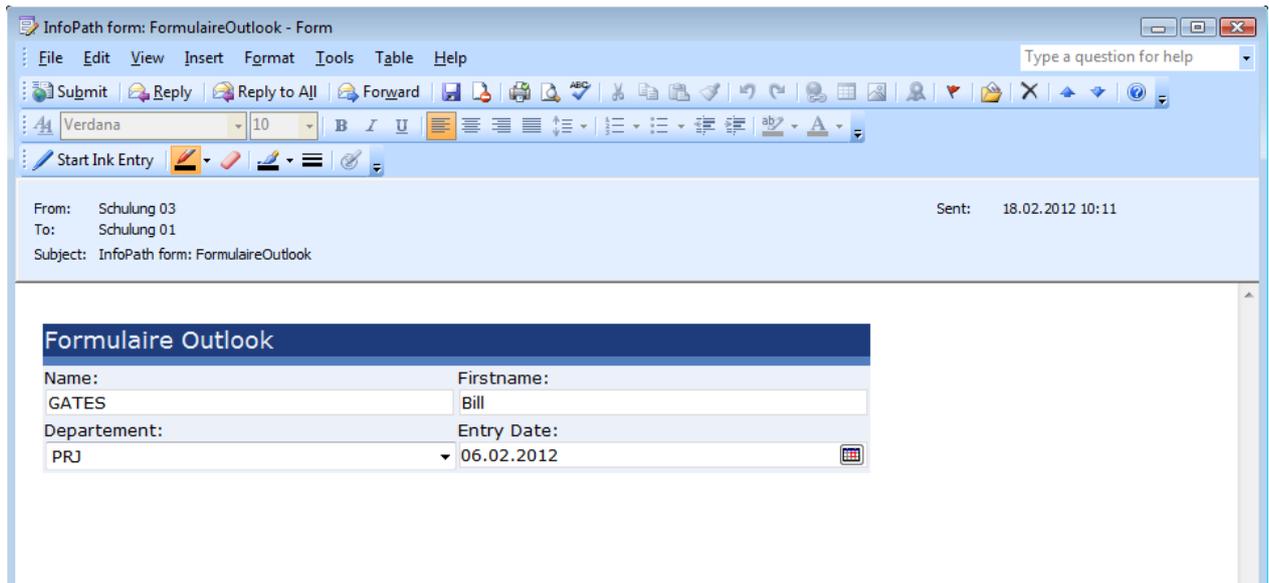
Vincent ISOZ



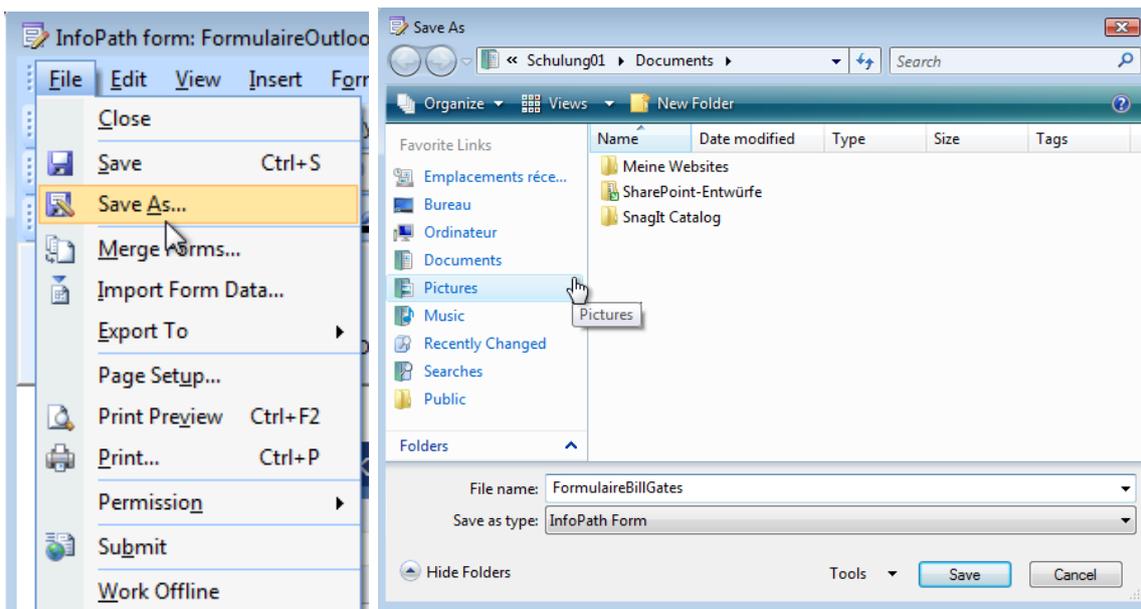
De son côté, l'expéditeur initial reçoit la réponse dans sa boîte aux lettres:

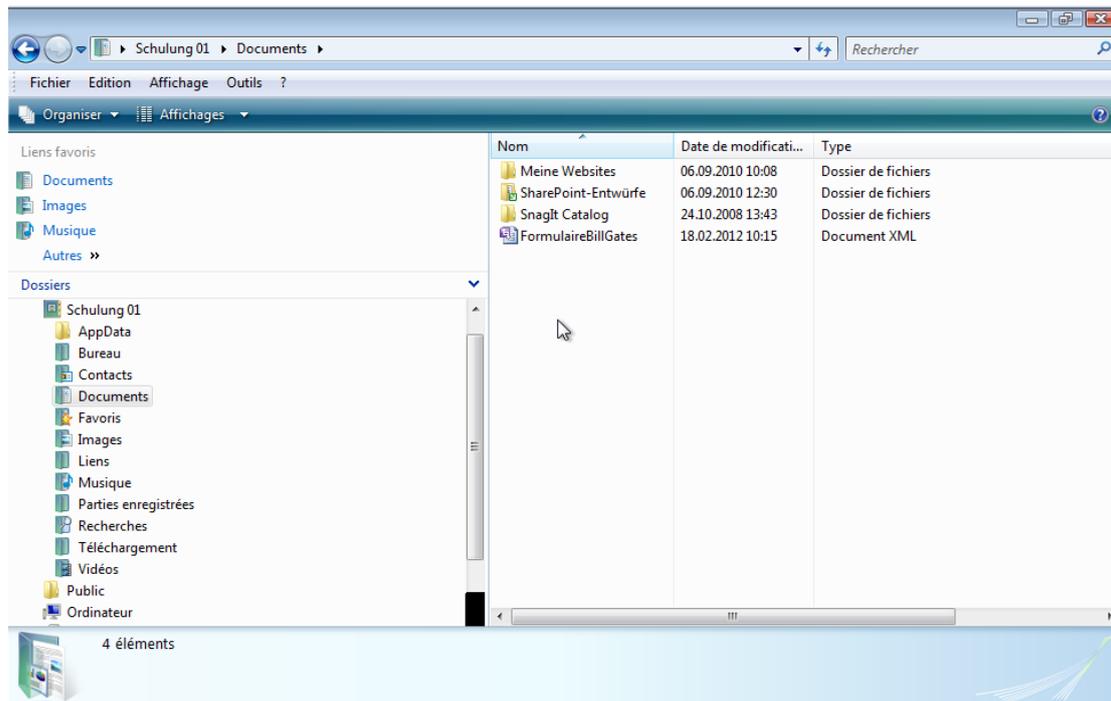


S'il double clique sur le mail il aura simplement la vue suivante:

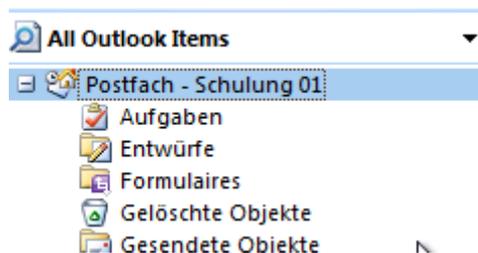
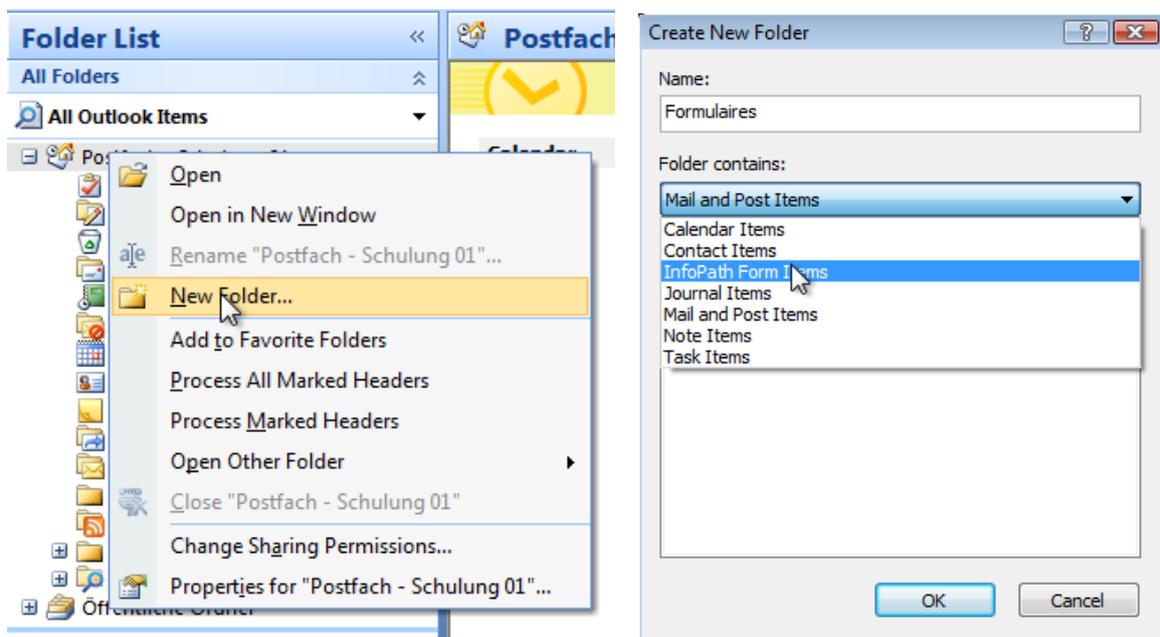


Il est intéressant de remarquer que si nous faisons un simple File/Save As, nous obtenons par défaut un fichier \*.xsn comme le montrent les trois captures ci-dessous:



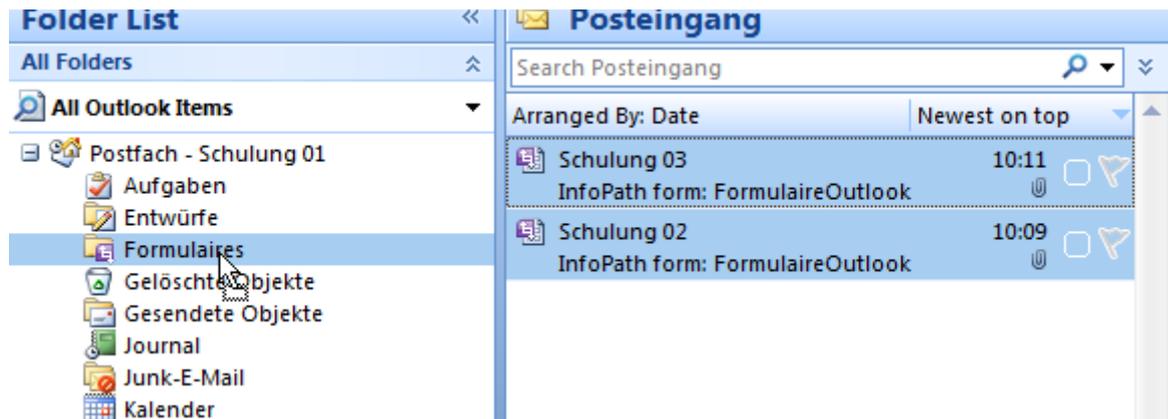


Par contre, observons que si le responsable crée dans Outlook un nouveau dossier de type **InfoPath Form Items**:

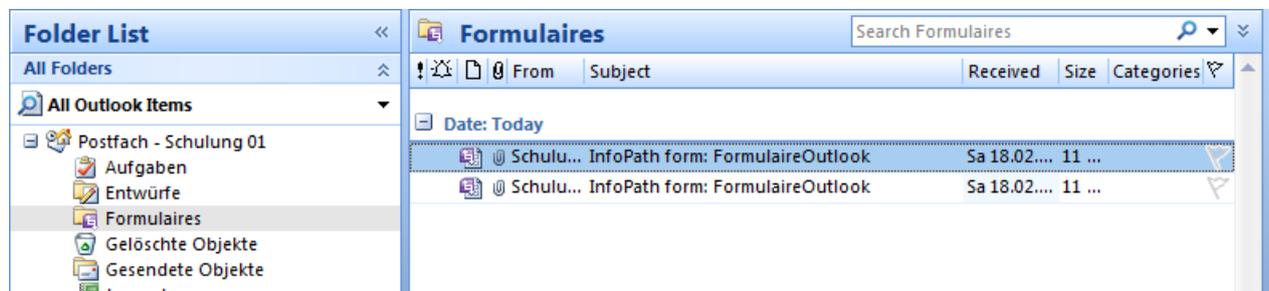


et y glisse les réponses:

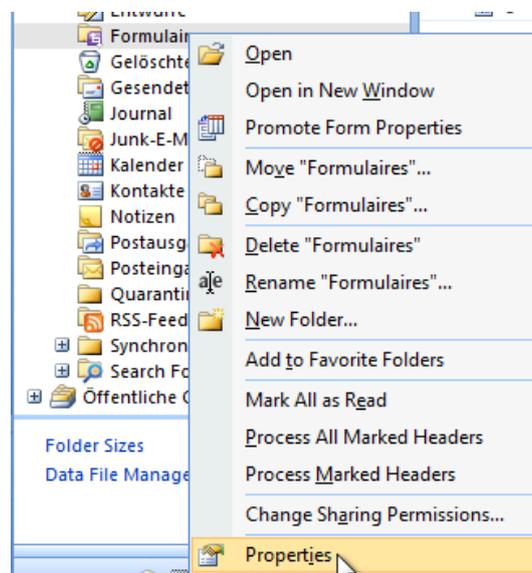
Vincent ISOZ



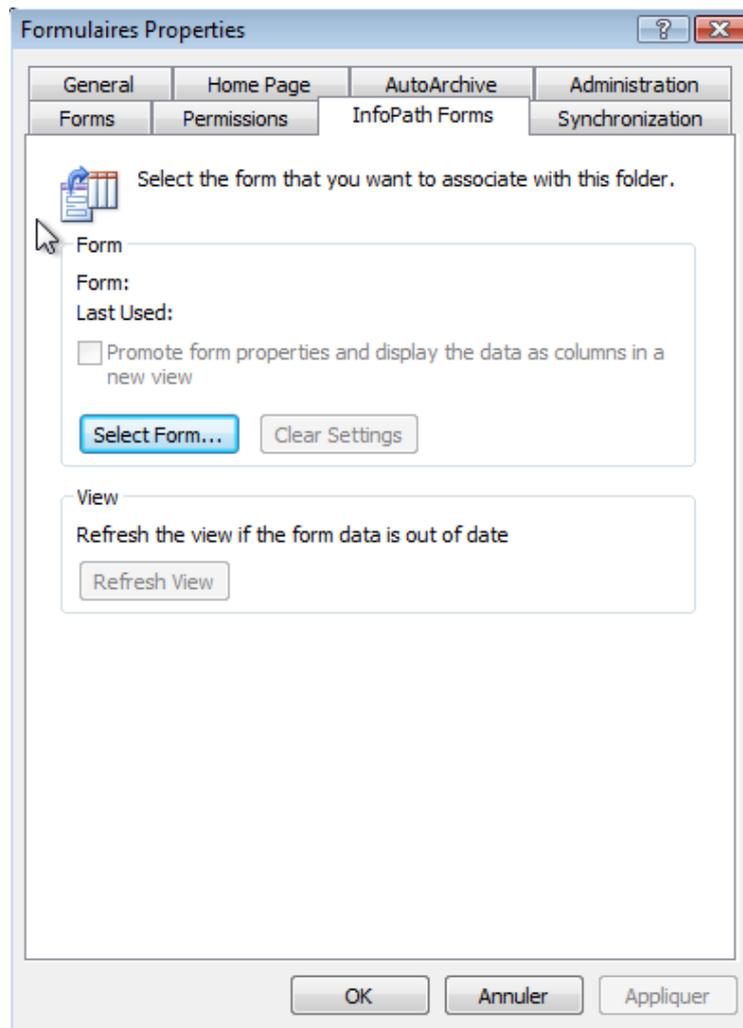
Si tout marche correctement, nous devrions y voir en colonne, les métadonnées des formulaires (bon ici c'est loupé sur la machine du client où j'ai fait les captures d'écran car quelque chose n'a pas fonctionné...):



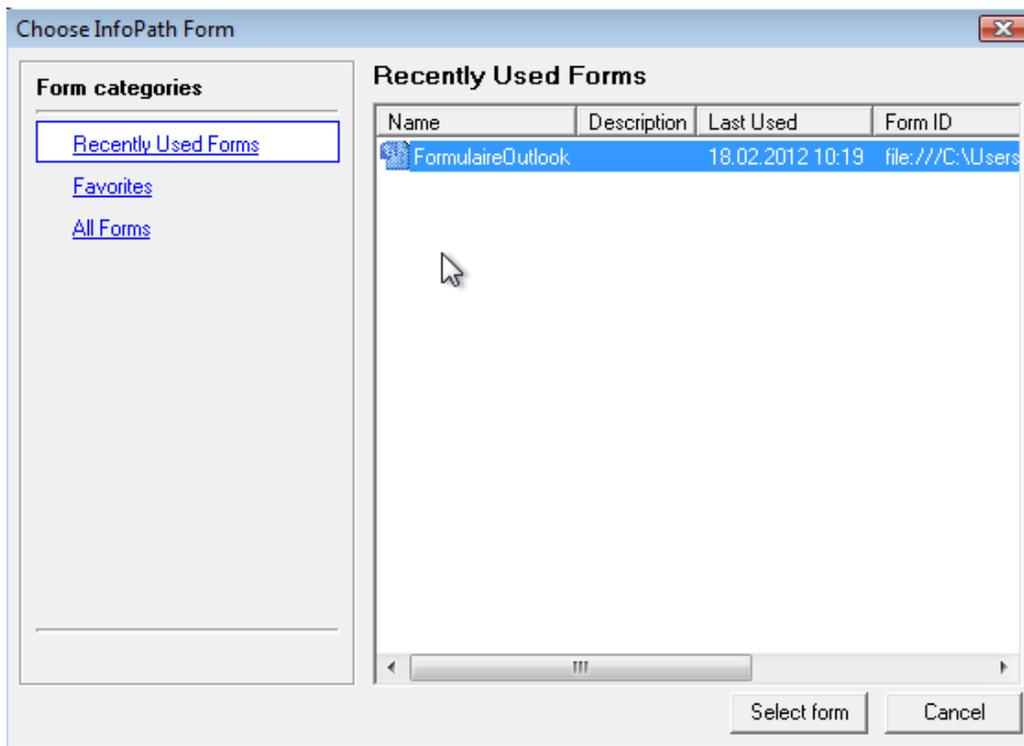
Si la promotion des métadonnées n'a pas fonctionné, on peut tenter de forcer en allant faire un clic droit sur les dossier de formulaires InfoPath et en allant dans **Properties**:



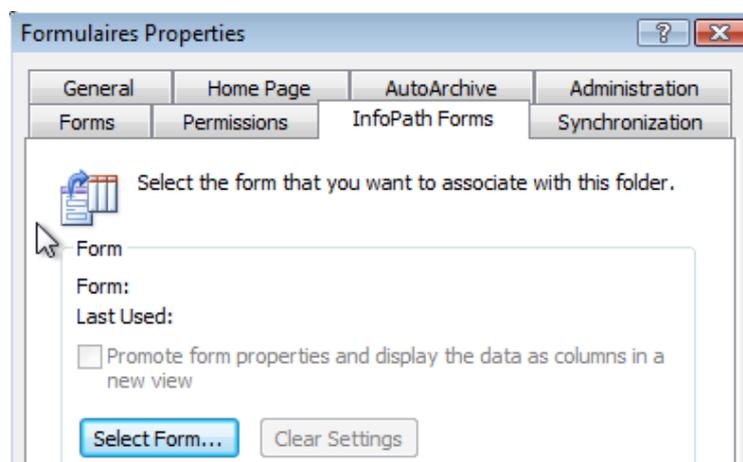
et dans l'onglet **InfoPath Forms**:



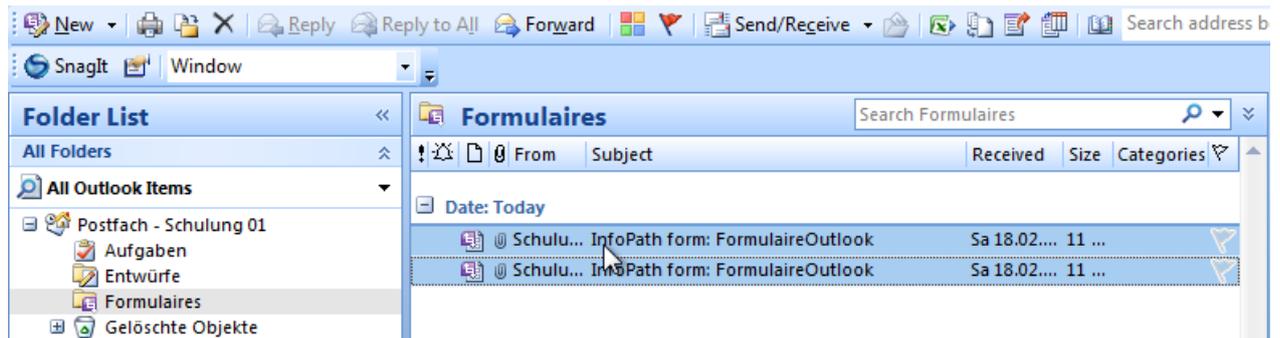
on clique sur le bouton **Select Form...** et s'ouvre alors la boîte de dialogue de sélection de formulaires d'InfoPath et nous y prenons notre formulaire précédemment créé:



et une fois ceci fait, de notre retour dans la boîte de dialogue précédente, nous devrions pouvoir cocher la case **Promote form properties and display the data as columns in a new view** mais chez le client chez qui j'ai fait les captures, cela ne marchait pas.



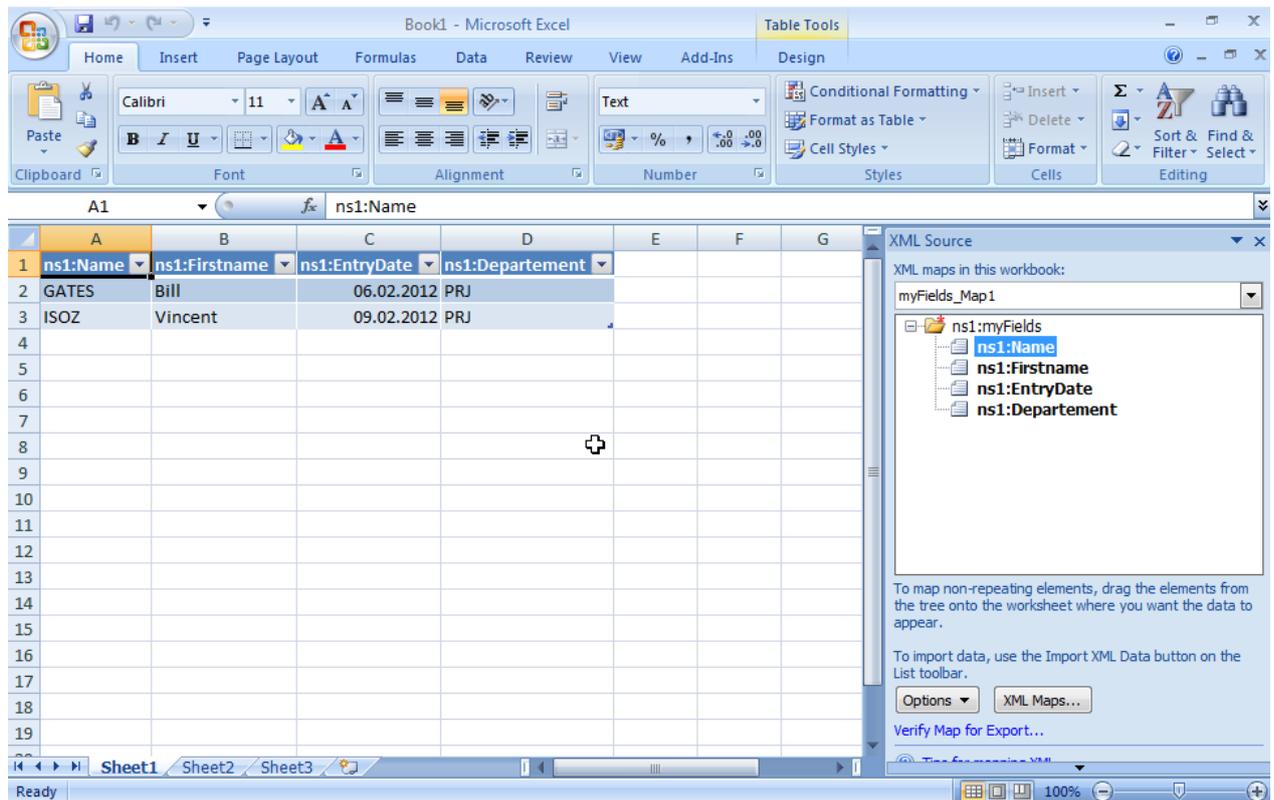
Bon ceci étant dit... Si le responsable sélectionne va dans son dossier de formulaires InfoPath et sélectionne toutes les réponses (ou une partie d'entre elles):



il remarquera dans la barre d'outils les 4 boutons suivants (j'ai zoomé le capture d'écran précédente)



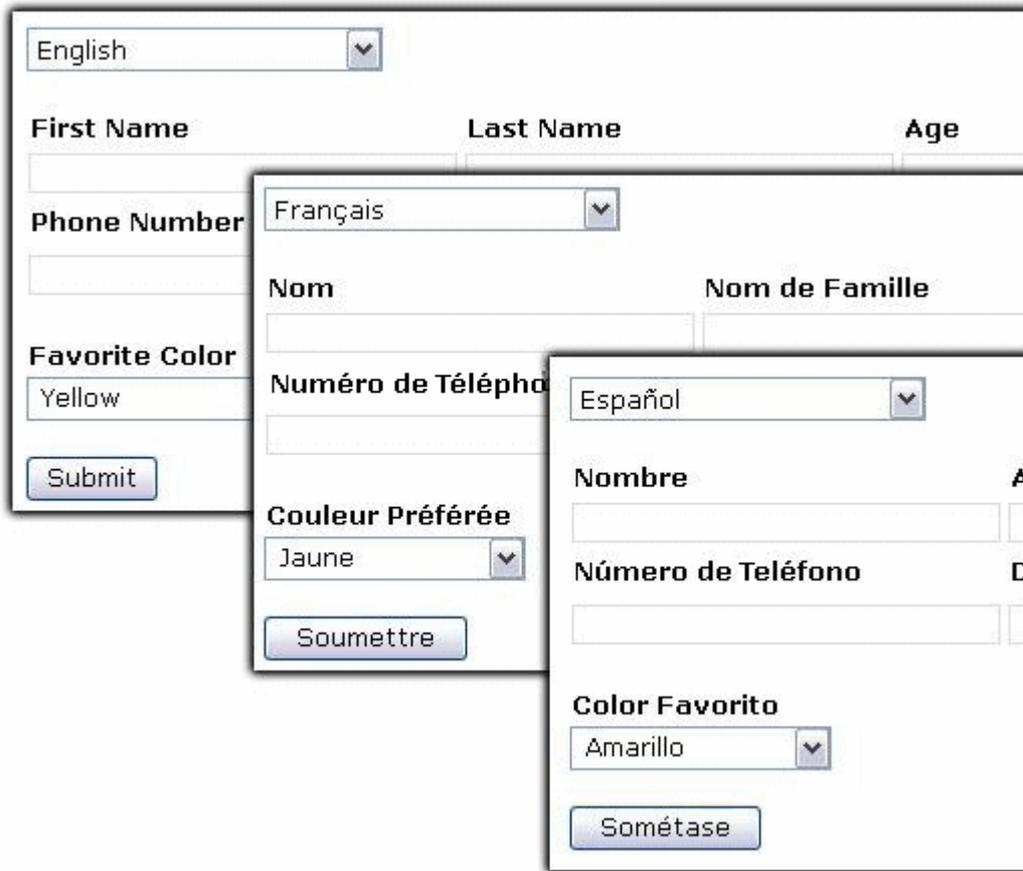
Nous allons cliquer sur le bouton le plus fréquemment utilisé qui est celui d'export vers MS Excel et nous obtenons:



## 9.16 Formulaires multi-langues

Quand un formulaire est utilisé dans différents pays les besoins d'adaptation de l'interface dans la langue principale devient un des objectifs prioritaire.

En utilisant des fonctions, une source de données secondaire et le filtrage de données il est possible avec InfoPath d'assez facilement répondre à cette problématique.



Afin d'éviter le travail à double ou à triple (facteur correspondant au nombre de langues existantes) il ne faut surtout pas dans ce genre de situation créer plusieurs vues. Certes cela marche mais les modifications deviennent alors à chaque fois pénibles à effectuer.

Comme il est d'usage depuis quelques années, nous allons faire appel au XML pour l'interface. Effectivement, nous allons créer un fichier XML qui contiendra les données standard de notre environnement et l'utiliserons comme source de données externe.

Créer un fichier *Language Settings.xml* contenant le code suivant:

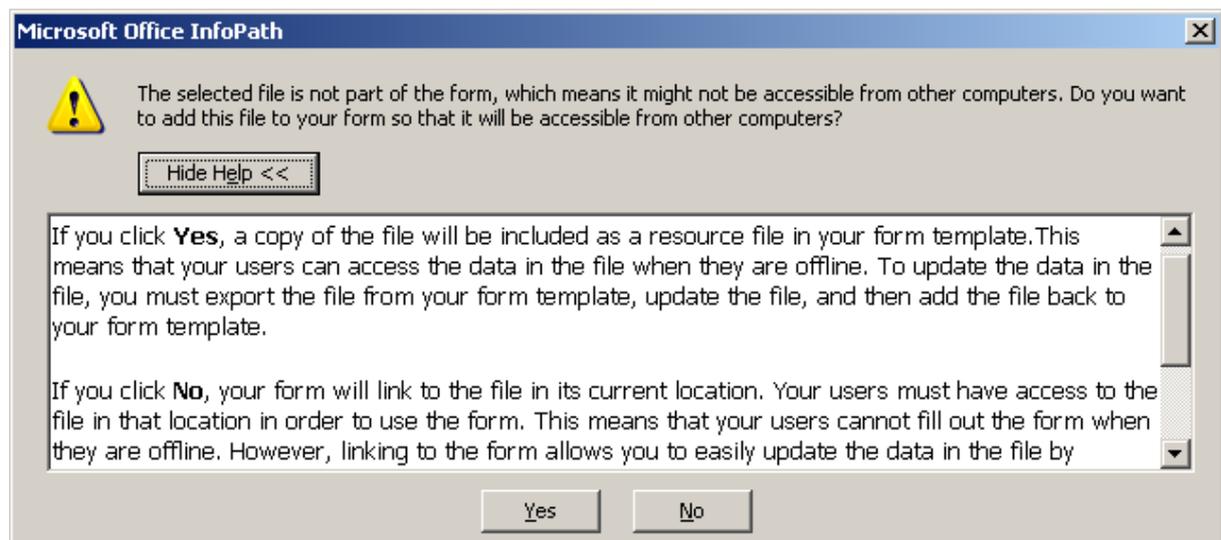
```
<?xml version="1.0" encoding="UTF-8"?>
<LocalSettings>
  <Languages>
    <Language Name="English" Code="ENG"/>
    <Language Name="Español" Code="SPA"/>
    <Language Name="Français" Code="FRE"/>
  </Languages>
  <Labels>
    <Label Name="First Name">
      <LocalLabel Code="ENG" Value="First Name"/>
      <LocalLabel Code="SPA" Value="Nombre"/>
      <LocalLabel Code="FRE" Value="Nom"/>
    </Label>
    <Label Name="Last Name">
      <LocalLabel Code="ENG" Value="Last Name"/>
      <LocalLabel Code="SPA" Value="Apellido"/>
      <LocalLabel Code="FRE" Value="Nom de Famille"/>
    </Label>
    <Label Name="Phone Number">
      <LocalLabel Code="ENG" Value="Phone Number"/>
      <LocalLabel Code="SPA" Value="Número de Teléfono"/>
      <LocalLabel Code="FRE" Value="Numéro de Téléphone"/>
    </Label>
  </Labels>
</LocalSettings>
```

## Vincent ISOZ

```
</Label>
<Label Name="Address">
  <LocalLabel Code="ENG" Value="Address"/>
  <LocalLabel Code="SPA" Value="Dirección"/>
  <LocalLabel Code="FRE" Value="Adresse"/>
</Label>
<Label Name="Age">
  <LocalLabel Code="ENG" Value="Age"/>
  <LocalLabel Code="SPA" Value="Edad"/>
  <LocalLabel Code="FRE" Value="Age"/>
</Label>
<Label Name="FavoriteColor">
  <LocalLabel Code="ENG" Value="Favorite Color"/>
  <LocalLabel Code="SPA" Value="Color Favorito"/>
  <LocalLabel Code="FRE" Value="Couleur Préférée"/>
</Label>
</Labels>
<Colors>
  <LocalColor ID="Blue" Code="ENG" Value="Blue"/>
  <LocalColor ID="Blue" Code="SPA" Value="Azul"/>
  <LocalColor ID="Blue" Code="FRE" Value="Bleu"/>
  <LocalColor ID="Green" Code="ENG" Value="Green"/>
  <LocalColor ID="Green" Code="SPA" Value="Verde"/>
  <LocalColor ID="Green" Code="FRE" Value="Vert"/>
  <LocalColor ID="Red" Code="ENG" Value="Red"/>
  <LocalColor ID="Red" Code="SPA" Value="Rojo"/>
  <LocalColor ID="Red" Code="FRE" Value="Rouge"/>
  <LocalColor ID="Yellow" Code="ENG" Value="Yellow"/>
  <LocalColor ID="Yellow" Code="SPA" Value="Amarillo"/>
  <LocalColor ID="Yellow" Code="FRE" Value="Jaune"/>
</Colors>
</LocalSettings>
```

Ensuite, créez un nouveau formulaire, choisissez *Tools/Data Connections* et cliquez sur *Add*. Dans l'assistant sélectionnez *Receive Data* et ensuite cliquez sur *Next*. Sélectionnez en cliquant sur *Browse* votre fichier *Language Settings.xml* et cliquez ensuite sur *Next* et *Finish*.

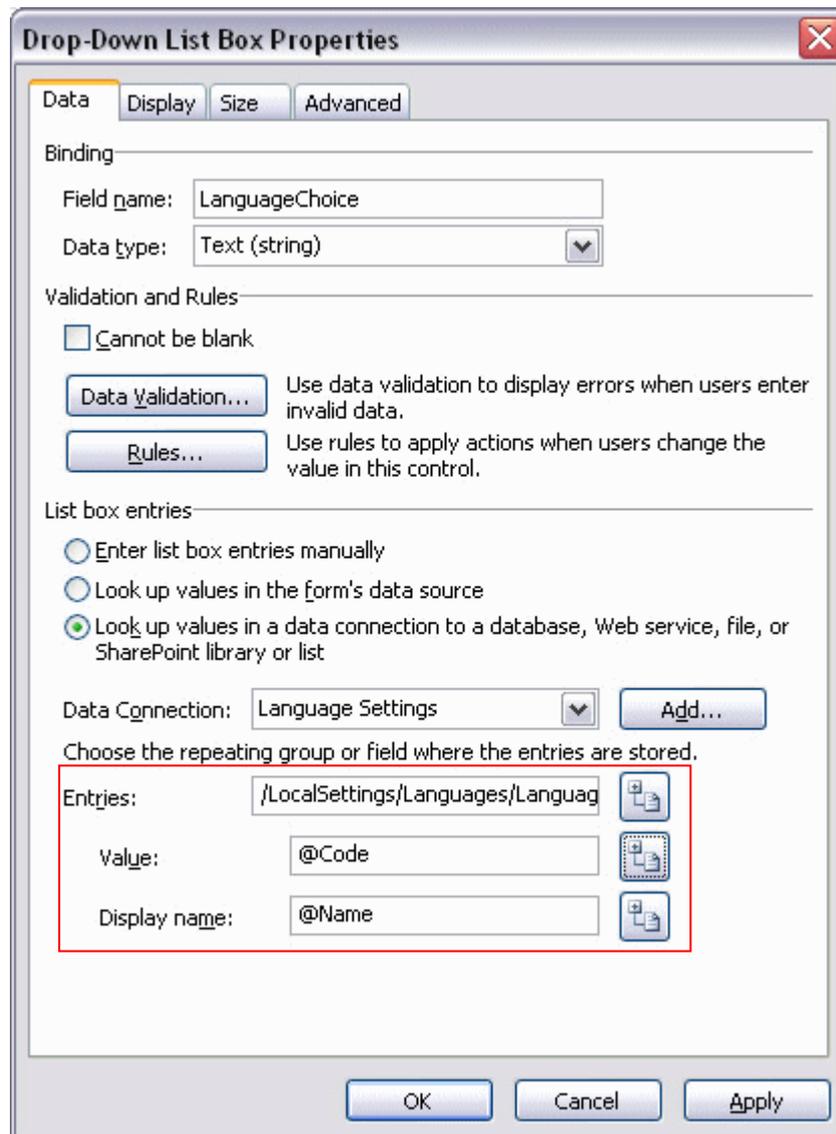
La boîte de dialogue suivante apparaîtra:



Cliquez sur *No*.

Nous allons maintenant créer la liste déroulante avec le choix des langues. Nous savons déjà comment faire cette étape. Petite précision cependant, nommez le champ *LanguageChoice* et

faites en sorte que nous ayons les noms des langues dans la liste déroulante et non le code des langues de plus:



Ensuite, nous ajoutons les contrôles:

1. Ouvrez le panneau *Office Layout*
2. Ajoutez une table personnalisée de 3 colonnes et 4 lignes
3. Fusionnez les deux dernières colonnes de la troisième et quatrième ligne (voir capture d'écran plus bas)
4. Insérez ensuite depuis le panneau *Controls* une *Text box* dans la deuxième ligne de la première colonne.
5. Double cliquez sur ce nouveau contrôle et changez son nom à *FirstName* et cliquez sur *OK*.

6. Insérez une *Text box* dans la deuxième ligne de la deuxième colonne que nous nommerons *LastName*.
7. Insérez une *Text Box* dans la dernière colonne de la deuxième ligne et changez son nom à *Age*.
8. Insérez une *Text Box* dans la dernière ligne de la première colonne que nous nommerons *PhoneNumber*.
9. Insérez une *Text Box* dans la dernière ligne de la dernière colonne que nous nommerons *Address*.

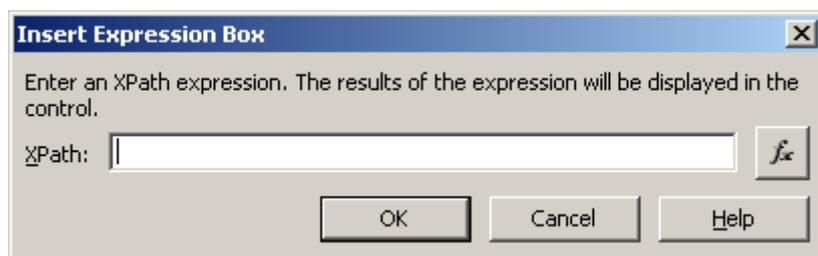
Maintenant que nous avons tous ces contrôles, nous allons y ajouter les étiquettes. En utilisant les fonctions d'expression d'InfoPath, nous allons pouvoir contrôler le contenu de chacune de ces étiquettes.

Pour ce faire:

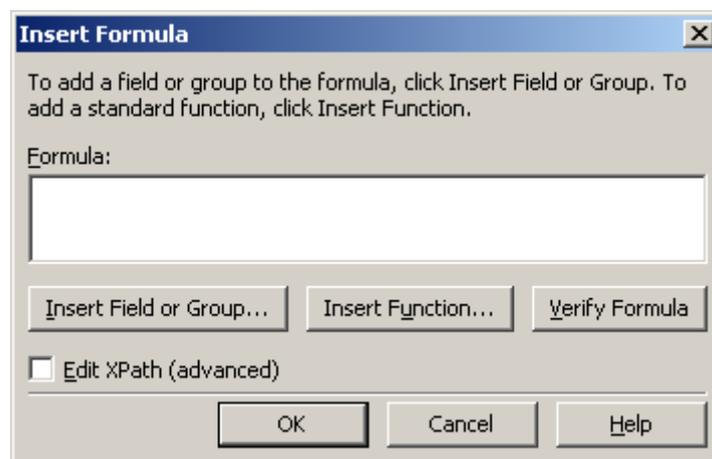
1. Cliquez dans la première ligne de la première colonne
2. Sélectionnez le contrôle *Expression Box* du panneau Office des contrôles

 Expression Box

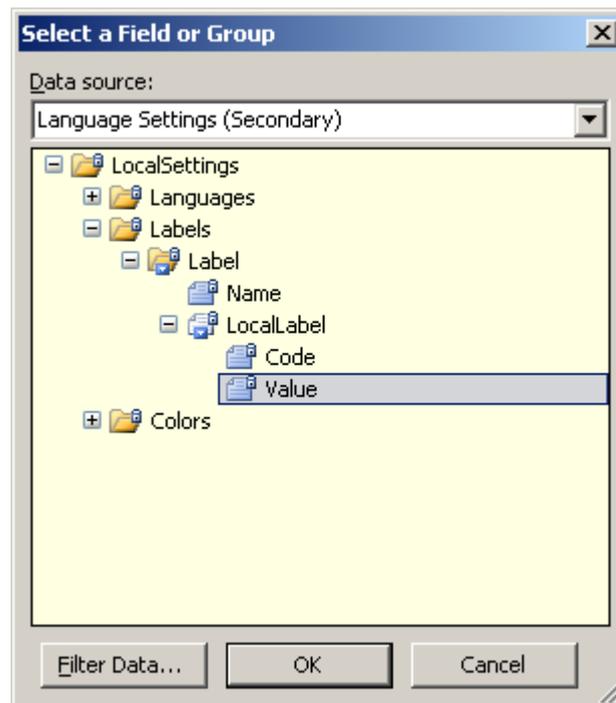
3. Cliquez sur le bouton *fx* dans la boîte de dialogue qui apparaît.



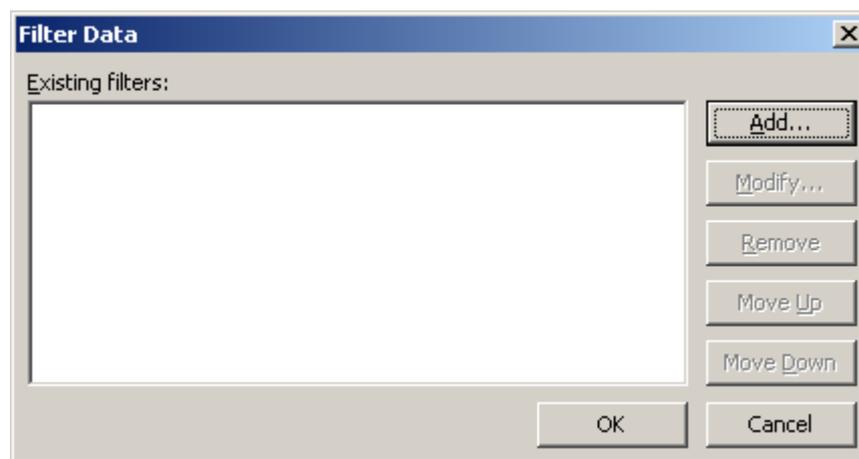
4. Cliquez sur le bouton *Insert Field or Group* dans la boîte de dialogue suivante.



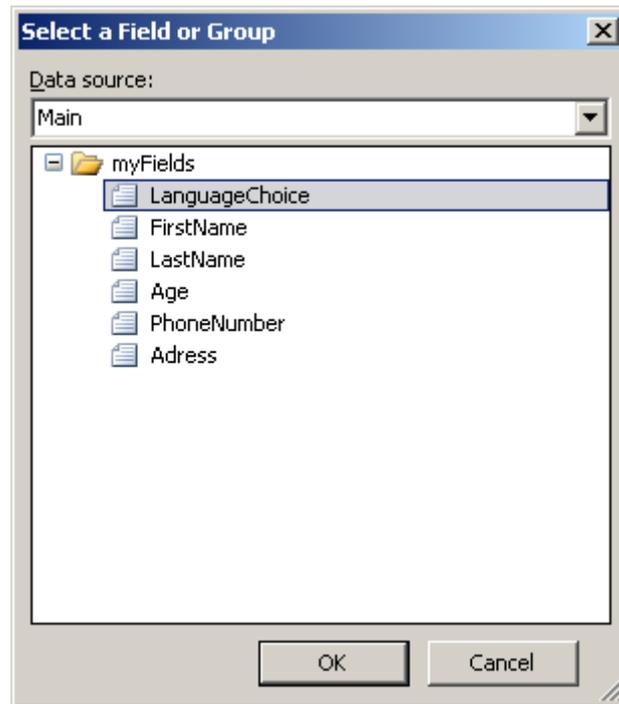
5. Sélectionnez dans la liste déroulante de la boîte de dialogue *Data Source* la source de données *Language Settings (Secondary)* et ensuite:



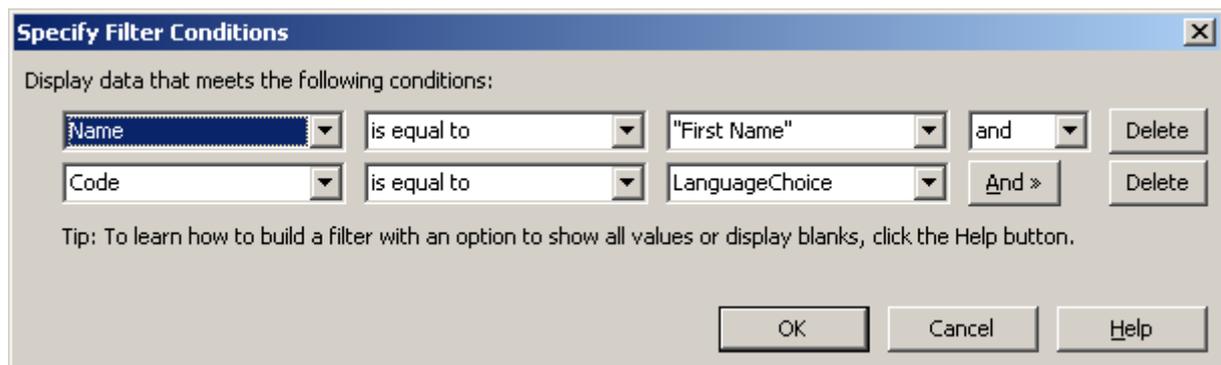
et ensuite cliquez sur *Filter Data*. Apparaît alors:



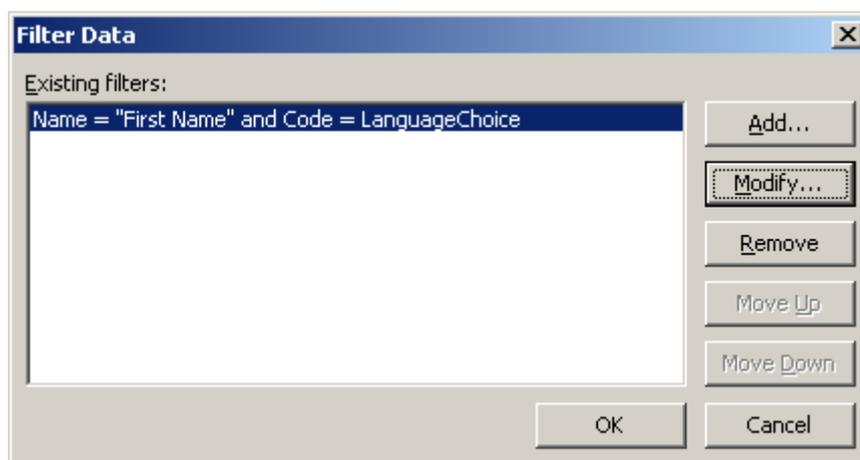
6. Cliquez sur *Add*.
7. Sélectionnez *Name* de l'option *Select a field or group* de la première liste déroulante. L'option *Name* provenant de la source *Labels/Label/Name*. Ensuite saisissez "*First Name*" dans la troisième liste déroulante.
8. Cliquez sur *And* et sélectionnez *Code* de la première liste déroulante et ensuite choisissez *Select a field or group* de la troisième liste déroulante et sélectionnez de la data source *Main* le chemin suivant *myFields/LanguageChoice*.



9. Validez par *OK*:



10. Encore par *OK*:



11. et encore trois fois par *OK*.

12. Répétez les étapes précédents à l'identique (ou faites un copier/coller c'est selon...) pour tous les labels excepté en ce qui concerne "First Name" de l'étape 7 qu'il faudra adapter à l'étiquette que vous êtes entrain de créer.

Pour mettre les étiquettes en gras il suffit de les sélectionner et de cliquer sur le bouton habituel qui formate le texte en gras.

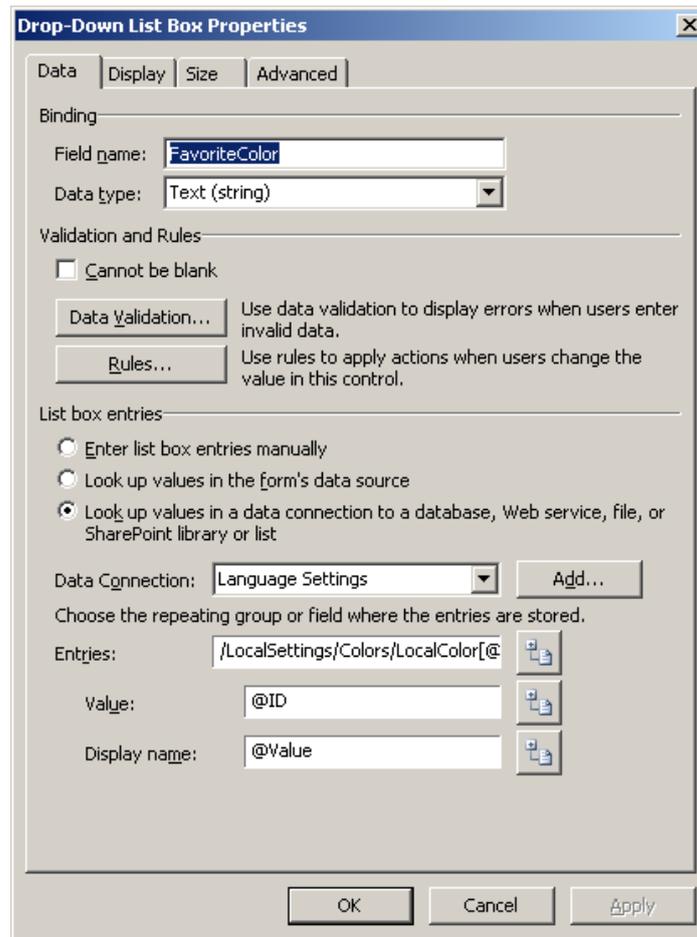
Si vous testez le formulaire vous noterez qu'il est vide au début ce qui est assez gênant... Pour remédier à ce problème nous allons ajouter une valeur par défaut.

Pour ce faire, dans le panneau *Data Source* du volet office faites un double clic sur le nœud *myFields/LanguageChoice* et tapez *ENG* comme valeur par défaut et validez par *OK*.

Nous allons maintenant ajouter un composant supplémentaire. Nous allons ajouter une liste déroulante utilisant des fonctionnalités similaires aux étiquettes.

Pour faire notre liste déroulante:

1. En-dessous de la table (quelques lignes en-dessous) insérez une étiquette qui affichera selon la langue choisie le texte *Favorite Color*
2. Ouvrez le panneau des contrôles et ajoutez une *Drop-Down List* en-dessous de l'étiquette
3. Double cliquez sur le contrôle et nommez le *FavoriteColor*
4. Sélectionnez *Look Up Values In A Data Connection*, sélectionnez *Language Setting* de la liste déroulante été ensuite cliquez sur le bouton de sélection de champ
5. Sélectionnez *Colors/LocalColor* et ensuite cliquez sur le bouton *Filter Data*
6. Cliquez sur le bouton *Add*, sélectionnez *Code* dans la première liste déroulante et ensuite sélectionnez *Select A field Or Group* dans la troisième liste.
7. Choisissez la source *Main* et sélectionnez-y */myfields/LanguageChoice* et validez par *Ok* et retournez dans la boîte de dialogue des propriétés de la liste déroulante.
8. Sélectionnez l'attribut *@ID* dans le champ *Value* et *@Value* dans le champ *Display Name* et cliquez sur *OK*.



Le dernier aspect dont nous devons nous occuper sont les boutons de soumission du formulaire. Malheureusement, InfoPath ne permet pas de contrôler la légende des boutons via programmation. Nous utiliserons donc plusieurs boutons sur lesquels nous appliquerons un formatage conditionnel.

1. Ouvrez le panneau de contrôle et ajoutez un bouton dans le formulaire
2. Double cliquez sur le bouton et changez le *Label* à *Submit* à l'*ID* à *btnSubmit*.
3. Cliquez sur l'onglet *Display* et ensuite sur le bouton *Conditional Formatting* et cliquez sur *Add*
4. Dans la première liste déroulante choisissez *LanguageChoice*, sélectionnez *Is Not Equal* dans la seconde, et sélectionnez *Type Text* et saisissez *ENG* dans la troisième liste déroulante
5. Cochez la case *Hide This Control* et validez par *OK* trois fois
6. Copiez et collez deux fois le bouton
7. Changez le *Label* du deuxième bouton à *Sométase* et l'*ID* à *btnSubmit* et changez le formatage conditionnel avec le critère *SPA* au lieu de *ENG*.
8. Changez le *Label* du troisième bouton à *Soumettre* et l'*ID* à *btnSubmit* et changez le formatage conditionnel avec le critère *FRE* au lieu de *ENG*.

Vincent ISOZ

Remarque: Les trois boutons doivent avoir le même ID parce qu'ils effectueront la même action et tout code que nous attacherions à un des boutons s'appliquera aux autres.

The image shows a screenshot of a web form. At the top, there is a dropdown menu. Below it is a table with a dashed border, containing several empty cells. Underneath the table is another dropdown menu. At the bottom of the form, there are three buttons labeled "Submit", "Sométase", and "Soumettre".

## 10. Applications pratiques (avec dév.)

Nous allons voir ici dans les détails comment appliquer des techniques de développement JavaScript ou .Net (smart forms) avec des exemples à complexité croissante.

### 10.1 Utilisation de JS comme calculatrice

Nous allons ici nous amuser à faire une petite calculatrice effectuant seulement une addition. Pour cela créez le formulaire suivant:



The screenshot shows a web browser window with a simple calculator form. The form consists of three text input fields. The first is labeled 'Valeur 1 :', the second 'Valeur 2 :', and the third 'Somme :'. Below these fields is a button labeled 'Ajouter'. The browser's address bar shows 'Dessiner un tableau' and 'Aucune bordure'.

avec les champs respectivement nommées dans l'ordre *val1*, *val2*, *total* et le bouton ayant comme ID *AddAction*.

Ensuite, dans l'éditeur de script, saisissez le code suivant sur l'événement clic du bouton:

```
function AddAction::OnClick(eventObj)
{
    var val1Field = XDocument.DOM.selectSingleNode("//my:val1");
    var val2Field = XDocument.DOM.selectSingleNode("//my:val2");
    var totalField = XDocument.DOM.selectSingleNode("//my:total");

    var valint1 = new Number(val1Field.text);
    var valint2 = new Number(val2Field.text);
    totalField.text=new String(valint1+valint2);
}
```

On peut si nous le désirons ajouter le résultat dans une boîte d'alerte avec la commande ad hoc.

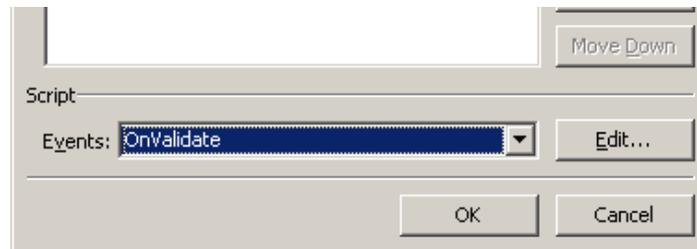
### 10.2 Utilisation de JS pour la validation des données

Nous allons dans cet exemple faire une démonstration de comment InfoPath utilise les *RegularExpressions* pour valider le format de données saisies et ce avec du code JS.

Pour commencer, nous allons créer un formulaire vide avec un champ de type téléphone à l'américaine basique (au format 111-111-1111). Le champ sera nommé *fldPhone*:

Téléphone (US) :

Si vous faites un double clic sur le champ, sélectionnez *Data Validation* et choisissez *On Validate* et ensuite cliquez sur *Edit*:



et saisissez-y le code suivant avec la *RegExpr* correspondant au n° de téléphone US:

```
function msosxd_my_fldPhone::OnValidate(eventObj)
{
  if(eventObj.Site.nodeTypeValue != "")
  {
    var strPhone = eventObj.Site.nodeTypeValue;
    var RegEx = /^(\d{3})?(\d{3}-)\d{3}-\d{4}$/g;
    if(!RegEx.test(strPhone))
    XDocument.UI.Alert( "Syntaxe du numéro de téléphone incorrecte" );
  }
}
```

Si maintenant vous testez ce champ en saisissant quelque chose d'incohérent vous aurez:



D'autres expressions régulières types:

`/[Bb]ug\s(#)\d{6}/g`

oblige l'utilisateur à saisir un numéro de bug du type *Bug #123456* ou *bug #123456*.

Pour contrôler que l'information saisie correspond bien à une adresse électronique:

`/([a-zA-Z0-9_-\.\])+@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. )|([a-zA-Z0-9-\.\+])|([a-zA-Z]{2,4}[0-9]{1,3})\[\]?)/g`

c'est un peu plus long...

### 10.3 Utilisation de ADO avec JS

InfoPath propose comme nous l'avons vu la possibilité de faire une requête sur une table ou une requête *MS Access* simple.

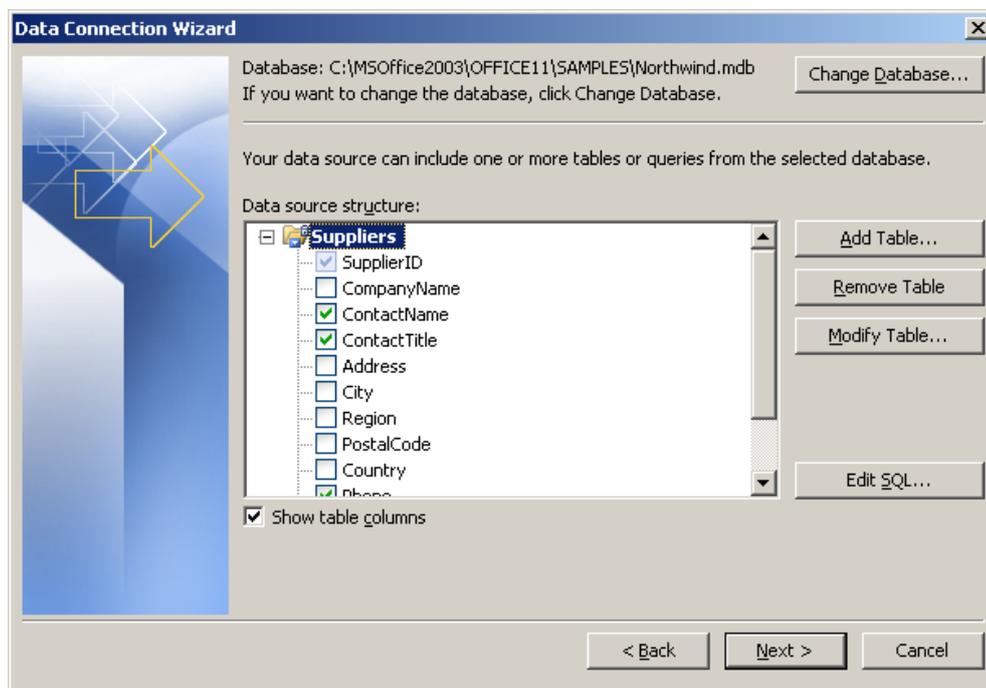
Vincent ISOZ

Pour faire des requêtes plus complexes il faudra passer par la programmation.

Nous allons dans cet exemple utiliser la base *Northwind* pour changer.

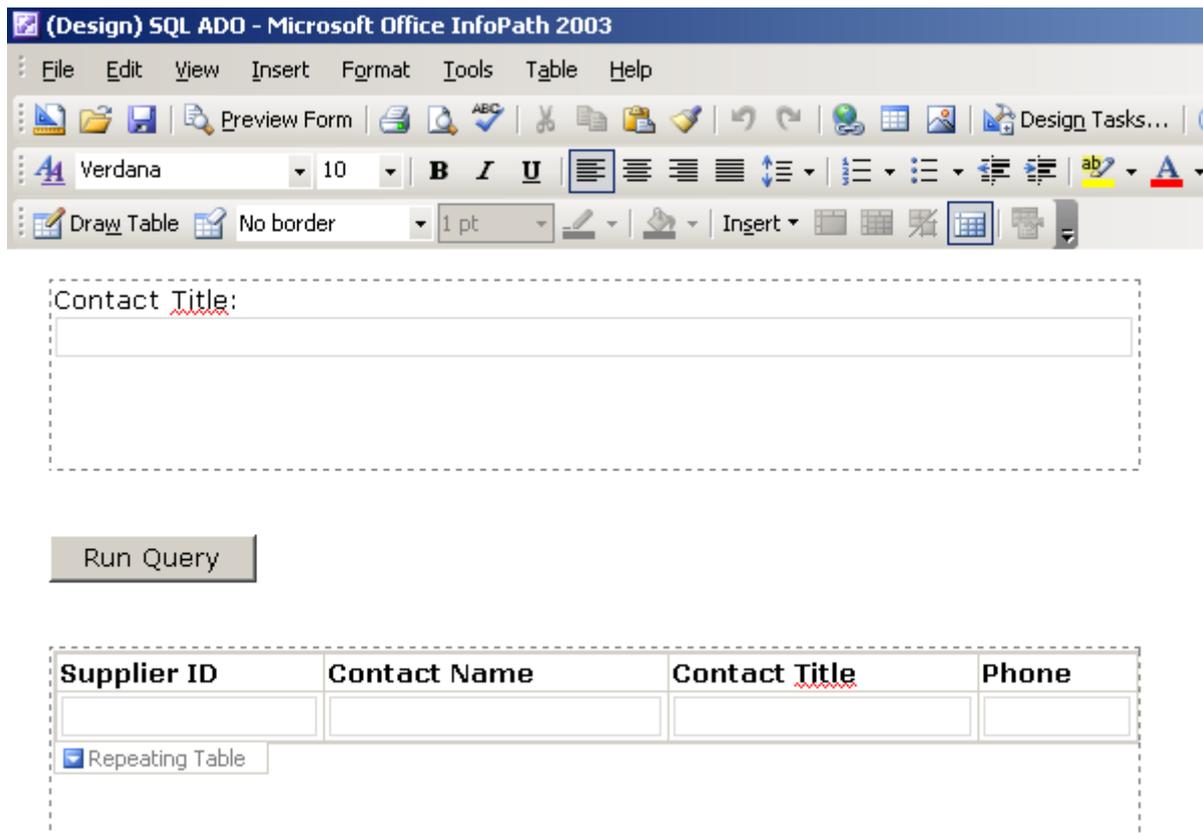
Dans un premier temps il faudra construire une requête simple:

1. Démarrez *InfoPath*
2. Créez un nouveau formulaire vierge en mode *Design*
3. Sélectionnez *New from Data Source*
4. L'assistant démarre. Sélectionnez *Database (Microsoft SQL Server or Microsoft Office Access only)*, et cliquez sur *Next*.
5. Cliquez sur *Select Database*
6. Allez chercher sur votre disque dur la base de données *Northwind.mdb* (*Comptoirs.mdb* en français)
7. Dans la boîte de dialogue *Select Table*, allez chercher la table *Suppliers* et cliquez sur *OK*.
8. Dans la liste déroulant *Data source structure*, choisissez seulement les champs *SupplierID*, *ContactName*, *ContactTitle* et *Phone*

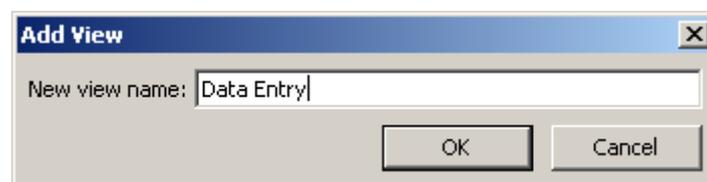


et cliquez sur *Next* et ensuite sur *Finish*

Maintenant, de même que pour l'exemple que le premier exemple que nous avons fait avec Access et la requête. Faites en sorte d'avoir le formulaire suivant:



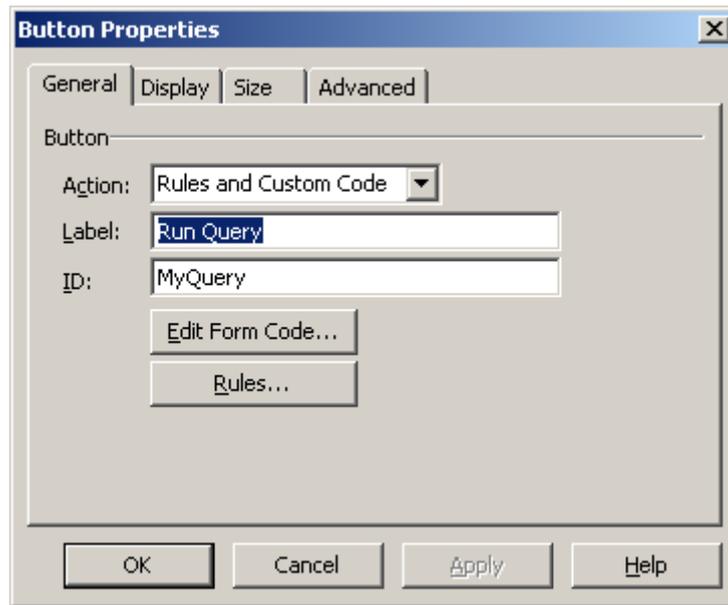
Ensuite, via le panneau Office cliquez *Views* et ensuite cliquez sur le lien en bas du panneau *Add a New view* et nommez cette nouvelle vue *Data Entry*



et glissez-y tous les champs *d:Suppliers* de manière à avoir:



Ensuite, retournez dans la première vue par défaut nommée *View 1* et faites un double clic sur le bouton *Run Query*:



Changez le *Label* et l'*ID* comme indiqué dans la capture d'écran ci-dessus.

Ensuite, cliquez sur *Edit Form Code...* et saisissez y le code suivant (provenant du site MSDN de Microsoft):

## Vincent ISOZ

```
function MyQuery::OnClick(eventObj)
{
    // Get the default SQL command for the form.
    var strOrigSQLCommand = XDocument.QueryAdapter.Command;

    // Get the query node that you want to modify.
    var querySuppliers = XDocument.DOM.selectSingleNode(
        "/dfs:myFields/dfs:queryFields/q:Suppliers");

    // Obtain the text that was entered for the wildcard character search, and then clear
    // the current query parameter so that InfoPath will leave the current query parameter blank.
    var strTitle = querySuppliers.selectSingleNode("@ContactTitle").text;
    querySuppliers.selectSingleNode("@ContactTitle").text = "";

    // Ask InfoPath to construct an SQL command that is based on all other field values.
    var strMySQLCommand = XDocument.QueryAdapter.BuildSQLFromXMLNodes(querySuppliers);
    var strContactName = querySuppliers.selectSingleNode("@ContactName").text;
    querySuppliers.selectSingleNode("@ContactName").text = "";
    // Save each of the other query items, and then clear the other query items before the next query.
    var strSupplierID = querySuppliers.selectSingleNode("@SupplierID").text;
    querySuppliers.selectSingleNode("@SupplierID").text = "";

    var strPhone = querySuppliers.selectSingleNode("@Phone").text;
    querySuppliers.selectSingleNode("@Phone").text = "";

    // Add ContactTitle to the query so
    // that ContactTitle can support wildcard characters.
    if (strTitle != "")
        if (strMySQLCommand != "")
            strMySQLCommand = strMySQLCommand + " AND ";

    strMySQLCommand = strMySQLCommand +
        "([Suppliers].[ContactTitle] LIKE \"\" + strTitle + \"\")";
}

// Construct the full query string.
var strSQLQuery = strOrigSQLCommand;
if (strMySQLCommand != "")
    strSQLQuery = strSQLQuery + " WHERE " + strMySQLCommand + " ORDER BY [Suppliers].[ContactName]";

// This is the query.
//XDocument.UI.Alert(strSQLQuery);

// Run the query.
XDocument.QueryAdapter.Command = strSQLQuery;
XDocument.Query();

// Restore all the user entries to the Query fields so that the user entries will
// be available if you want to modify and to rerun the query.
querySuppliers.selectSingleNode("@SupplierID").text = strSupplierID;
querySuppliers.selectSingleNode("@ContactName").text = strContactName;
querySuppliers.selectSingleNode("@ContactTitle").text = strTitle;
querySuppliers.selectSingleNode("@Phone").text = strPhone;

// Restore the default table command (for the next time).
XDocument.QueryAdapter.Command = strOrigSQLCommand;

// Switch to data entry view to see results.
XDocument.View.SwitchView("Data Entry");
}
```

Remarque: On peut très bien personnaliser le code en mettant au lieu du *LIKE* la relation d'ordre > ou < ou encore d'autres choses selon les besoins.

Ensuite, quittez l'application *Microsoft Script Editor* et revenez dans votre formulaire.

Activez *Preview Form*:

Vincent ISOZ

Contact Title:

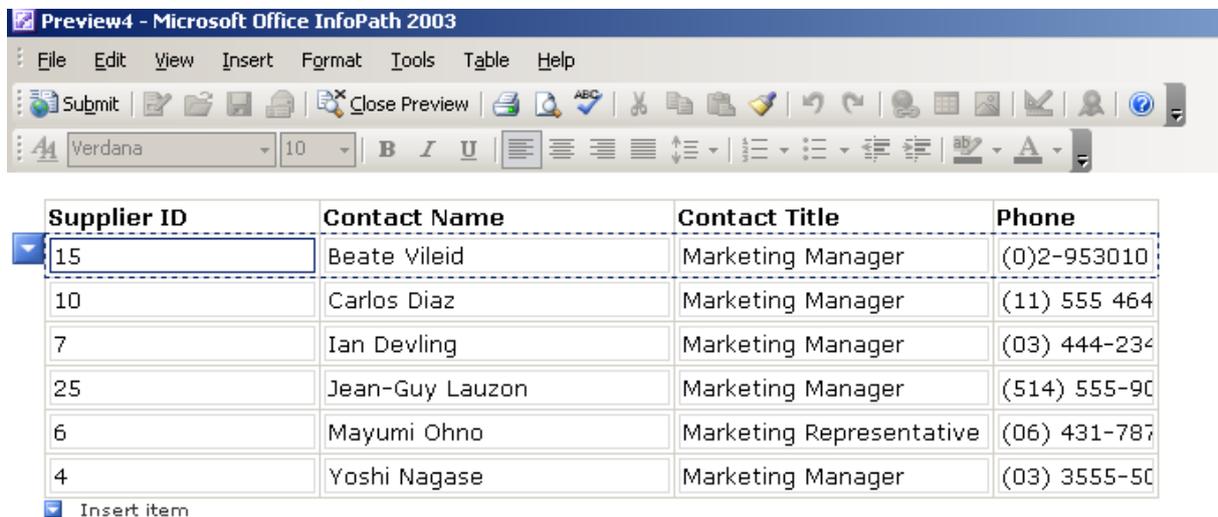
M%

Run Query

Supplier ID	Contact Name	Contact Title	Phone

Insert item

Ensuite saisissez *M%* (l'équivalent InfoPath de *M\** dans Access). Si vous cliquez sur *Run Query* InfoPath change automatiquement de vue et vous aurez:



Preview4 - Microsoft Office InfoPath 2003

File Edit View Insert Format Tools Table Help

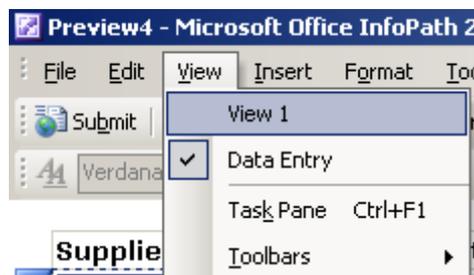
Submit Close Preview

Verdana 10 B I U

Supplier ID	Contact Name	Contact Title	Phone
15	Beate Vileid	Marketing Manager	(0)2-953010
10	Carlos Diaz	Marketing Manager	(11) 555 464
7	Ian Devling	Marketing Manager	(03) 444-234
25	Jean-Guy Lauzon	Marketing Manager	(514) 555-90
6	Mayumi Ohno	Marketing Representative	(06) 431-787
4	Yoshi Nagase	Marketing Manager	(03) 3555-50

Insert item

Pour revenir à la vue précédente:



#### 10.4 Création d'une boîte de dialogue de sauvegarde avec C#

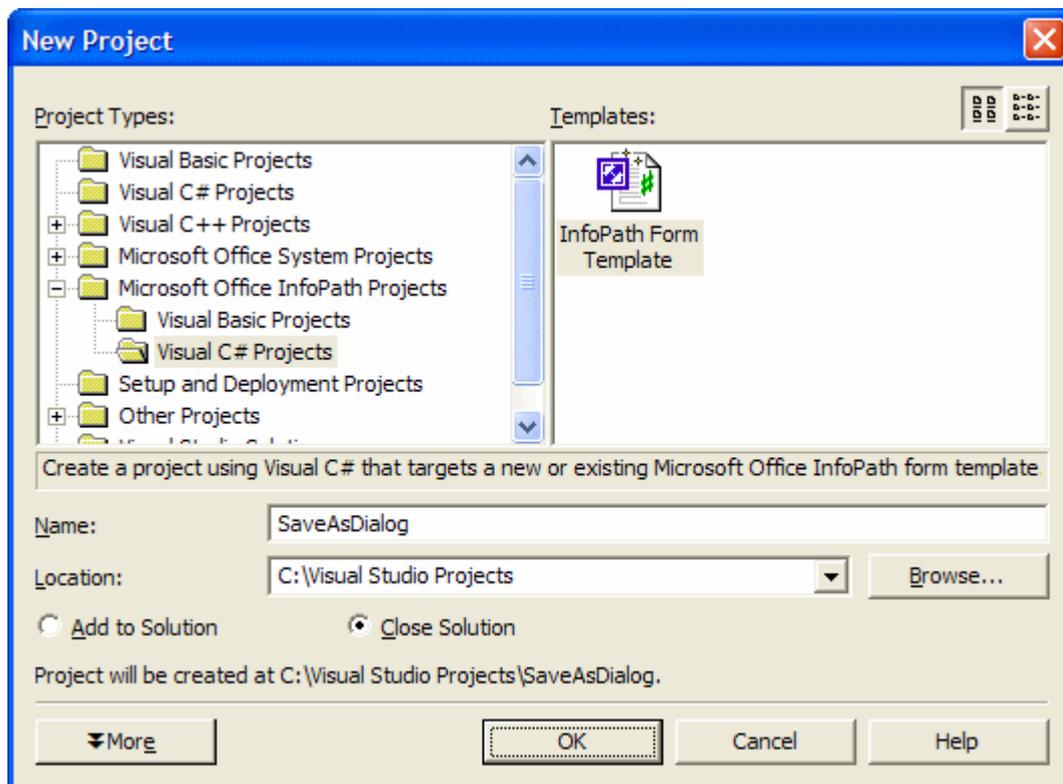
InfoPath ne permet pas à un utilisateur non-informaticien de créer un bouton qui lance la boîte de dialogue de sauvegarde standard de Windows pour soumettre un formulaire. Pour arriver à créer un tel élément il faut passer par le .Net Framework.

Vincent ISOZ

Pour arriver à ce résultat, nous devons avoir au préalable installé le Toolkit InfoPath de Visual Studio .Net et le kit de développement. Par ailleurs, nous utiliserons l'outil *RegForm.exe* pour créer un package d'installation full trut de notre modèle de formulaire

Pour créer un nouveau projet InfoPath:

1. Lancez Visual Studio.
2. Choisissez *File/New/Project*
3. Dans les types de projets proposes prenez celui d'InfoPath version C#
4. Nommez votre projet *SaveAsDialog* et cliquez sur *OK* et ensuite sur *Finish*



Ensuite éditez la vue (InfoPath doit s'être ouvert normalement):

1. Ouvrez le panneau *Controls*
2. Insérez un champ de texte et un bouton dans la vue

Définissons le niveau de sécurité du formulaire comme Full Trust:

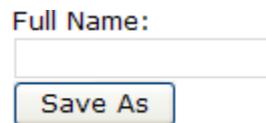
1. Ouvrir *Tools/Form Options*
2. Dans l'onglet *Security* désactivez l'option *Automatically Determine Security Level* et activez à la place *Full Trust* et validez par *OK*.

Nous allons maintenant préparer le bouton *Save As*:

1. Double cliquez sur le bouton

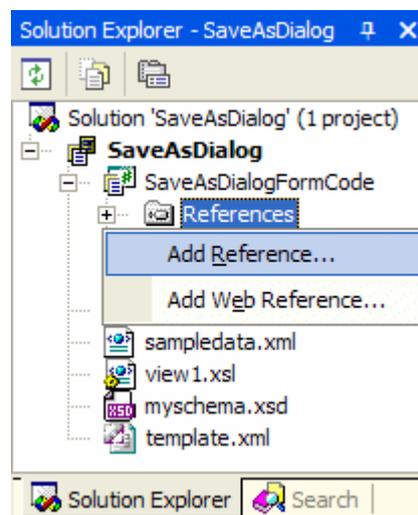
Vincent ISOZ

2. Dans les propriétés du bouton changez son *Label* à *Save As*
3. Changez l'ID de manière à avoir *btnSaveAs* (comme montré sur la figure suivante)
4. Cliquez sur le bouton *Edit Form Code*

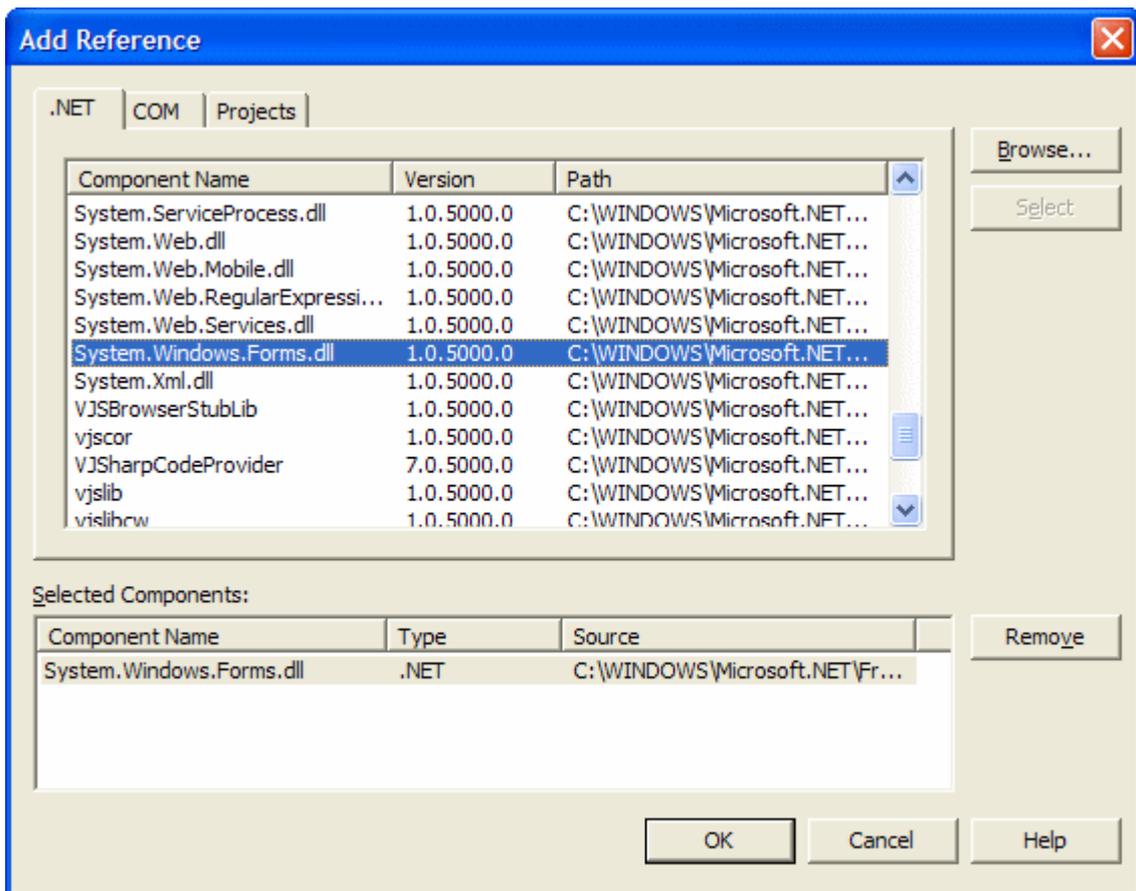


Normalement vous arrivez alors dans VS.NET. Il nous faut maintenant rajouter une référence:

1. Dans l'explorateur de solution, faites un clic droit sur *References* et choisissez *Add Reference*



2. Dans l'onglet *.Net* sélectionnez *System.Windows.Forms.dll*, cliquez sur *Select* et ensuite sur *OK*.



3. Tout en haut du fichier *FormCode.cs* ajouter la ligne de code suivante:

```

FormCode.cs*
SavaAsDialog.SavaAsDialog
using System;
using Microsoft.Office.Interop.InfoPath.SemiTrust;
using System.Windows.Forms;

```

System.Windows.Forms inclus déjà l'objet *Application*. Ce qui fait qu'il va entrer en conflit avec une partie du code déjà incluse dans le formulaire. Pour résoudre ceci il faut:

1. Modifier les lignes comme ci-dessous:

```

public class SavaAsDialog
{
    private XDocument thisXDocument;
    //on rajoute cette ligne
    private Microsoft.Office.Interop.InfoPath.SemiTrust.Application thisApplication;
    //on enlève cette ligne qui peut créer un conflit
    //private Application thisApplication;

```

2. Et la déclaration de la fonction *\_Startup* par:

```
public void _Startup(Microsoft.Office.Interop.InfoPath.SemiTrust.Application app, XDocument doc)
{
    thisXDocument = doc;
    thisApplication = app;
}
```

Ensuite, ajoutons le *Code Behind* du bouton *Save As*:

1. Il faut insérer dans l'événement *btnSaveAs\_OnClick* le code suivant:

```
public void btnSaveAs_OnClick(DocActionEvent e)
{
    string sFullName = thisXDocument.DOM.selectSingleNode("/my:myFields/my:field1").text;
    FileDialog oDialog = new SaveFileDialog();
    oDialog.DefaultExt = "xml";
    oDialog.FileName = "My Form - " + sFullName;
    oDialog.Filter = "InfoPath Form (*.xml) | *.xml";
    if(oDialog.ShowDialog() == DialogResult.OK)
    {
        string sFilename = oDialog.FileName;
        thisXDocument.SaveAs(sFilename);
    }
}
```

Sauvegardez ensuite le modèle de formulaire. Pour ce faire:

1. Retournez dans le designer *InfoPath* designer
2. Dans le menu *File* cliquez sur *Publish*
3. Dans l'assistant *Publishing Wizard* qui apparaît à l'écran cliquez sur *Next*
4. Sélectionnez *To A Shared Folder*, et ensuite cliquez sur *Next*.
5. Enregistrez-le (pour l'exemple) dans *C:\SaveAsDialog.xsn*, et cliquez sur *Next* et ensuite sur *Finish* et enfin sur *Close*.
6. Sortez de Visual Studio

Nous allons maintenant créer un package d'installation *Full trust* de notre fichier sinon quoi il ne pourra être ouvert:

1. Ouvrez le shell CMD
2. Changez de répertoire pour aller au C:\
3. Et tapez la ligne de commande suivante:

```
C:\>regform SaveAsDialog.xsn /MSI /T Yes
```

4. After the successful build of your install package, type the following command:

```
C:\>regform SaveAsDialog.xsn /MSI /T Yes
Microsoft Office InfoPath RegForm tool. Version 1.0.0.2
Copyright (C) Microsoft Corporation 2003. All rights reserved.
Building MSI using devenv...
Result MSI in C:\SaveAsDialog.msi file.
Completed.
```

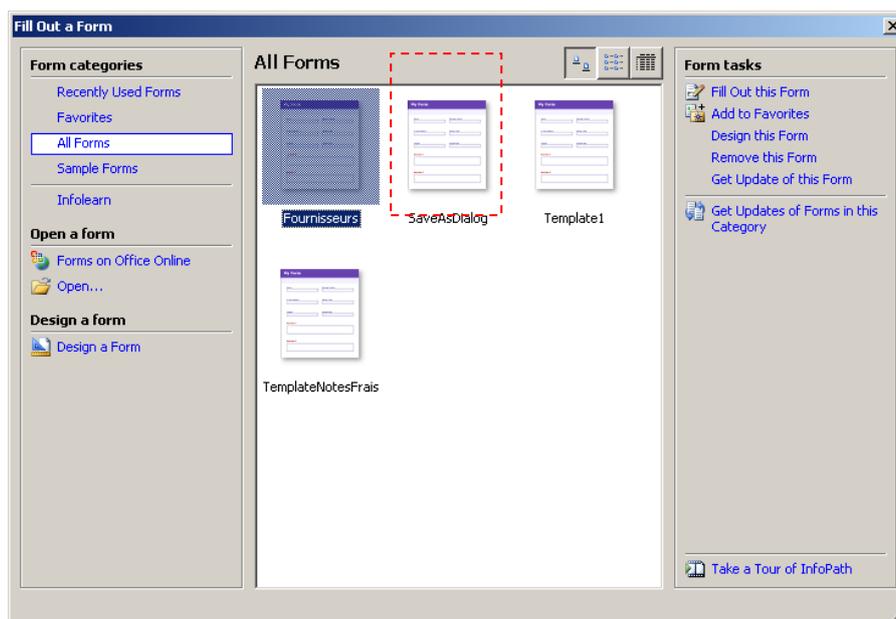
5. Ensuite, toujours dans le CMD tapez:

```
C:\>SaveAsDialog.msi
```

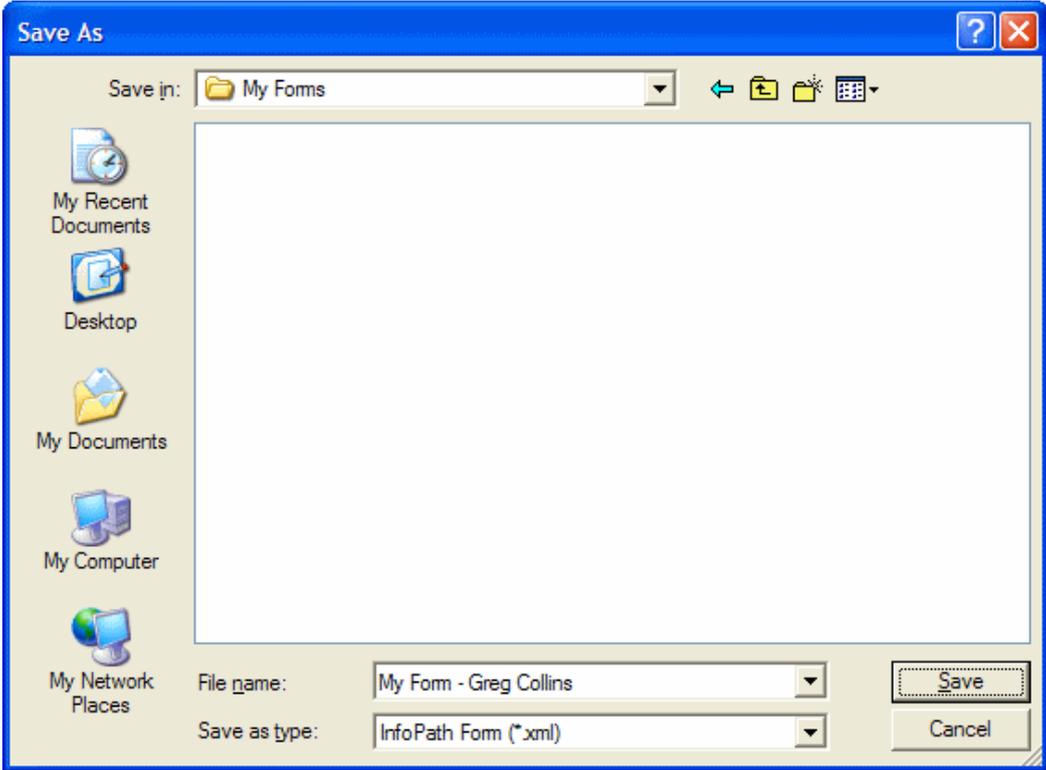
6. Suivez les instructions de la boîte dialogue SaveAsDialog Setup Wizard pour installer le formulaire.

Ensuite, testez si tout fonctionne:

1. Lancez InfoPath.
2. Dans la fenêtre *Fill Out A Form* la fenêtre sélectionnez *SaveAsDialog*



3. Saisissez un texte dans l'unique champ disponible et cliquez ensuite sur *Save As*
4. Vous aurez alors la boîte de dialogue suivante qui apparaîtra à l'écran.



## 10.5 Peupler une liste avec les utilisateurs SharePoint et C#

Depuis le SP1 d'InfoPath il est réellement aisé de créer des listes déroulantes ayant pour source de données des listes SharePoint. Mais remplir une liste SharePoint avec les utilisateurs d'un site particulier reste quelque chose actuellement de relativement compliqué à faire. Mais à peu près rien n'est impossible avec InfoPath donc allons-y!

Pour cet exemple nous allons supposer avoir deux machines:

1. La première sera la machine supportant SharePoint Server
2. La deuxième sera la machine de développement supportant VS.Net 2003

Dans un premier temps, ouvrez InfoPath sur la machine SharePoint et insérez une *drop-down list* et ouvrez sa fenêtre de propriétés.

Sélectionnez l'option *Look up values in a data connection to a database, Web service, file or SharePoint library or list*. Cliquez sur le bouton *Add* pour ajouter une nouvelle source de données.

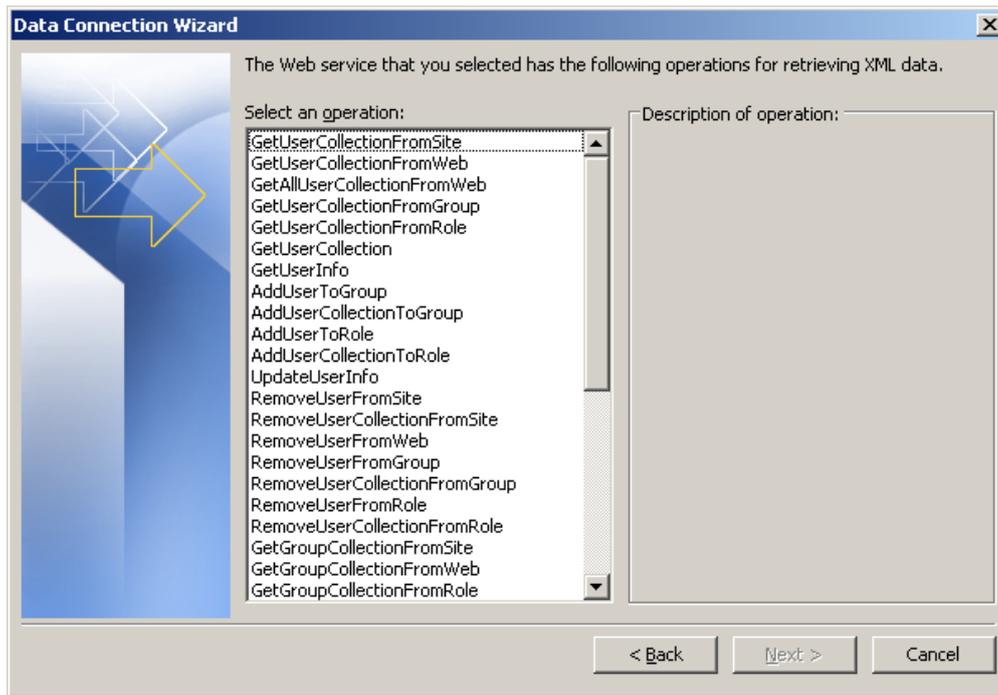
Nous souhaiterions dans cet exemple obtenir des données de la méthode *GetUserCollectionFromSite*, donc dans la boîte de dialogue de l'assistant sélectionnez *Web service*. Ensuite, nous allons utiliser un service WSDL (Web Service Description Language) localisé par une URL du type:

`http://Server_Name/[Site_Name/]_vti_bin/UserGroup.asmx?wsdl`

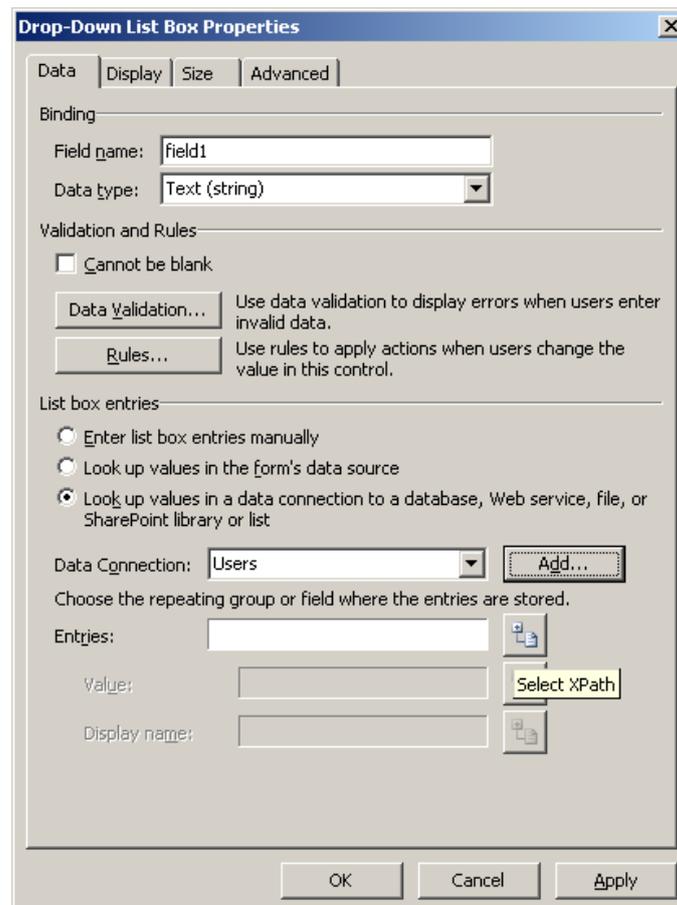
Ainsi, dans notre exemple nous avons un serveur SharePoint nommée *srvsps2003* et le site qui nous intéresse se nomme *SPS2007*.



Quand vous cliquez sur *Next* la liste des méthodes du WS apparaît:

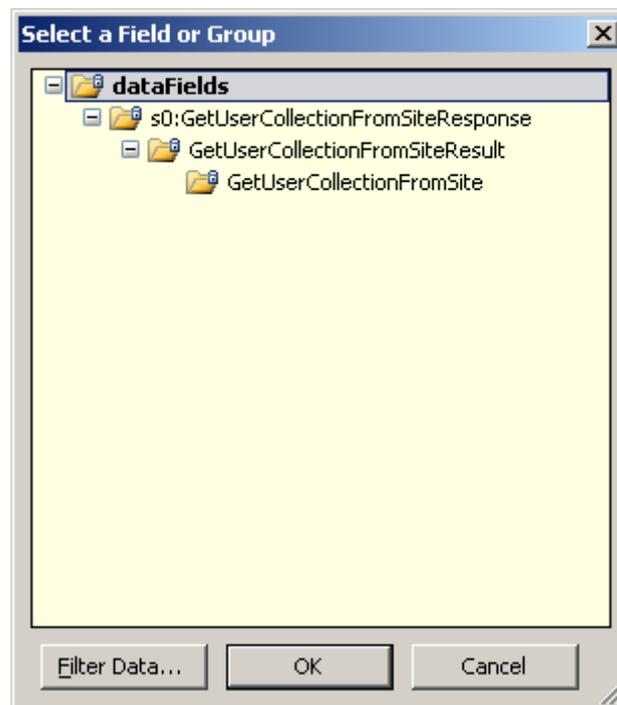


Sélectionnez *GetUserCollectionFromSite* et cliquez sur *Next*. Nommez votre source de données *Users*:



Vincent ISOZ

Lorsque vous tentez de cliquer sur le bouton *XPath* (à droite du champ *Entries*) les problèmes commencent... Effectivement, on voit qu'il n'y a aucun groupe répétitif nécessaire à remplir la liste déroulante:



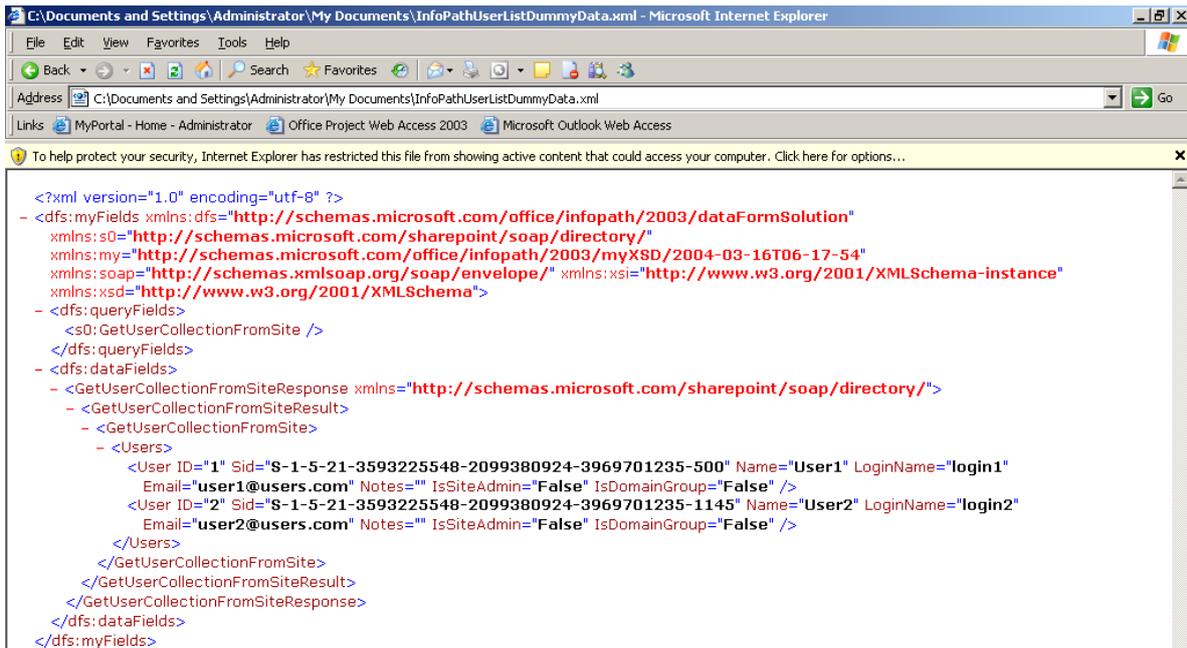
Le problème ici n'en est pas un. C'est simplement une limitation de rôle du WS. Donc soit:

1. On développe une solution .Net
2. On édite manuellement la source XSLT du formulaire InfoPath pour accéder directement au XPath
3. Ou on utilise une astuce...

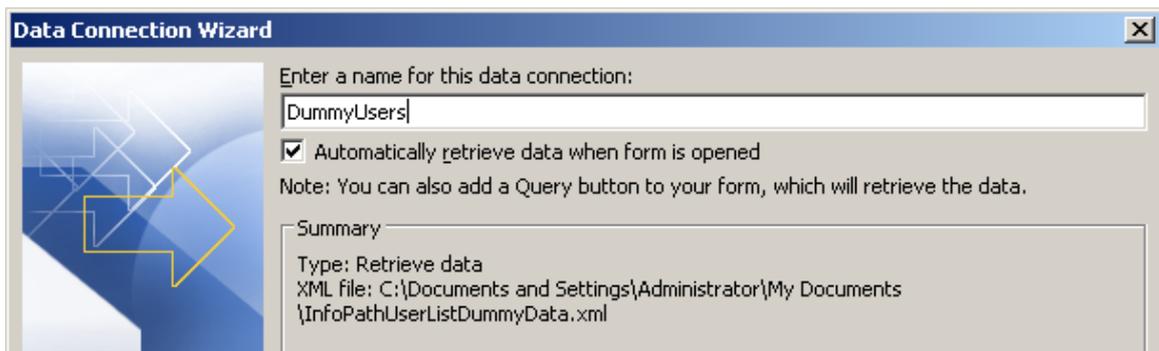
La troisième solution consiste à utiliser une source XML bidon (étalon) du webservice que l'on peut obtenir avec des outils comme WebServiceStudio 2.0:

<http://www.gotdotnet.com/Community/UserSamples/Details.aspx?SampleGuid=65a1d4ea-0f7a-41bd-8494-e916ebc4159c>

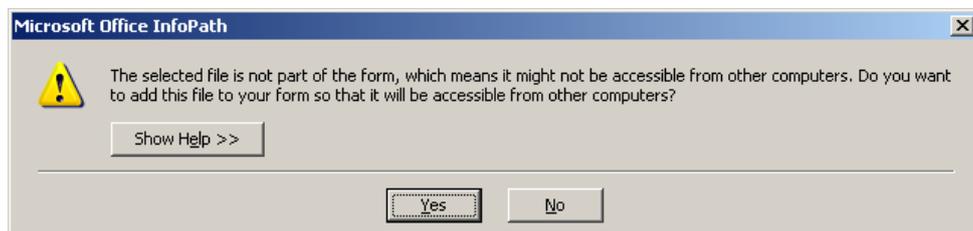
Cet outil nous donnerait un fichier XML typique du genre que nous nommerons *InfoPathUserListDummyData.xml*:



Cliquez sur *Add...* (pour ajouter une nouvelle source de données) et sélectionnez une source *XML* et pointez sur le fichier XML que nous avons préparé tout à l'heure. Nommez cette source *DummyUsers*:

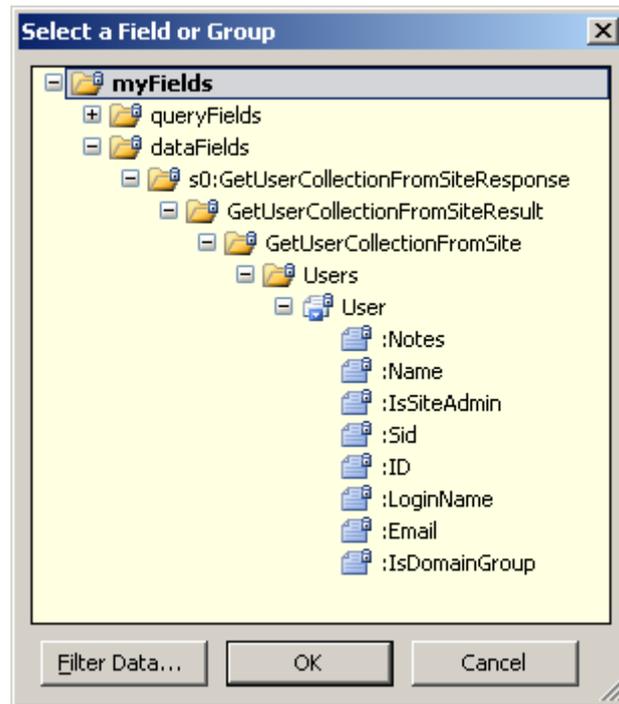


Prenez garde à désactiver la coche *Automatically retrieve data when form is opened* et répondez à la question:

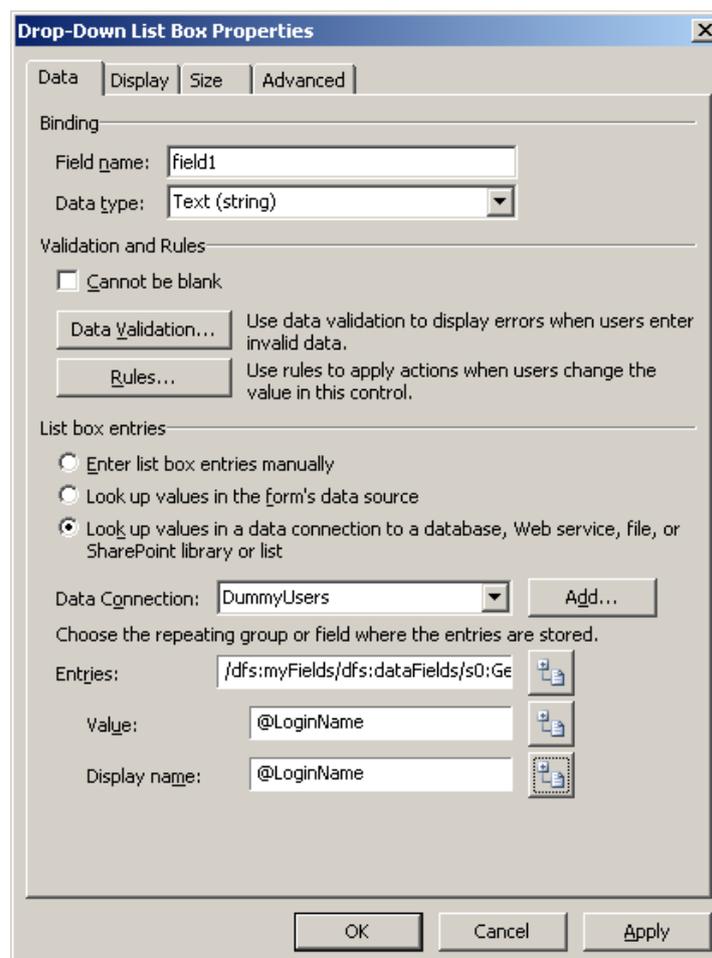


par *No*.

Ensuite, cliquez à nouveau sur le bouton *XPath* à droit du champ *Entries*:



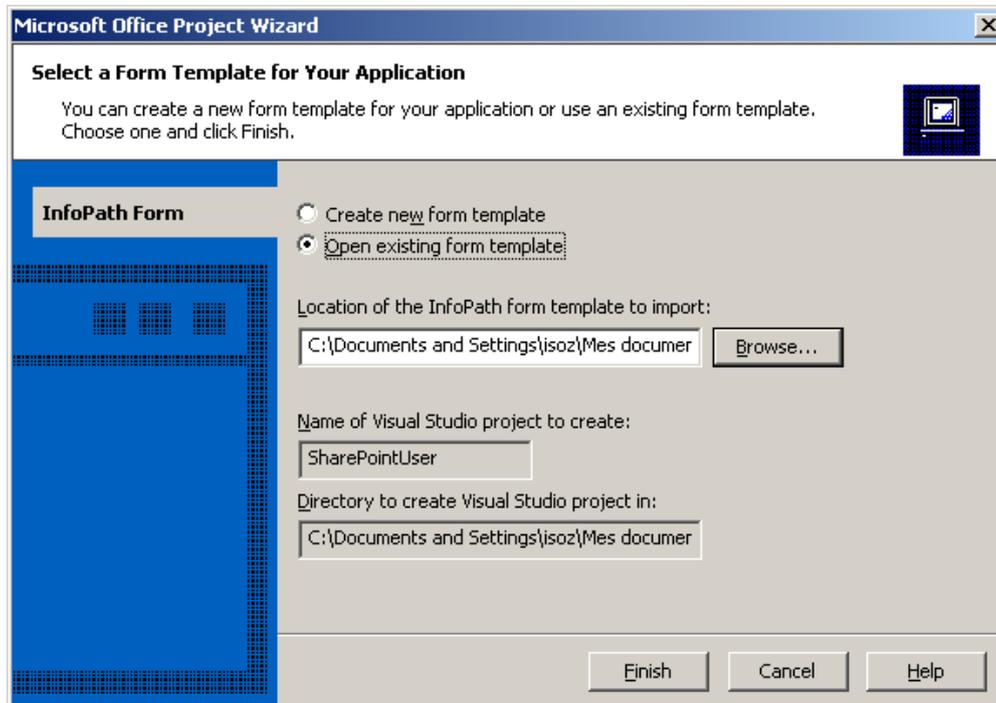
Sélectionnez *User* et ensuite prenez les valeurs *LoginName*:



Vincent ISOZ

Enregistrez votre formulaire sous le nom *SharePointUser.xsn* et passez-le ensuite sur la machine de développement.

Créez un nouveau projet .Net InfoPath en C#. Lors de la création du projet, VS.Net va vous demander si vous désirez utiliser un \*.xsn existant... pointez celui créé précédemment sur le serveur SharePoint et qui se trouve maintenant sur la machine de dev.:



Une fois dans VS.Net, allez dans le formulaire InfoPath qui s'est ouvert en même temps que la création du projet .Net et allez dans le menu *Tools/Programming/On Load Event*:



VS:Net va s'ouvrir. Il vous faudra ajouter le code suivant:

```

FormCode.cs*
SharePointUser.SharePointUser
OnLoad(DocReturnEvent e)

{
    thisXDocument = doc;
    thisApplication = app;

    // You can add additional initialization code here.
}

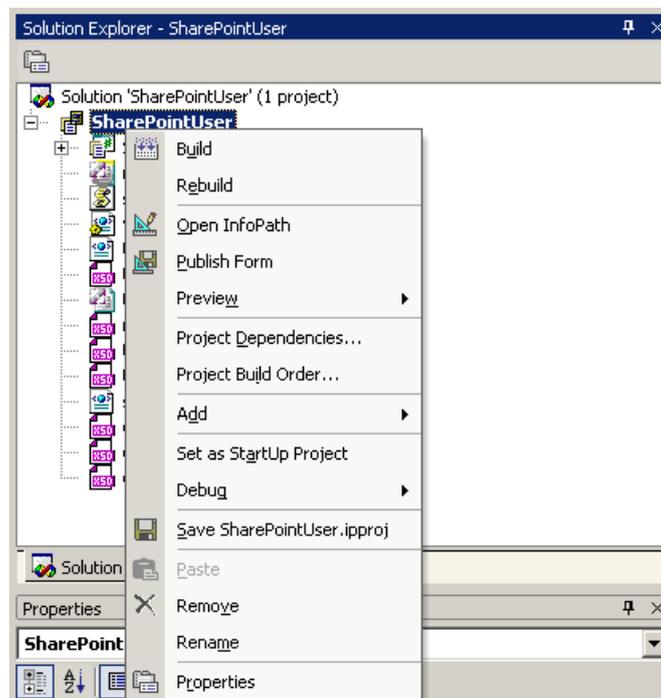
public void _Shutdown()
{
}

// The following function handler is created by Microsoft Office InfoPath. Do not
// modify the type or number of arguments.
[InfoPathEventHandler (EventType=InfoPathEventType.OnLoad) ]
public void OnLoad(DocReturnEvent e)
{
    // Write your code here.
    thisXDocument.DataObjects["DummyUsers"].DOM.loadXML(thisXDocument.DataObjects["Users"].DOM.xml);
}
    
```

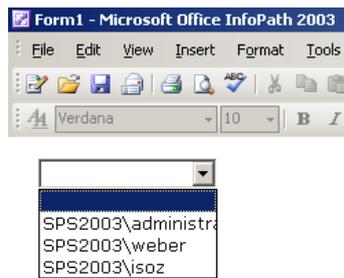
qui va remplacer au chargement du formulaire la source *DummyUsers* par la source *Users*.

Remarque: Prenez bonne note de cette commande car elle est très souvent utilisée!

Une fois ceci fait, déployez sur les machines client le formulaire avec l'outil *Regform* que nous avons déjà utilisé dans l'exemple précédant ou publiez le sur une librairie de formulaire dans SharePoint en faisant un clic droit sur la solution dans VS.Net et de choisir *Publish Form*:



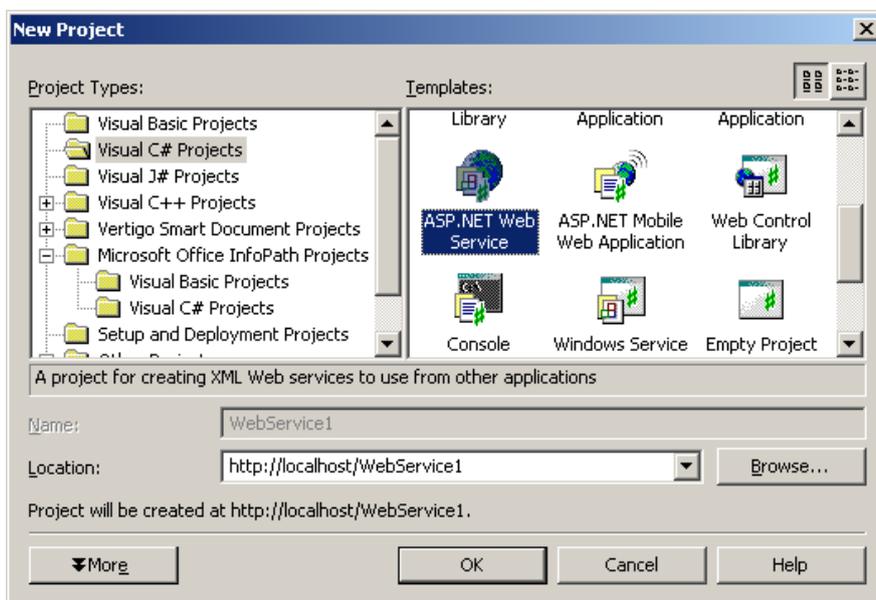
Quand le formulaire est ouvert, nous avons alors:



## 10.6 Utilisation d'un webservice simple en C#

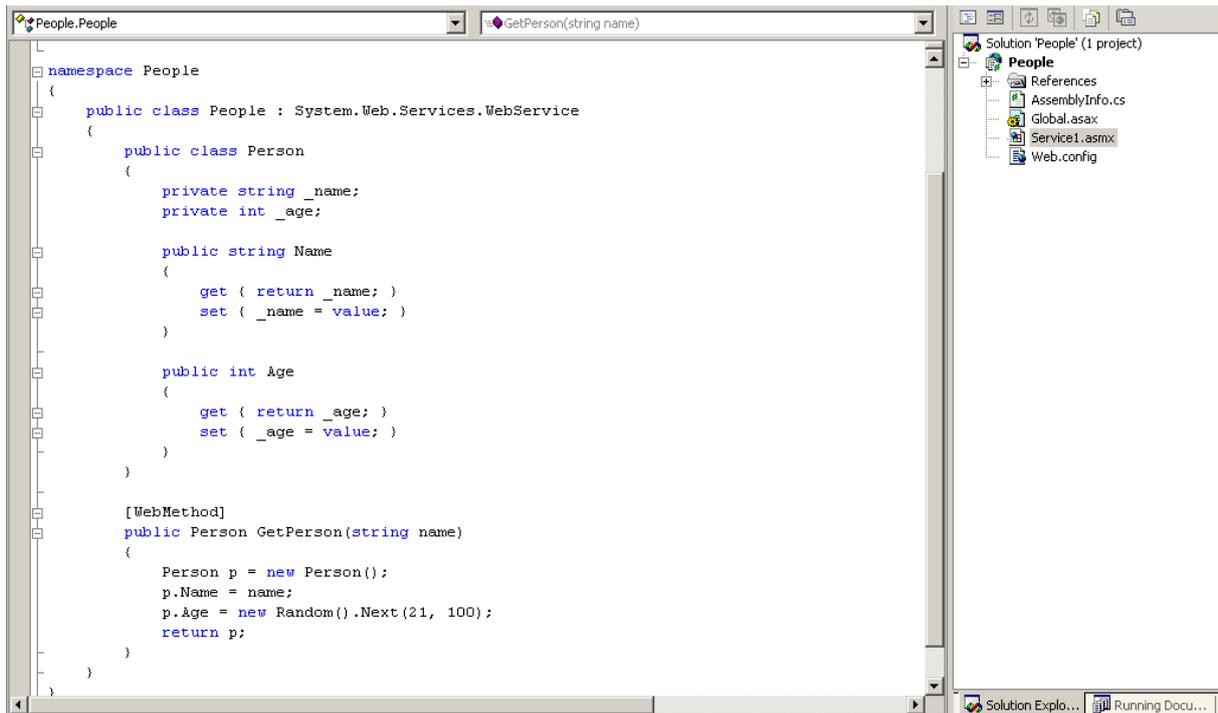
Nous allons faire ici un exemple d'utilisation d'un webservice absolument inutile pour présenter la manière de les utiliser (et développer) avec InfoPath. Cet exemple permettra de préparer l'étude de cas plus complexes.

Ouvrez VS.Net 2003 et allez dans *File/New Project* et choisissez:



Nommez ce webservice *People* et par défaut laissez-le pour l'exemple sur le serveur local IIS (*localhost*).

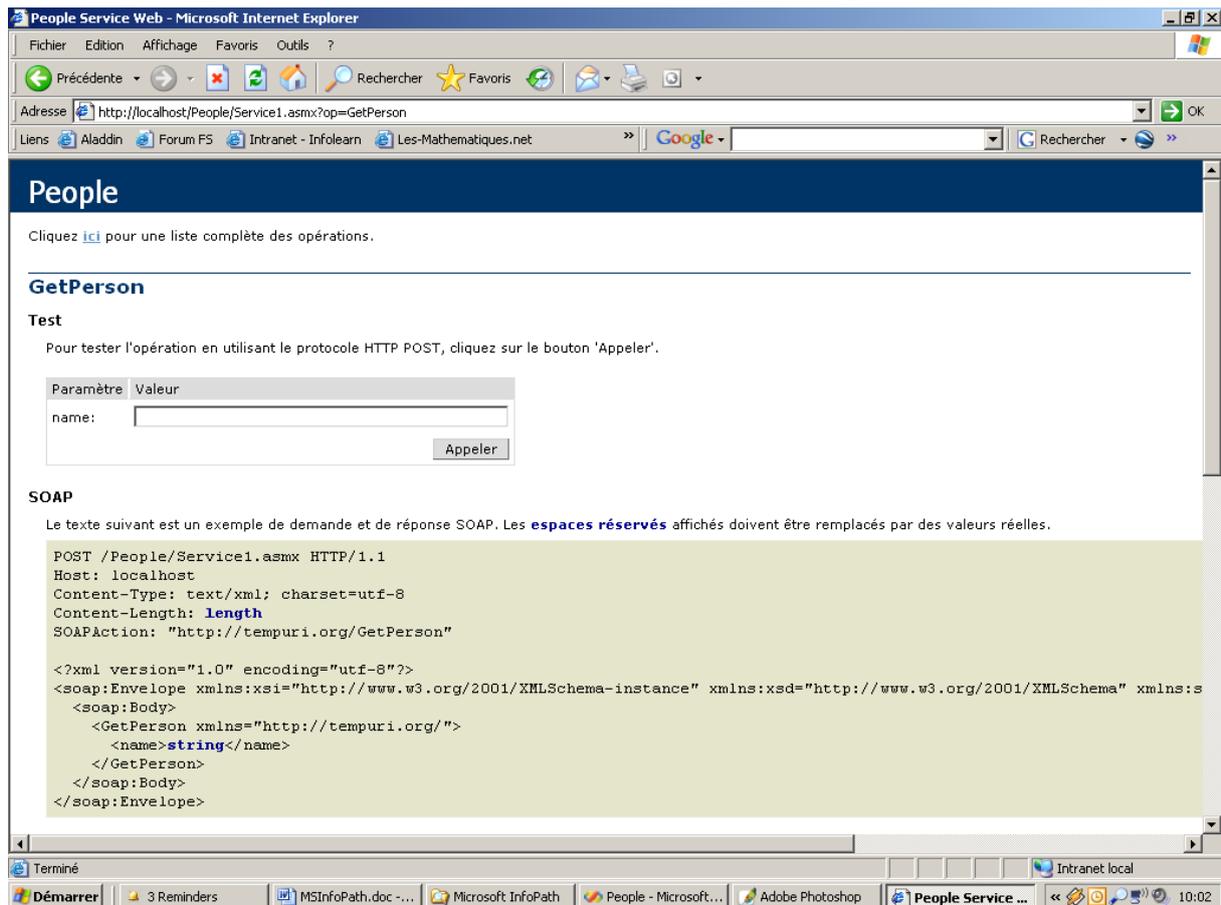
Mettez le code suivant dans *Service1.aspx* (au besoin renommez-le):



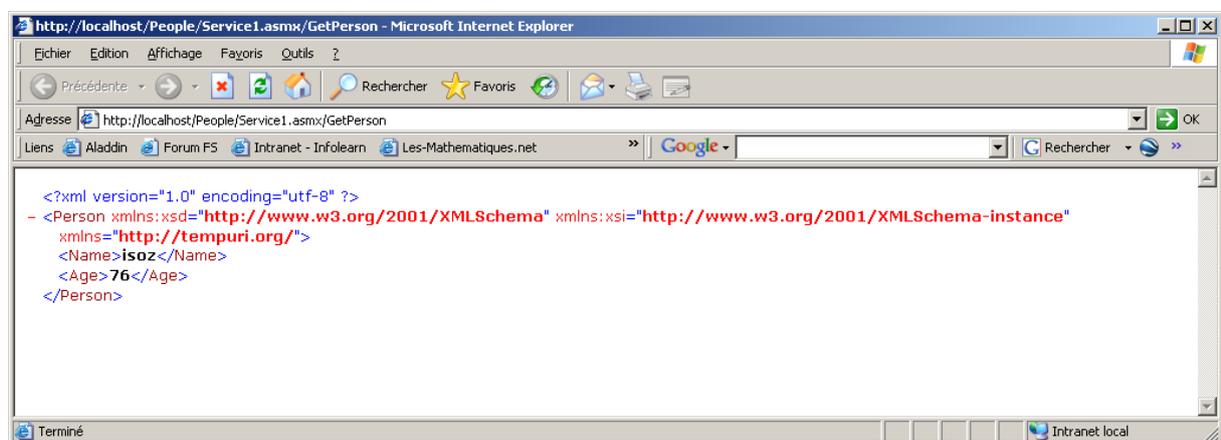
Ensuite, cliquez sur  pour lancer le code. Vous devriez avoir IE qui s'ouvre avec:



et si vous cliquez sur *GetPerson*:



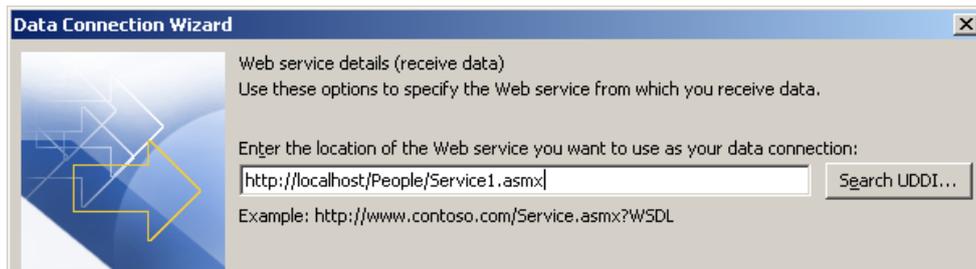
Saisissez un nom quelconque dans le champ *Name* et cliquez sur *Appeler*. Vous obtiendrez:



Maintenant, nous allons utiliser ce WebService dans un formulaire InfoPath.

Créez un nouveau formulaire InfoPath en mode *Design* en choisissant *New from Data Connection...* et dans l'assistant choisissez que la source provienne d'un *Web Service* et cliquez sur *Next* et ensuite choisissez *Receive and submit data* et ensuite cliquez sur *Next*:

Saisissez l'URL de votre web service:



Cliquez sur *Next* et ensuite sélectionnez la méthode *GetPerson* et cliquez sur *Next*:

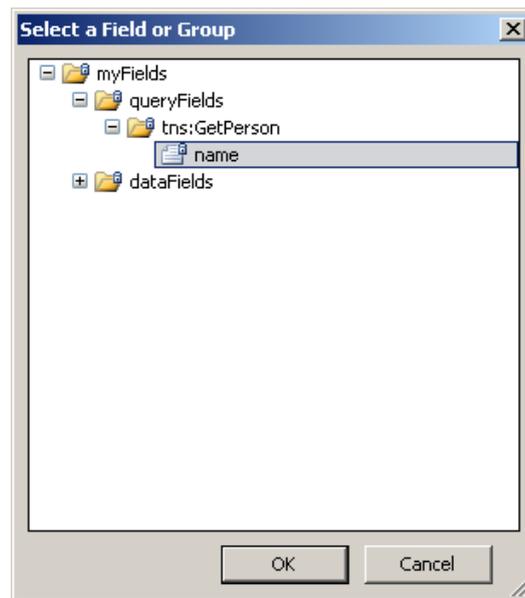


Cliquez sur *Next* deux fois et ensuite sur *Finish* (nous laissons le nom standard de la requête *Main query*).

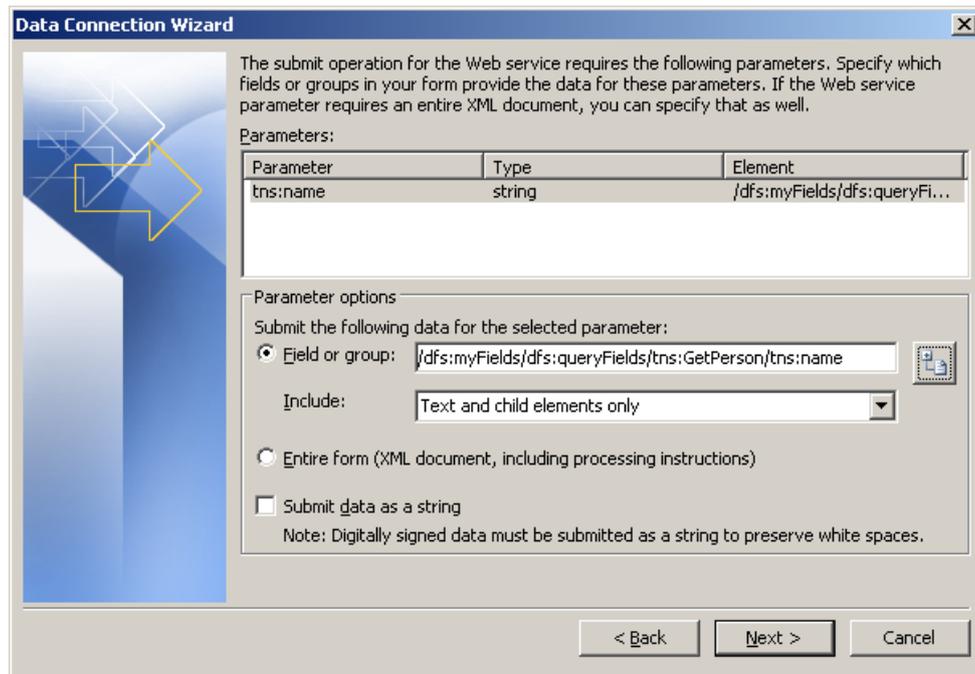
Cliquez sur *Next* jusqu'à arriver à la boîte de dialogue suivante:



Sélectionnez l'option *Field or group* et cliquez sur le bouton à droite du champ et sélectionnez-y:



et validez par *OK*. Vous aurez alors:



et activez la case *Submit data as string*. Cliquez enfin sur *Next* et *Finish*. Laissez le nom par défaut pour la source de donnée (*Main submit*).

Le formulaire suivant apparaît:

## Vincent ISOZ

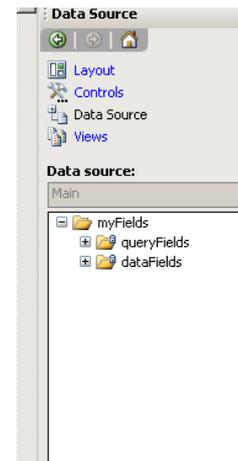
Click to add a title

Click to add form content

Drag query fields here

Run Query

Drag data fields here



Glissez maintenant le champ *name* du dossier *queryFields* dans la zone *Drag query fields here*. De même, sélectionnez les champs *Name* et *Age* du dossier *dataFields* et glissez-les dans la zone *Drag data fields here*.

Vous obtiendrez:

Click to add a title

Click to add form content

Name:

Run Query

Name:

Age:

et testez votre formulaire en passant en mode *Preview Form*.

---

Name:

ISOZ

Run Query

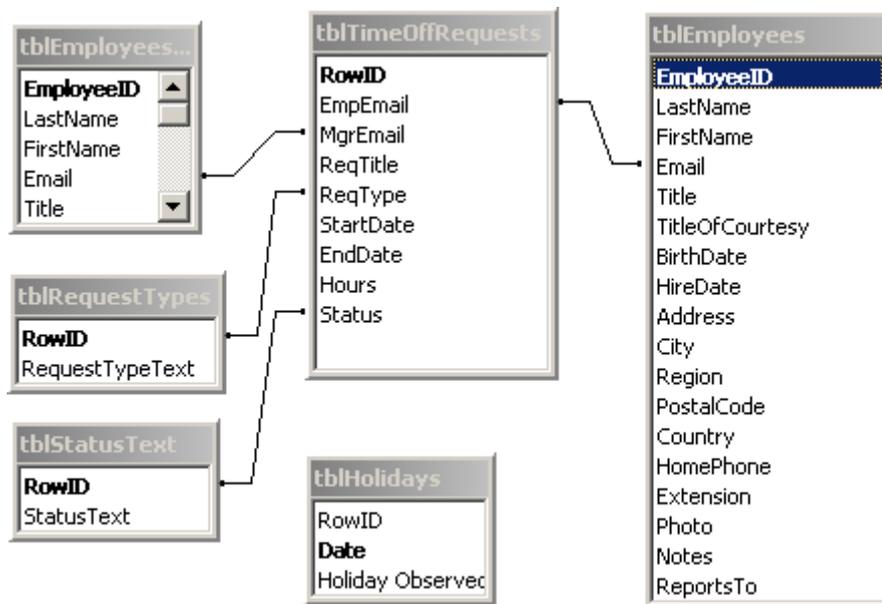
Name: ISOZ

Age: 61

## 10.7 Utilisation d'un webservice en C#

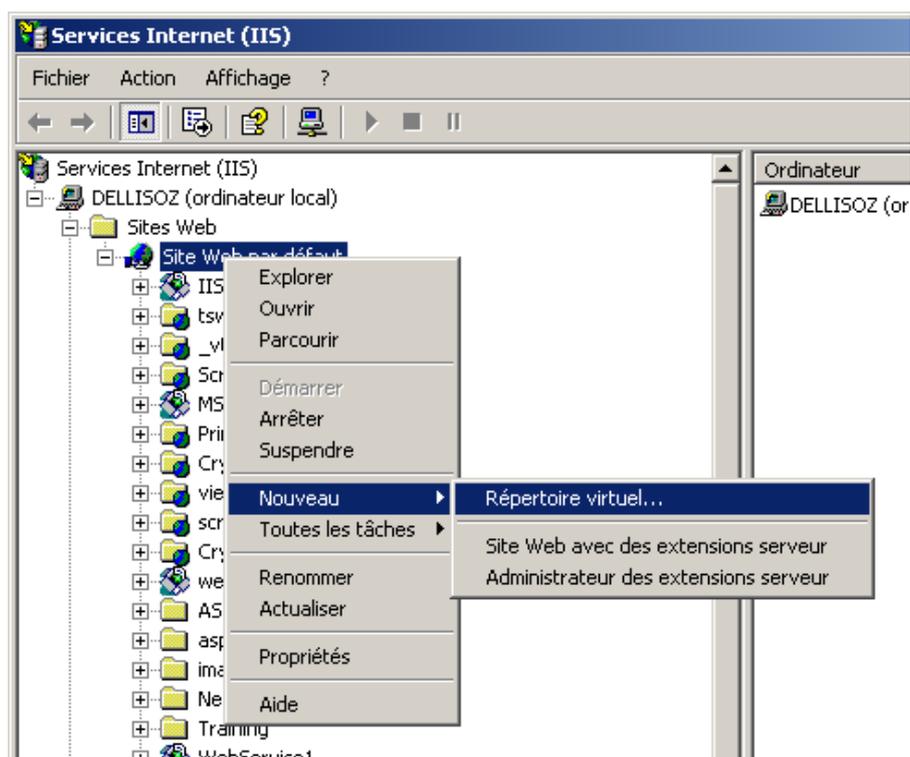
Dans cet exemple, nous allons faire usage d'une base de données MS Access de gestion des absences et créer un webservice qui accède en lecture/écriture à celle-ci.

La base de données nommée *TimeOff.mdb* a le schéma suivant:



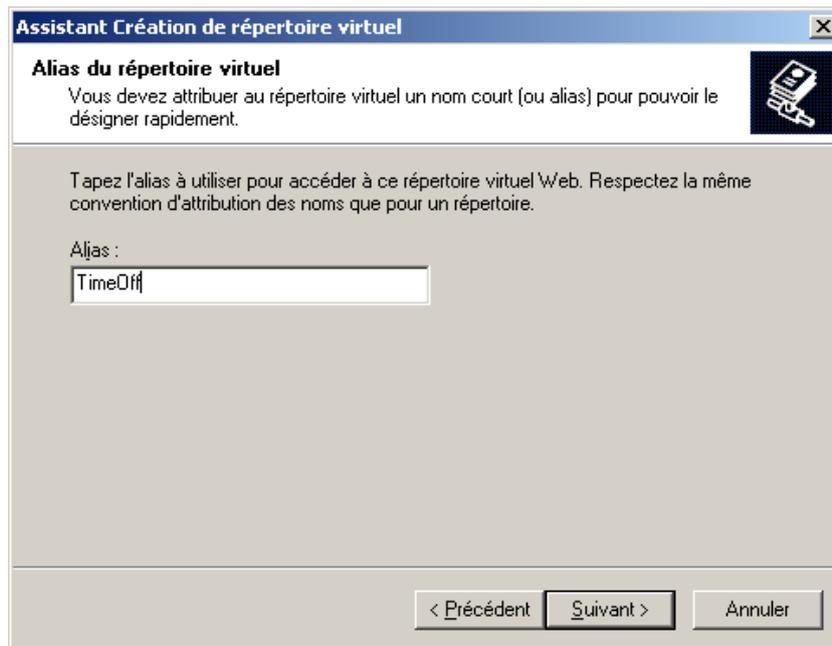
Posez cette base de données à la racine du disque c:\ dans un dossier spécial que nous nommerons *TimeOff*.

Ensuite, dans IIS créez un nouveau répertoire virtuel (ce répertoire va se créer automatiquement dans le dossier *InetPut/wwwroot*):

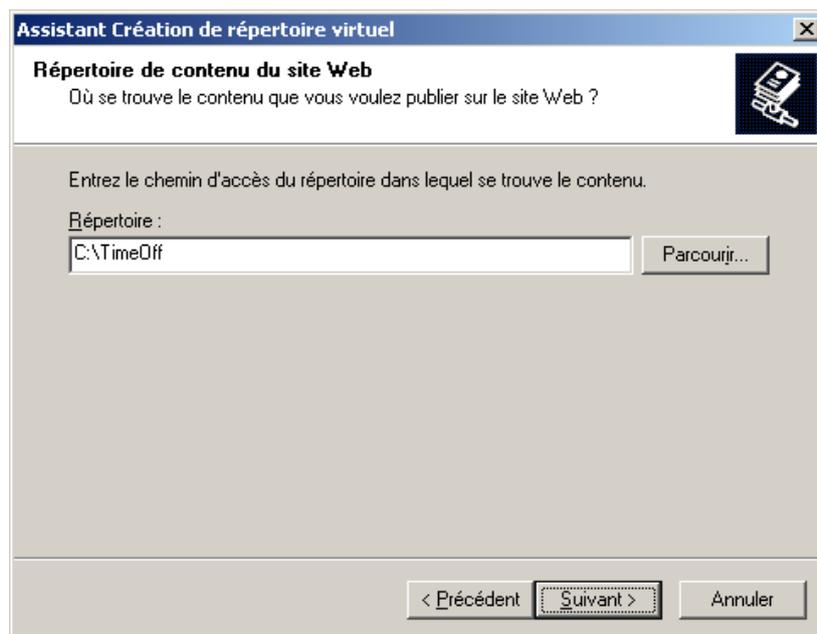


Vincent ISOZ

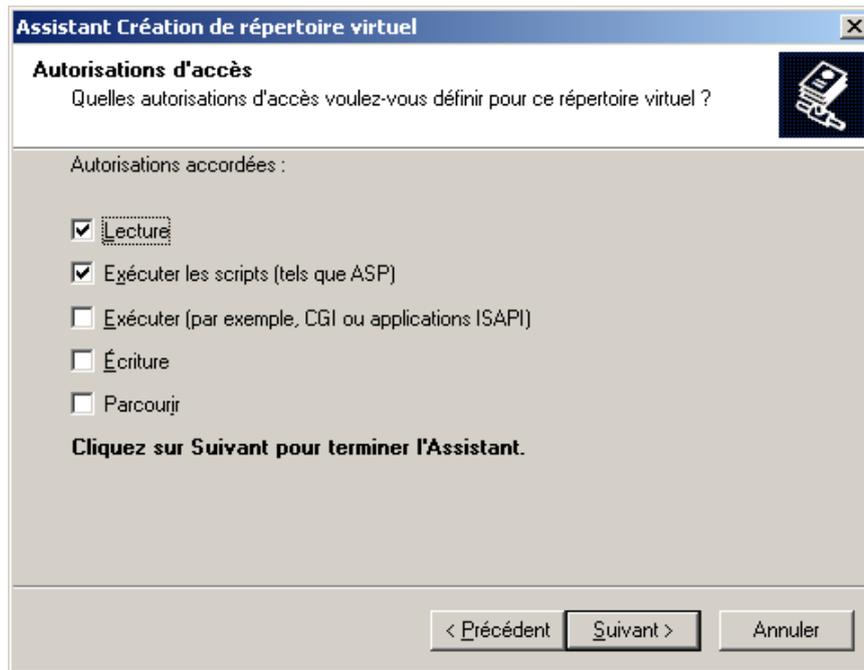
Lors de l'assistant de création, nommez ce répertoire virtuel *TimeOff*:



et sélectionnez comme répertoire du contenu du site web, le répertoire créé précédemment:



Laissez les propriétés par défaut en ce qui concerne les droits:



Maintenant, créez un nouveau projet C# de type Webservice que nous nommerons *TimeOff*. Lorsque ce projet est créé, éditez le fichier *web.config* pour y mettre:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <appSettings>
    <add key="DBConnStr" value="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=c:\TimeOff\TimeOff.mdb" />
  </appSettings>
</configuration>
```

Ce fichier, et en particulier la variable *DBConnStr* sera utilisée par notre webservice pour aller chercher des données à l'endroit où se situe la source de données *TimeOff.mdb*.

**Attention!!!** Posez le fichier *web.config* à la racine du *wwwroot*!

Renommez maintenant le fichier *Service1.asmx* en *DataService.asmx* et saisissez-y le code suivant:

```

using System;
using System.Xml;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Configuration;
using System.IO;
using System.Data;
using System.Data.OleDb;

namespace TimeOff2003
{
    [WebService (Namespace="http://Office2003/TimeOff2003",
        Description="TimeOff 2003 Data Web service to be used by InfoPath.")]
    public class DataService : WebService
    {
        private string DBConnectionString;

        public DataService()
        {
            DBConnectionString = ConfigurationSettings.AppSettings["DBConnStr"];
        }

        [WebMethod (Description="Get Employees list.")]
        public XmlNode GetEmpList()
        {
            return GetXML("tblEmployees", "FirstName + ' ' + LastName as EmpName, Email", "");
        }

        [WebMethod (Description="Get absence request types.")]
        public XmlNode GetRequestTypes()
        {
            return GetXML("tblRequestTypes", "", "");
        }

        [WebMethod (Description="Get request status text and codes.")]
        public XmlNode GetStatusText()
        {
            return GetXML("tblStatusText", "", "");
        }

        [WebMethod (Description="Insert a new TimeOff request.")]
        public bool InsertRequest(string TimeOffDetailsXML)
        {
            bool bRetVal = true;
            try
            {
                DataSet TimeOffRequestsDS = new DataSet();
                OleDbDataAdapter Adapter = new OleDbDataAdapter("SELECT * FROM tblTimeOffRequests WHERE 1 = 2",
                    DBConnectionString);

                Adapter.Fill(TimeOffRequestsDS);
                OleDbCommandBuilder cmdBuilder;
                cmdBuilder = new OleDbCommandBuilder(Adapter);

                DataRow newRow = TimeOffRequestsDS.Tables[0].NewRow();

                XmlDocument doc = new XmlDocument();
                doc.LoadXml(TimeOffDetailsXML);

                XmlNamespaceManager nsmgr = new XmlNamespaceManager(doc.NameTable);
                nsmgr.AddNamespace("my", "http://schemas.microsoft.com/office/infopath/2003/myXSD/2003-04-20T21:29:06");
                newRow["EmpEmail"] = doc.SelectSingleNode("//my:EmpEmail", nsmgr).InnerText;
                newRow["MgrEmail"] = doc.SelectSingleNode("//my:MgrEmail", nsmgr).InnerText;
                newRow["ReqTitle"] = doc.SelectSingleNode("//my:ReqTitle", nsmgr).InnerText;
                newRow["ReqType"] = Convert.ToInt32(doc.SelectSingleNode("//my:ReqType", nsmgr).InnerText);
                newRow["StartDate"] = doc.SelectSingleNode("//my:StartDate", nsmgr).InnerText;
                newRow["EndDate"] = doc.SelectSingleNode("//my:EndDate", nsmgr).InnerText;
            }
            catch { }
        }
    }
}

```



tblEmployees : Table										
Employee ID	Last Name	First Name	Email	Title	Title Of Courtesy	Birth Date	Hire Date	Address		C
1	Davolio	Nancy	nancyd	Sales Representative	Ms.	08-déc-1968	01-mai-1992	507 - 20th Ave. E.		Seattl
2	Fuller	Andrew	andrewf	Vice President, Sales	Dr.	19-févr-1952	14-août-1992	908 W. Capital Way		Tacor
3	Leverling	Janet	janetl	Sales Representative	Ms.	30-août-1963	01-avr-1992	722 Moss Bay Blvd.		Kirkla
4	Peacock	Margaret	margaretp	Sales Representative	Mrs.	19-sept-1958	03-mai-1993	4110 Old Redmond Rd.		Redm
5	Buchanan	Steven	stevenb	Sales Manager	Mr.	04-mars-1955	17-oct-1993	14 Garrett Hill		Londc
6	Suyama	Michael	michaels	Sales Representative	Mr.	02-juil-1963	17-oct-1993	Coventry House		Londc
7	King	Robert	robertk	Sales Representative	Mr.	29-mai-1960	02-janv-1994	Edgeham Hollow		Londc
8	Callahan	Laura	laurac	Inside Sales Coordinator	Ms.	09-janv-1958	05-mars-1994	4726 - 11th Ave. N.E.		Seattl
9	Dodsworth	Anne	anned	Sales Representative	Ms.	02-juil-1969	15-nov-1994	7 Houndstooth Rd.		Londc
*	(AutoNumber)									

tblHolidays : Table		
RowID	Date	Holiday Observed
1	1/1/2003	New Years Day
2	5/26/2003	Memorial Day
3	7/4/2003	Independence Day
4	9/1/2003	Labor Day
5	11/27/2003	Thanksgiving Day
6	11/28/2003	Day after Thanksgiving
7	12/24/2003	Christmas Eve Day
8	12/25/2003	Christmas Day
9	12/31/2003	New Year's Eve Day
*	0	

tblRequestTypes : Table	
RowID	RequestTypeText
+	1 Compensation Time
+	2 Funeral
+	3 Floating Holiday
+	4 Other
+	5 Sick
+	6 Unpaid
+	7 Vacation
▶	(AutoNumber)

tblStatusText : Table	
RowID	StatusText
+	1 Pending
+	2 Approved
+	3 Rejected
+	4 Cancelled
*	(AutoNumber)

tblTimeOffRequests : Table									
RowID	EmpEmail	MgrEmail	ReqTitle	ReqType	StartDate	EndDate	Hours	Status	
▶	(AutoNumber)			0			0	1	

Le webservice exécuté directement dans IE donne:

## DataService

TimeOff 2003 Data Web service to be used by InfoPath.

Les opérations suivantes sont prises en charge. Pour une définition formelle, prenez connaissance de [Description du service](#).

- [GetRequestTypes](#)  
Get absence request types.
- [GetEmpList](#)  
Get Employees list.
- [InsertRequest](#)  
Insert a new TimeOff request.
- [GetStatusText](#)  
Get request status text and codes.

Si l'on clique sur *GetEmpList* on obtient:

DataService

Cliquez [ici](#) pour une liste complète des opérations.

### GetEmpList

Get Employees list.

**Test**

Pour tester l'opération en utilisant le protocole HTTP POST, cliquez sur le bouton 'Appeler'.

**SOAP**

Le texte suivant est un exemple de demande et de réponse SOAP. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /NetAccess/DataService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Office2003/TimeOff2003/GetEmpList"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:s="http://Office2003/TimeOff2003">
  <soap:Body>
    <GetEmpList xmlns="http://Office2003/TimeOff2003" />
  </soap:Body>
</soap:Envelope>
```

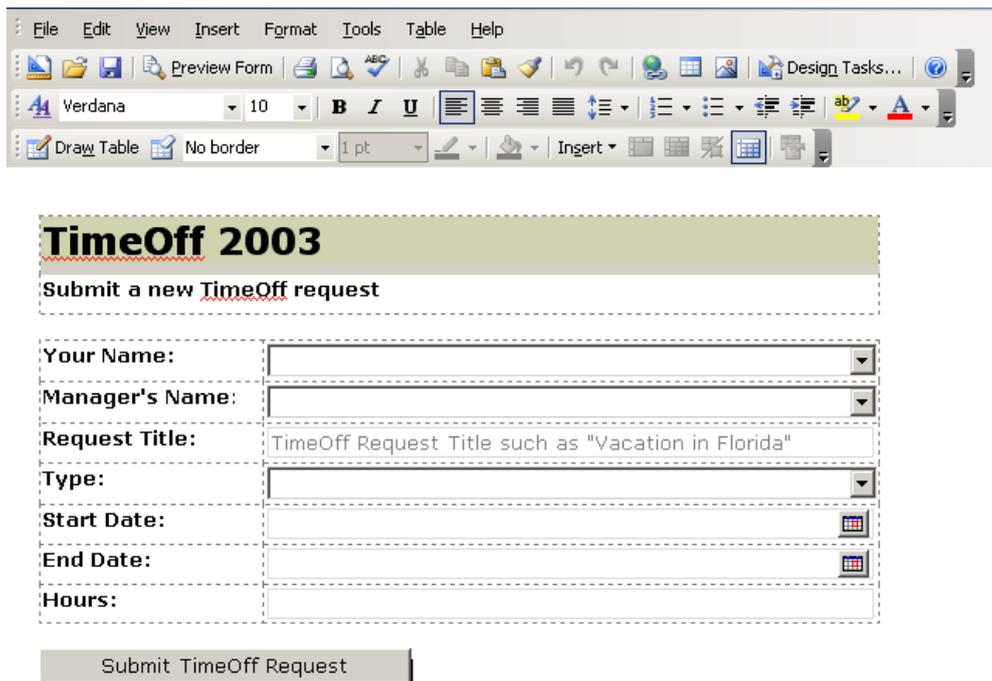
```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
```

et si l'on clique sur *Appeler*:

```
<?xml version="1.0" encoding="utf-8" ?>
- <NewDataSet>
  - <Table>
    <EmpName>Nancy Davolio</EmpName>
    <Email>nancyd</Email>
  </Table>
  - <Table>
    <EmpName>Andrew Fuller</EmpName>
    <Email>andrewf</Email>
  </Table>
  - <Table>
    <EmpName>Janet Leverling</EmpName>
    <Email>janetl</Email>
  </Table>
  - <Table>
    <EmpName>Margaret Peacock</EmpName>
    <Email>margaretp</Email>
  </Table>
  - <Table>
```

Ainsi, on voit que tout fonctionne parfaitement.

Maintenant créez un formulaire nommé *TimeOff.xsn* équivalent à celui-ci:



et enregistrez-le au même emplacement que le webservice lui-même (donc normalement dans le dossier *c:/TimeOff*).

Maintenant, nous souhaiterions avoir un petit plus...: une page web qui dans le volet Office d'InfoPath affiche à l'utilisateur les jours de vacances officiels.

Pour ce faire, nous allons créer une page aspx en VB.Net (juste pour mélanger...) nommée *Holidays.aspx* qui sera créée dans le même dossier que là où se situe notre fichier *xsn*:

```
Holidays.aspx*
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>
<%@ Import Namespace="System.Configuration" %>
<html>
<script language="VB" runat="server">
    Sub Page_Load(Sender As Object, E As EventArgs)

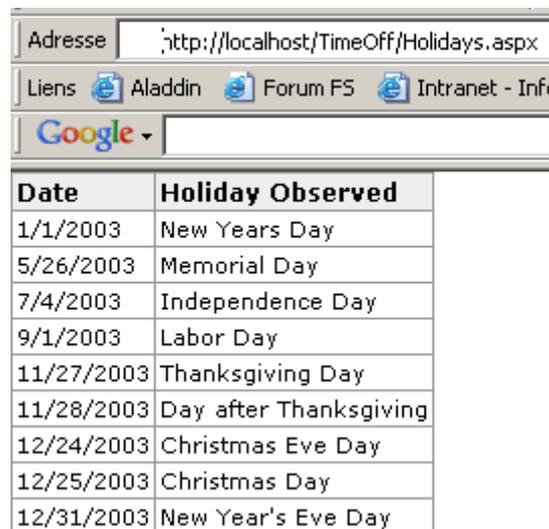
        Dim DS As DataSet
        Dim MyConnection As OleDbConnection
        Dim MyCommand As OleDbDataAdapter

        MyConnection = New OleDbConnection(ConfigurationSettings.AppSettings("DBConnStr"))
        MyCommand = New OleDbDataAdapter("SELECT [Date], [Holiday Observed] FROM [tblHolidays] " & _
            "ORDER BY [ROWID] ASC", _
            MyConnection)

        DS = New DataSet()
        MyCommand.Fill(DS, "Holidays")

        MyGrid.DataSource = DS.Tables("Holidays").DefaultView
        MyGrid.DataBind()
    </script>
</body topmargin="0" leftmargin="0" marginwidth="0" marginheight="0">
<ASP:DataGrid id="MyGrid" runat="server"
    BorderStyle="None"
    CellSpacing="1"
    BorderWidth="0px"
    BackColor="#999999"
    CellPadding="2">
    <ItemStyle ForeColor="#000000" BackColor="#FFFFFF"
        Font-Name="Verdana" Font-Size="11px"/>
    <HeaderStyle Font-Bold="True" ForeColor="#000000"
        Font-Name="Verdana" Font-Size="12px" BackColor="#EEEEEE" />
</ASP:DataGrid>
</body>
</html>
```

Ce qui exécuté donne:



Posez aussi cette page web dans le dossier c:\TimeOff

Maintenant revenons à notre formulaire et rendons-le fonctionnel.

Accédez aux propriétés de la liste déroulante devant *Your name* et renommez le champ *EmpMail*. Cochez aussi *Cannot be blank*. Activez *Lookup in a database, Web service of file* et ensuite cliquez sur le bouton *Secondary Data Source*.

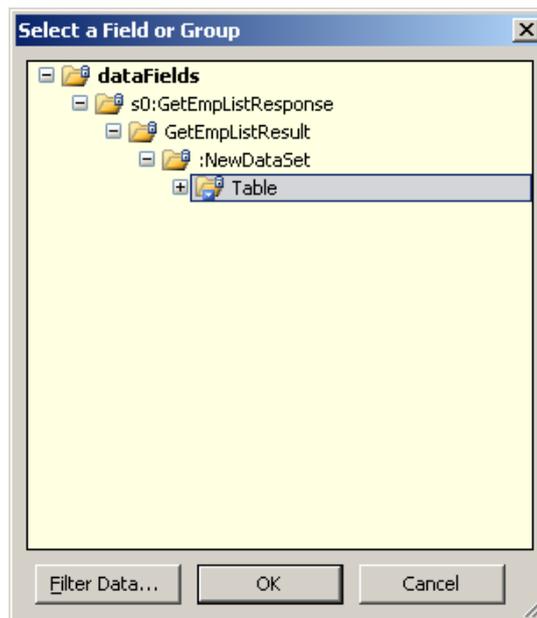
Vincent ISOZ

Prenez comme source de données secondaire le type *Web service* et cliquez ensuite sur *Next* et comme adresse du web service saisissez:

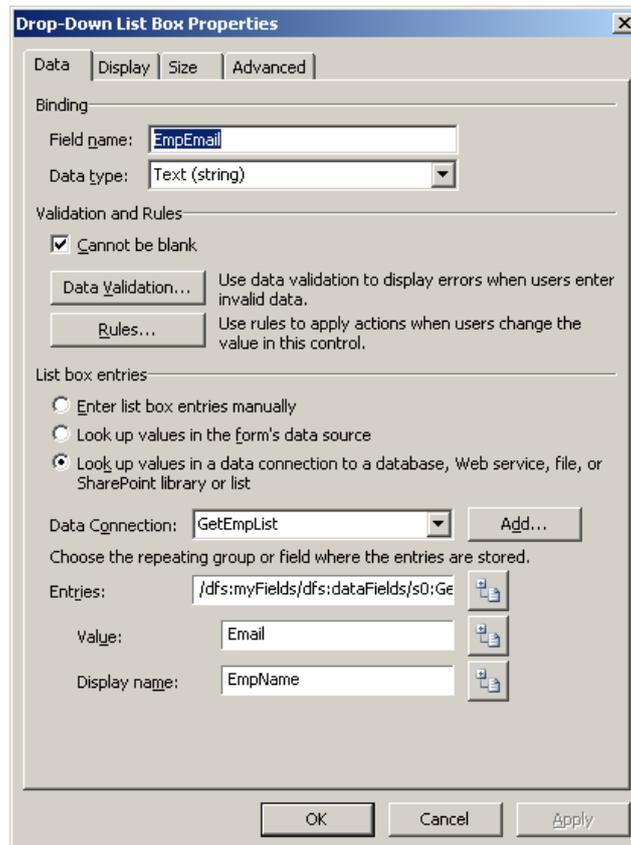
<http://localhost/TimeOff/DataService.asmx?WSDL>

sélectionnez la méthode *GetEmpList* et cliquez sur *Next* jusqu'à arriver à la fin de l'assistant.

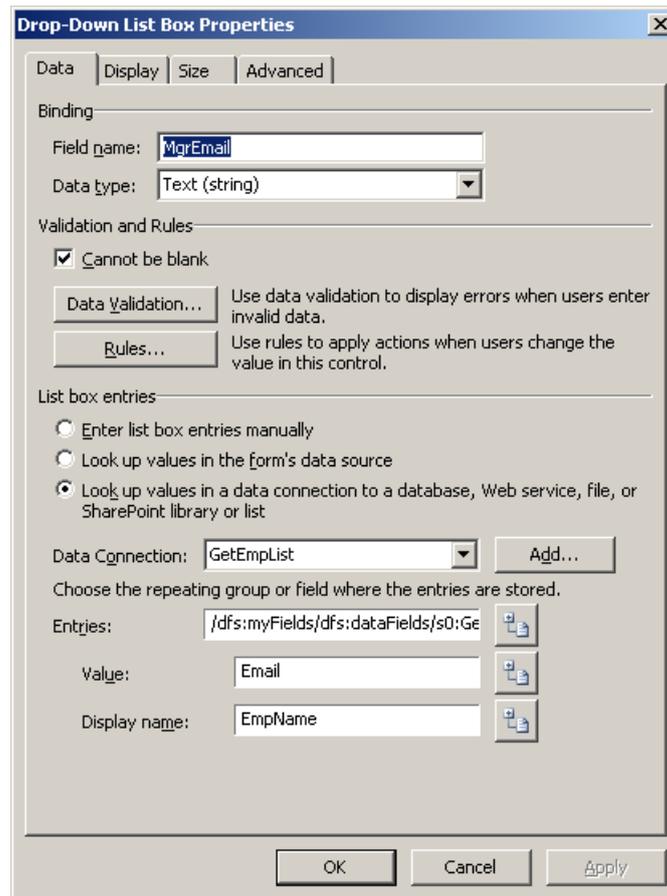
La source de données étant maintenant créée, nous avons besoin de lier la combo box avec la bonne expression XPath. Donc cliquez sur le bouton d'expressin *XPath* et sélectionnez:



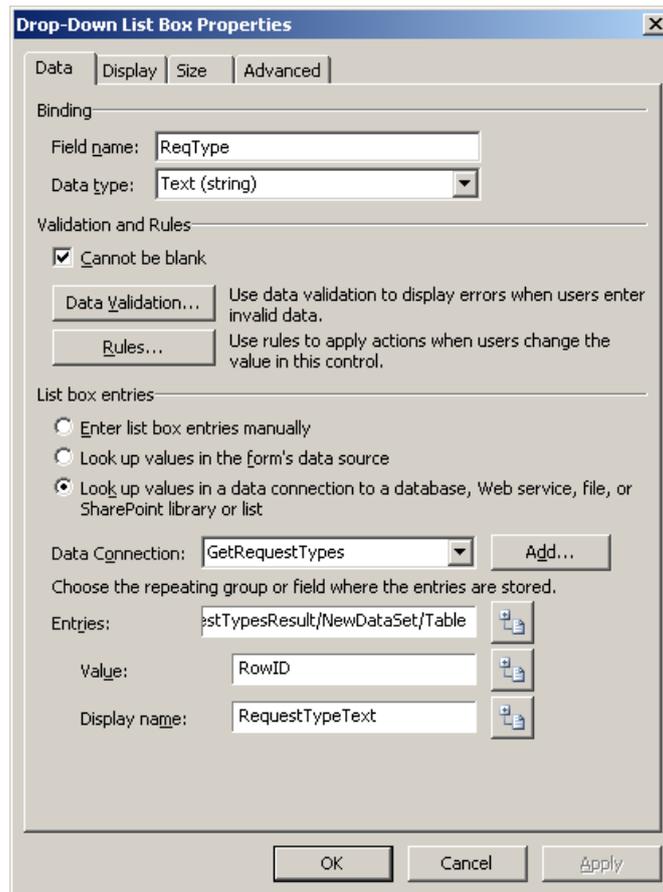
Ensuite faites en sorte d'avoir:



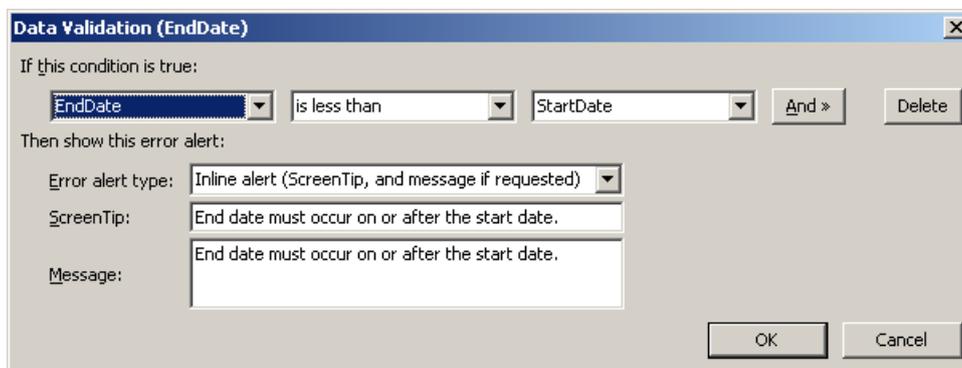
Procédez de même pour la liste *Manager's name* mais en utilisant la même source de données que précédemment:



Pour ce qui est de la combo box *Type* faites de même mais en créant un nouvelle source de données utilisant la méthode ad hoc du webservice:

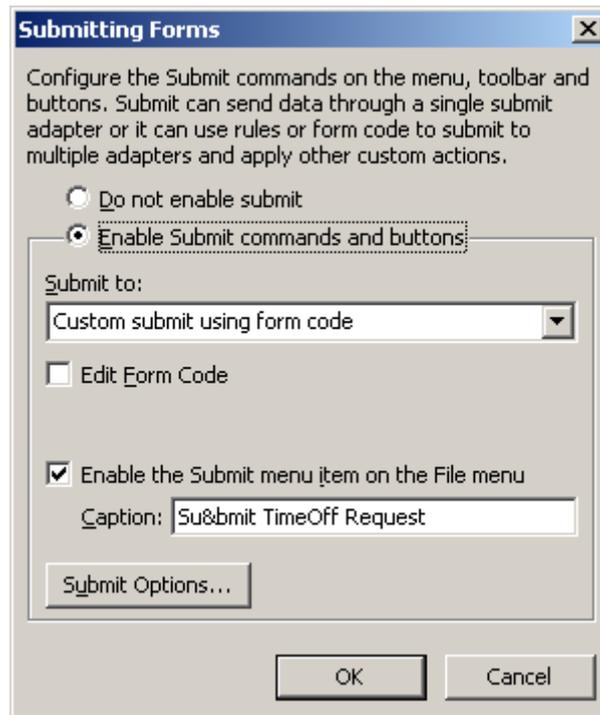


Nommez les deux *Date Picker* respectivement *StartDate* et *EndDate* tout en cochant *Cannot be blank*. Mettez ensuite une validation (*Data Validation*) sur le champ *EndDate* du type:



Ensuite, nommez le champ des heures *Hours* et sélectionnez le type de données comme *Whole Number (integer)* et cochez *Cannot be blank*.

Enfin, accédez aux propriétés du bouton de soumission et sélectionnez l'action *Submit*. Dans la boîte de dialogue qui apparaît alors:



sélectionnez *Enable Submit commands and buttons* et dans la liste déroulante *Submit to* sélectionnez *Custom submit using form code* et cochez *Edit Form Code*.

En cliquant sur *Submit Options* saisissez:



**Remarque:** Il est important de nommer correctement les champs car le webservice que nous avons développé utilise des noms bien spécifiques!!!

Ensuite, si vous validez deux fois par *OK* l'éditeur de script s'ouvre. Saisissez le code suivant sur l'événement de soumission:

## Vincent ISOZ

```
function XDocument::OnSubmitRequest(eventObj)
{
    // If the submit operation is successful, set
    // eventObj.ReturnStatus = true
    // Write your code here
    try
    {
        var objXH = new ActiveXObject("MSXML2.XMLHTTP.3.0");
    }
    catch(exception)
    {
        XDocument.UI.Alert("Could not create MSXML2.XMLHTTP.3.0 object.\r\n" +
            exception.number + " - " + exception.description);
        eventObj.ReturnStatus = false;
        return;
    }

    try
    {
        objXH.open("POST", "http://localhost/TimeOff/DataService.asmx/InsertRequest", false);
        objXH.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        strPostXMLText = XDocument.DOM.xml;

        strPostXMLText = strPostXMLText.replace(/\+/g, "%2B");
        strPostXMLText = strPostXMLText.replace(/\&/g, "%26");

        //XDocument.UI.Alert(strPostXMLText);

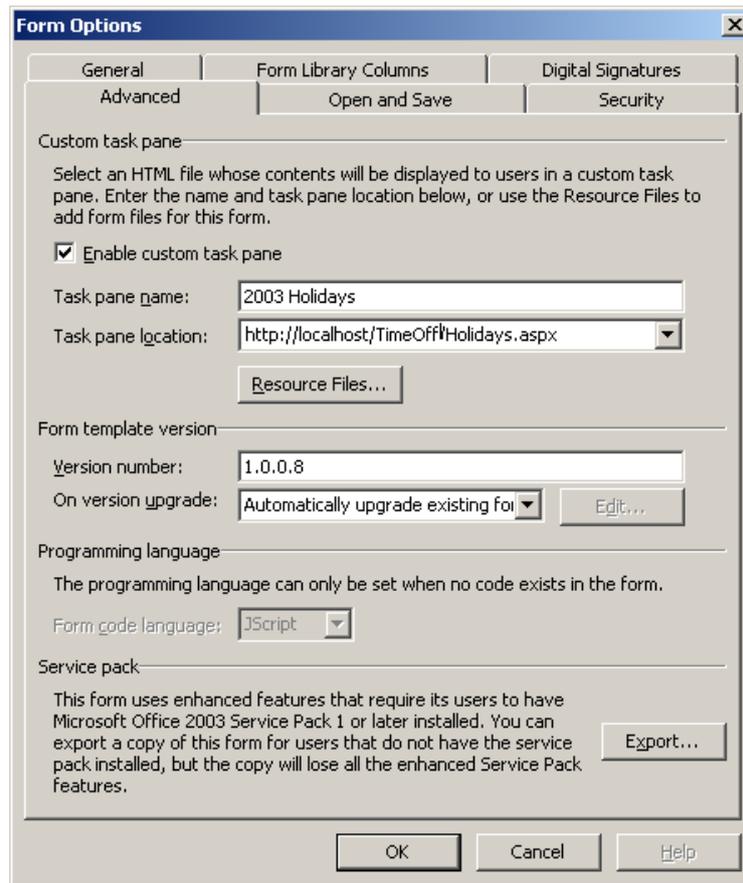
        objXH.send("TimeOffDetailsXML=" + strPostXMLText);

        if(objXH.status != 200)
        {
            XDocument.UI.Alert("Failed while sending the request.\r\n" + objXH.status + " - " +
                objXH.statusText);

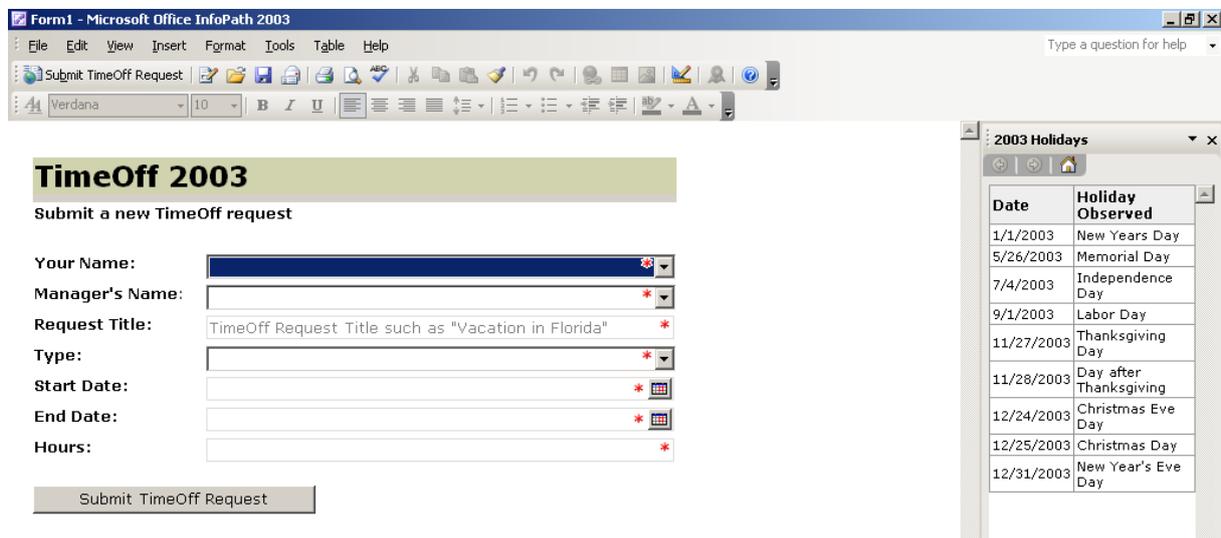
            eventObj.ReturnStatus = false;
            return;
        }
    }
    catch(exception)
    {
        XDocument.UI.Alert("Failed while sending the request.\r\n" + exception.number + " - " +
            exception.description);
        eventObj.ReturnStatus = false;
        return;
    }

    eventObj.ReturnStatus = true;
    return;
}
```

Il reste ensuite à demander à InfoPath d'afficher la page web *Holiday.aspx* dans le volet office du formulaire. Pour ce faire, il suffit d'aller dans *Tools/Form Options* et d'y saisir:



et finalement en sauvegardant le formulaire et en l'ouvrant pour la saisie, nous obtenons:

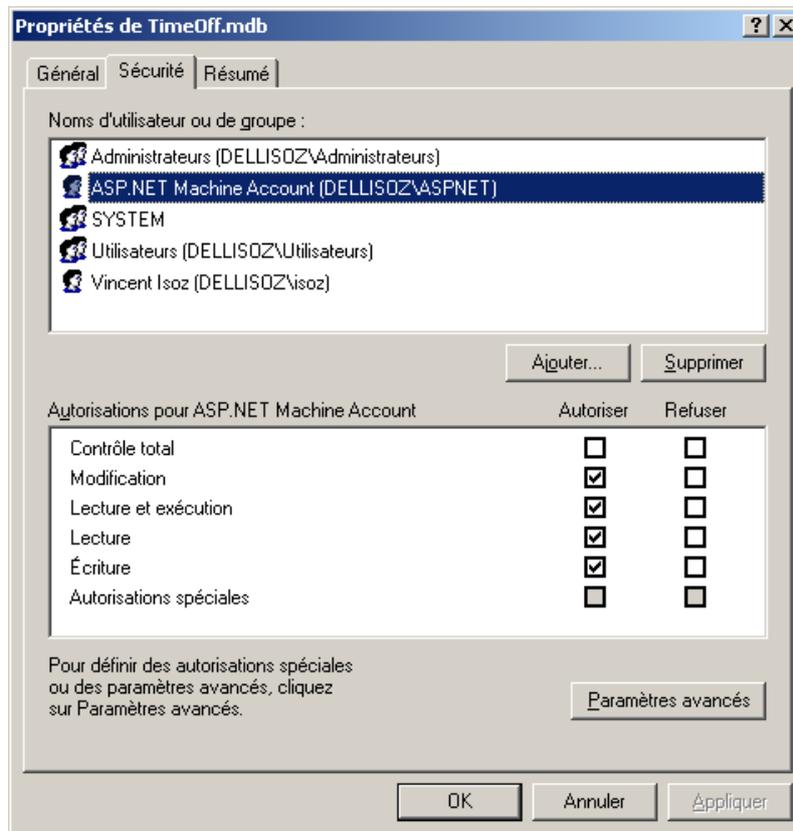


Si on remplit le formulaire et que plutôt que de l'envoyer on enregistre le fichier XML on peut obtenir un fichier XML du genre:

```

data.xml - Bloc-notes
Fichier Edition Format Affichage ?
<?xml version="1.0" encoding="UTF-8"?><?mso-infoPathSol
  <my:EmpEmail>margaretp</my:EmpEmail>
  <my:MgrEmail>margaretp</my:MgrEmail>
  <my:ReqType>4</my:ReqType>
  <my:ReqTitle>hé hé</my:ReqTitle>
  <my:StartDate>2006-04-24</my:StartDate>
  <my:EndDate>2006-04-25</my:EndDate>
  <my:Hours>4</my:Hours>
</my:myFields>
  
```

Si la soumission donnée ne fonctionne pas il faut faire un clic droit sur le fichier \*.mdb et aller dans *Propriétés* et ajouter l'utilisation *ASPNET* avec les droits en écriture et modification:



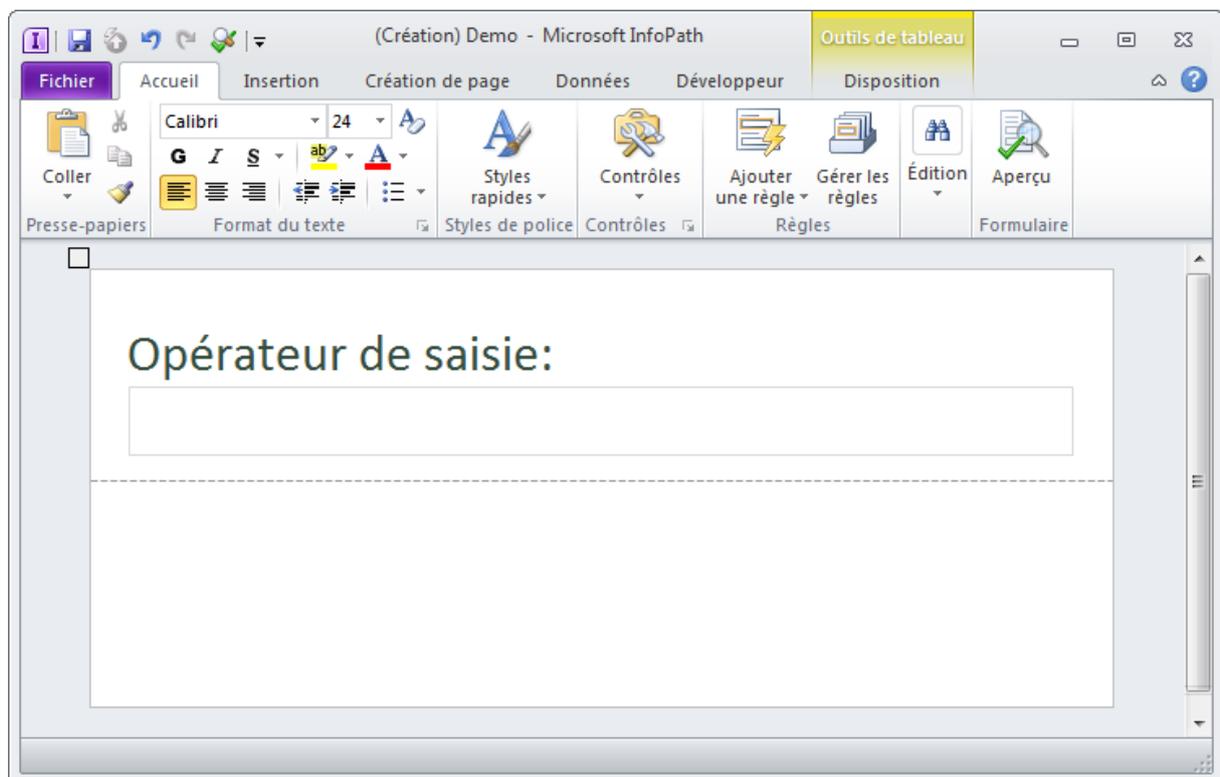
## 11. Applications pratiques (pour SharePoint)

### 11.1 Récupérer le nom de l'opérateur de saisie du formulaire à la prochaine ouverture du formulaire (v2010)

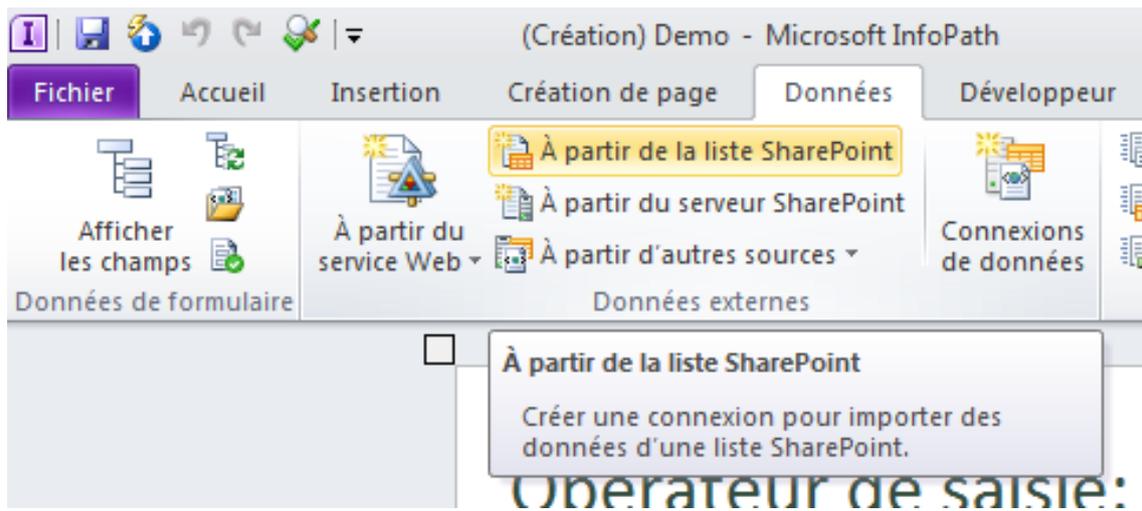
Si vous souhaitez récupérer le nom de l'opérateur de saisie lors de la prochaine ouverture du formulaire par un approbateur, vous pouvez tout à fait faire cela en vous connectant à la source SharePoint du formulaire et en rapatriant l'information dans un champ de votre choix qui sera typiquement en lecture seule et sans bordures.

Il existe plusieurs techniques et certains ont des problèmes. Donc voici les détails de la partie technique (les options de formatage étant supposées déjà connues) de celle qui selon moi fonctionne le mieux.

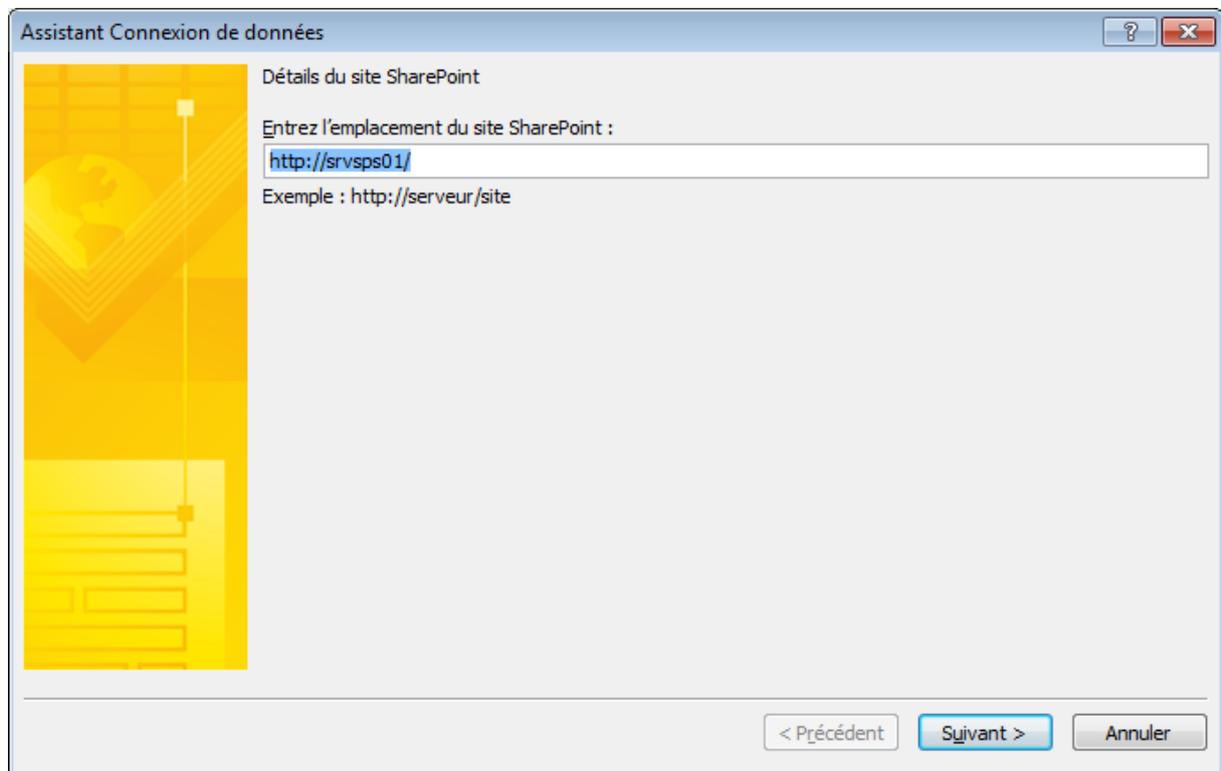
D'abord, il faut préparer le champ qui réceptionner le nom de l'opérateur de saisie provenant de SharePoint (nous l'avons nommé *OperateurSaisie*):



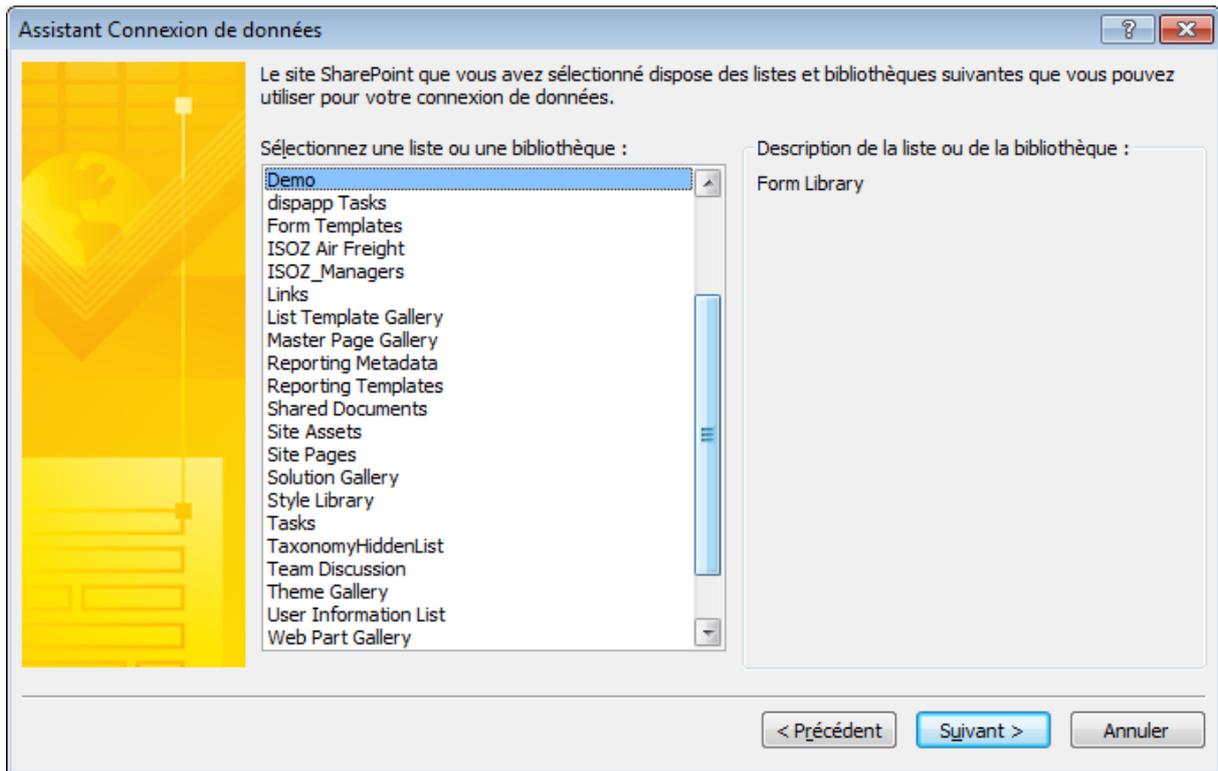
Il faut ensuite publier le formulaire (c'est classique voir plus haut). Une fois que c'est fait il faudra créer une connexion à la bibliothèque de ce formulaire:



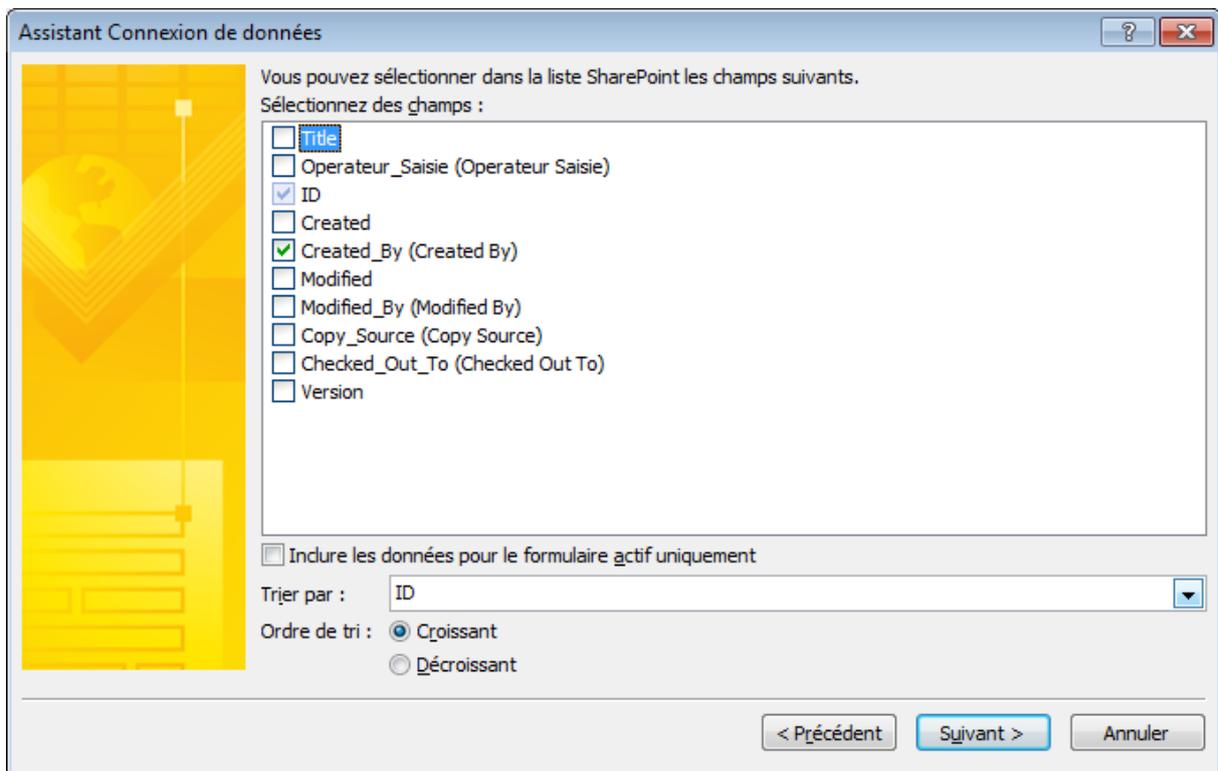
Nous mettons l'adresse du site SharePoint contenant votre bibliothèque de formulaire:



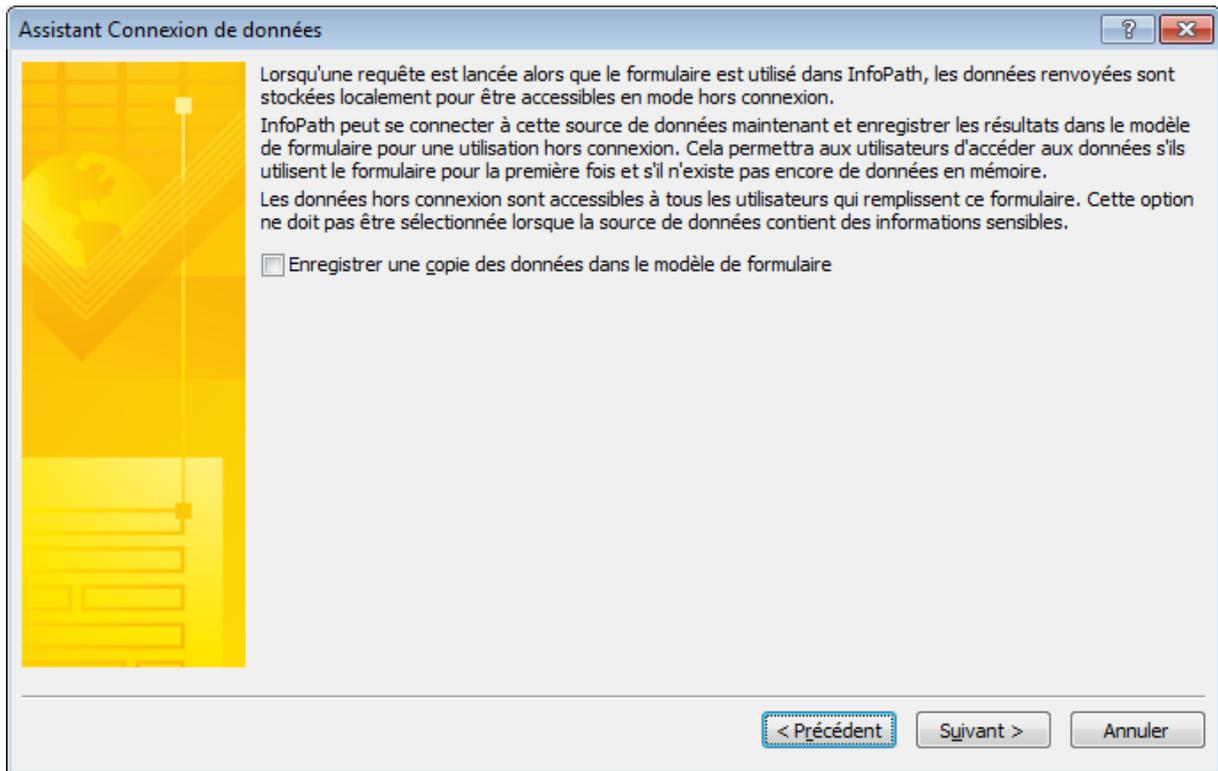
Ensuite, nous sélectionnons le nom de notre bibliothèque de formulaire (*Demo* dans le cas présent):



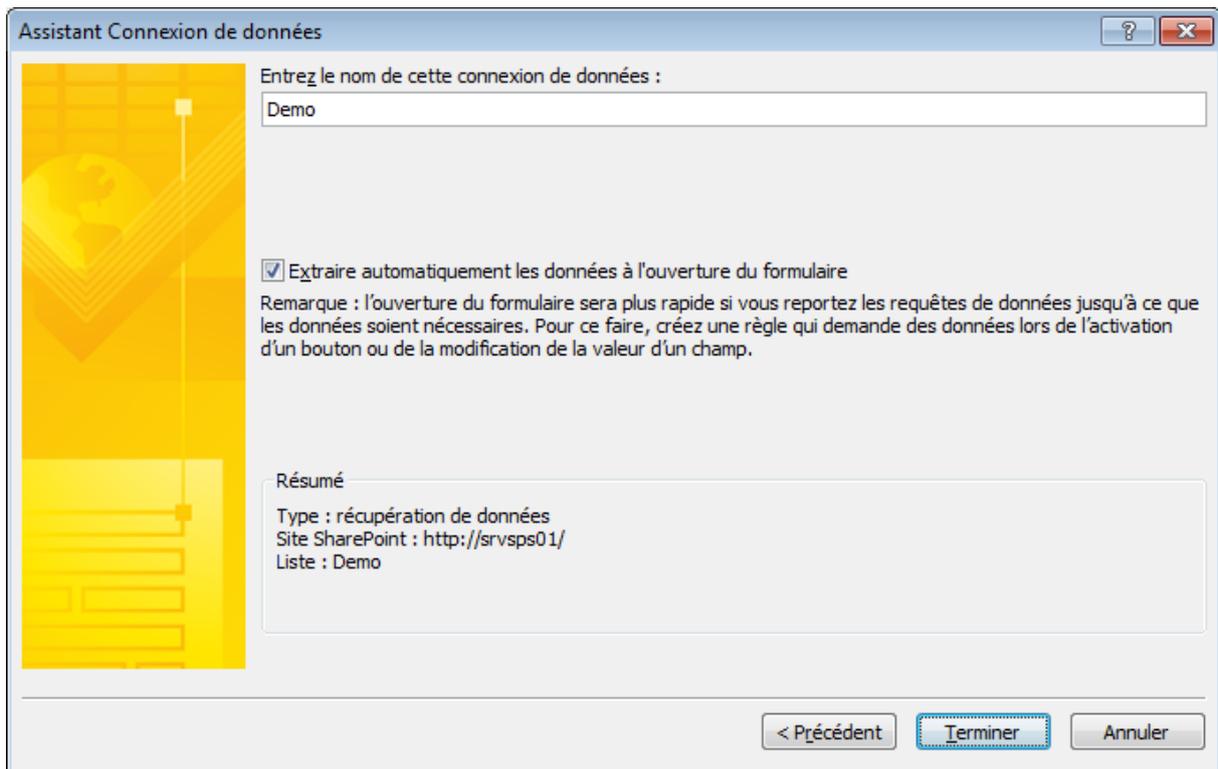
Nous rappatrons au moins la colonne *Created\_By* de SharePoint puisque c'est cette dernière qui stocke le nom de l'utilisateur ayant fait la saisie:



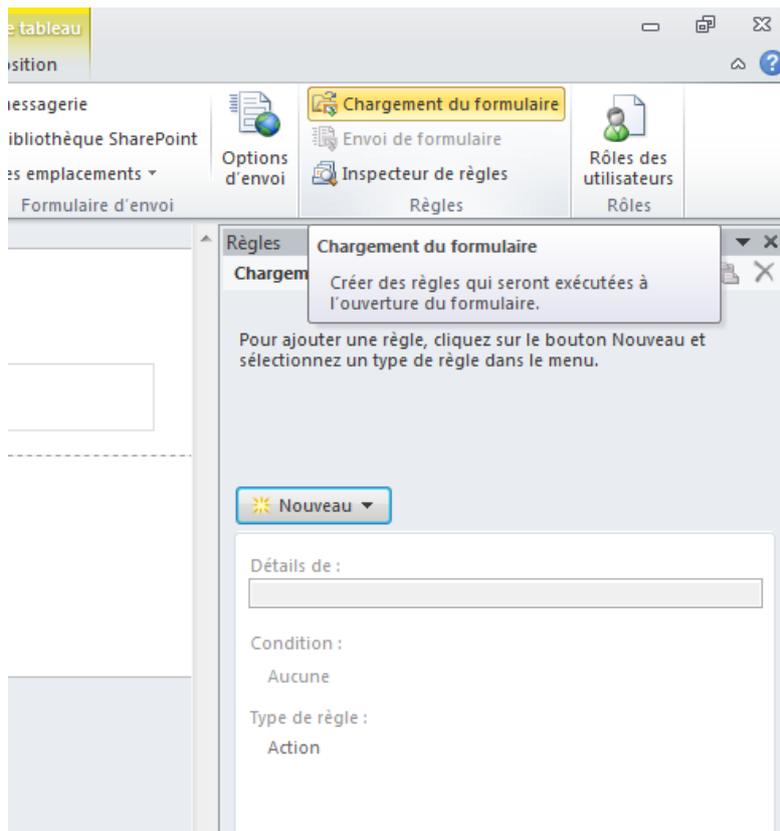
Nous cliquons sur *Suivant*:



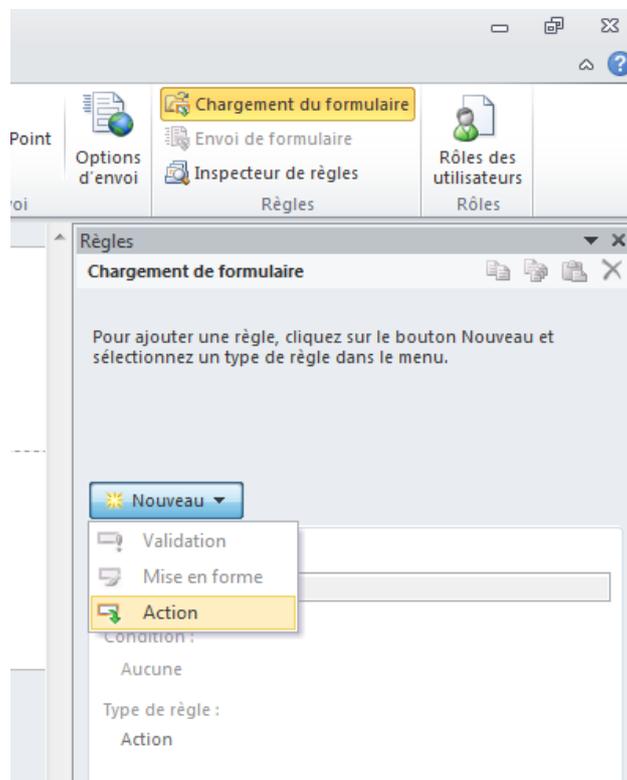
et encore *Suivant*:



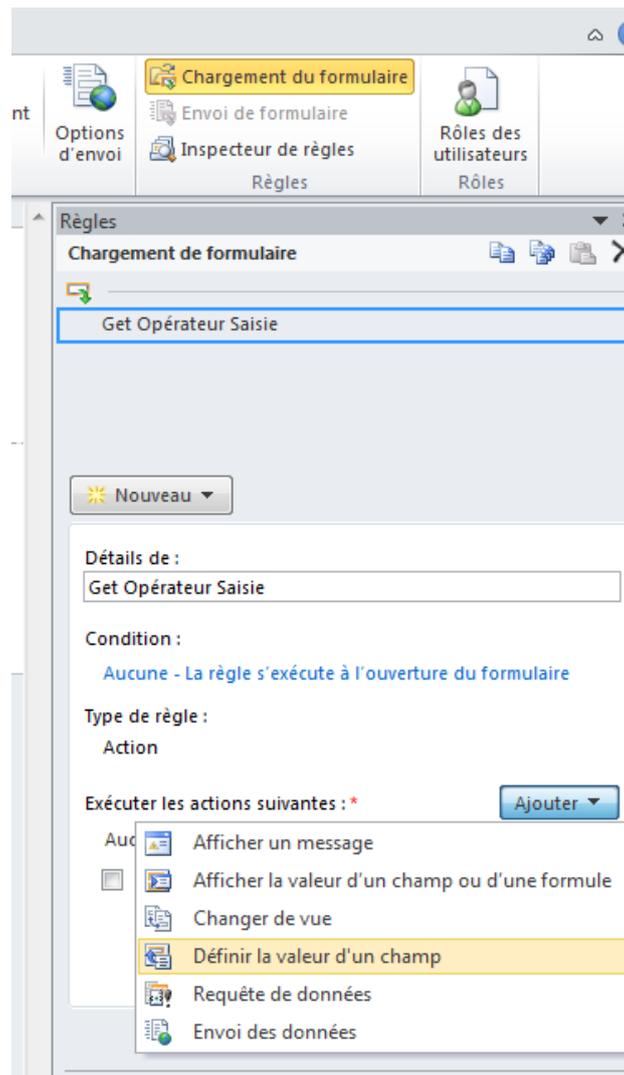
Nous validons par *Terminer*. Ensuite, dans le ruban *Données*, nous cliquons sur le bouton *Chargement du formulaire*:



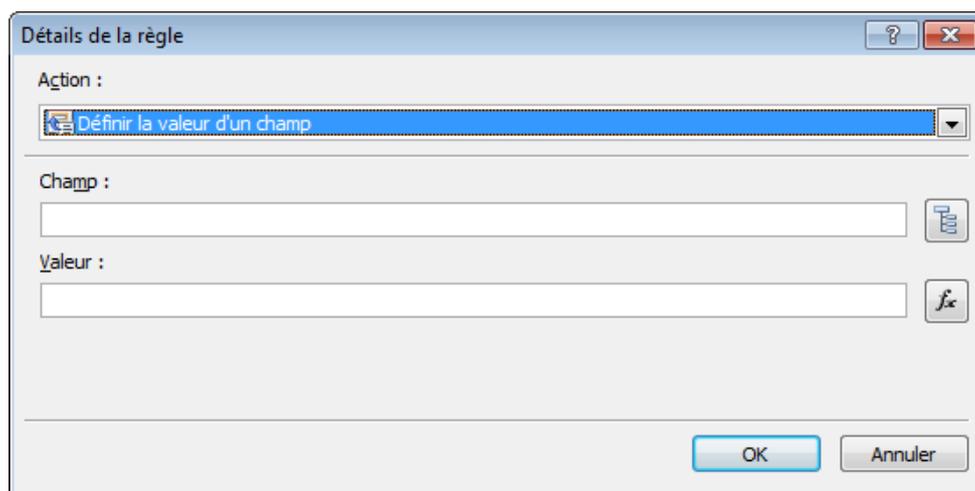
et nous cliquons sur le bouton *Nouveau* qui apparaît dans le panneau latéral:



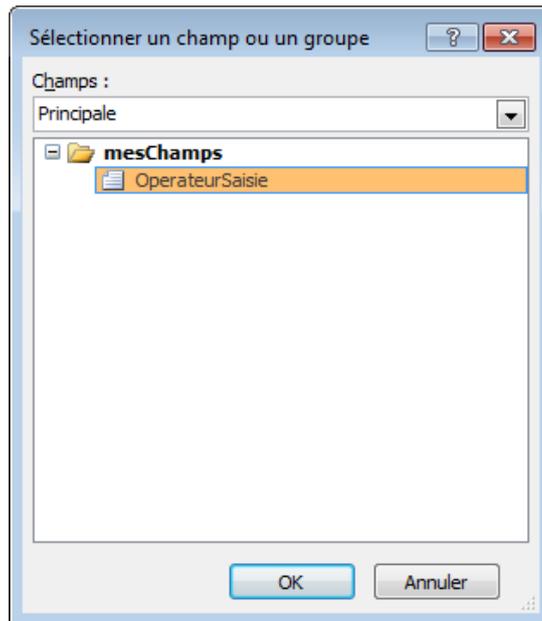
et nous créons une *Action*:



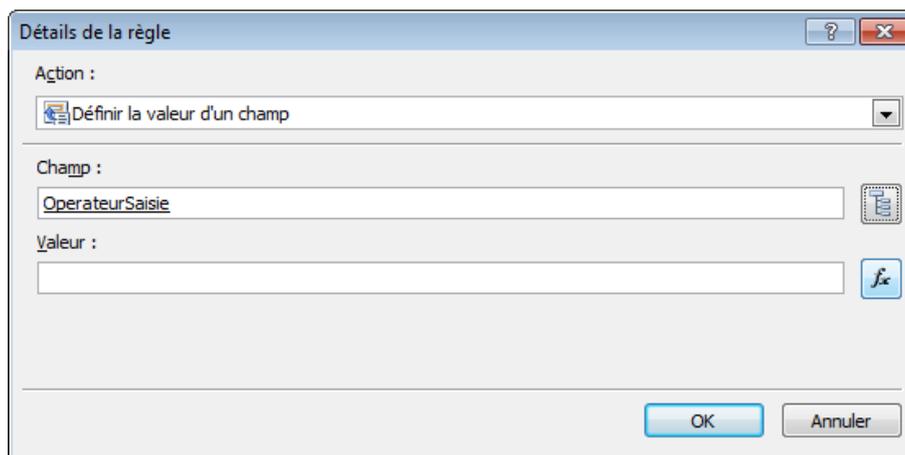
Après avoir donné un nom à l'action, dans le bouton *Ajouter* nous prenons l'option *Définir la valeur d'un champ* et la boîte de dialogue suivante s'ouvrira:



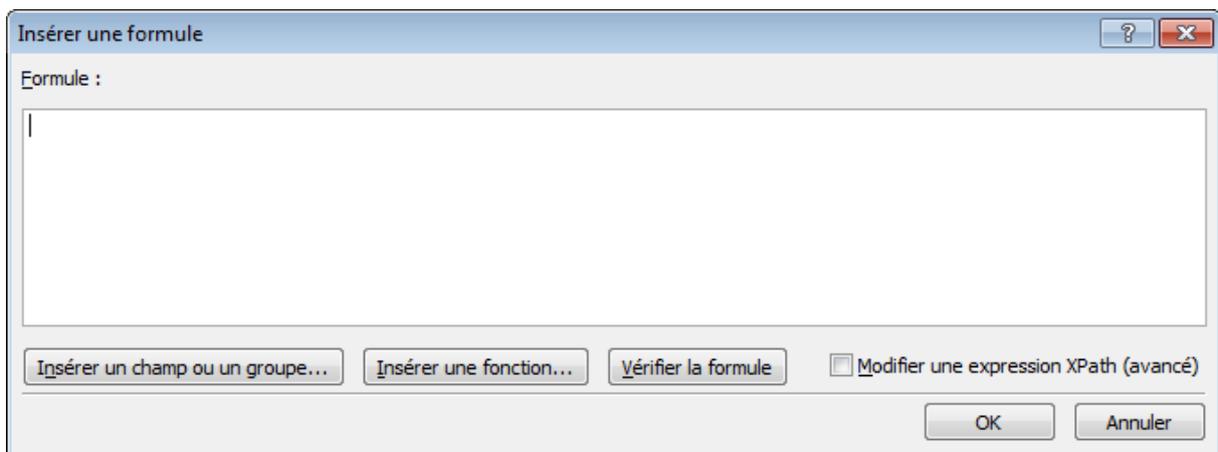
Dans *Champ* nous prenons celui qui se trouve dans notre formulaire et qui va donc réceptionner la valeur:



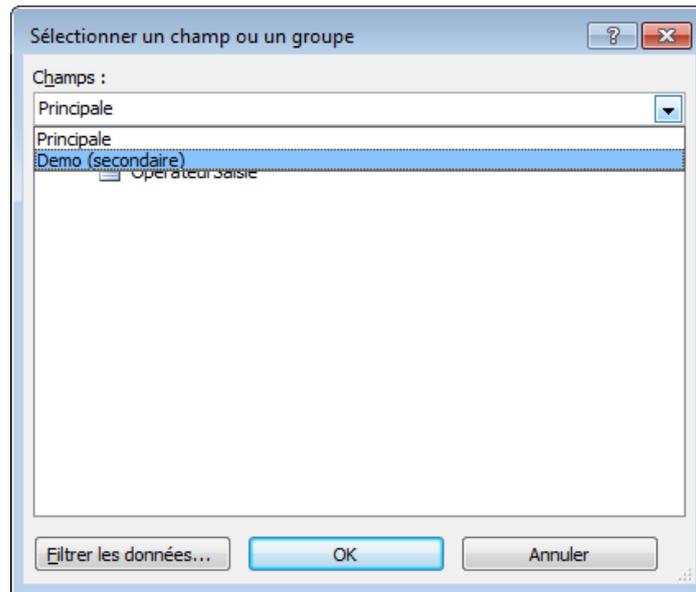
Nous validons par *OK*:



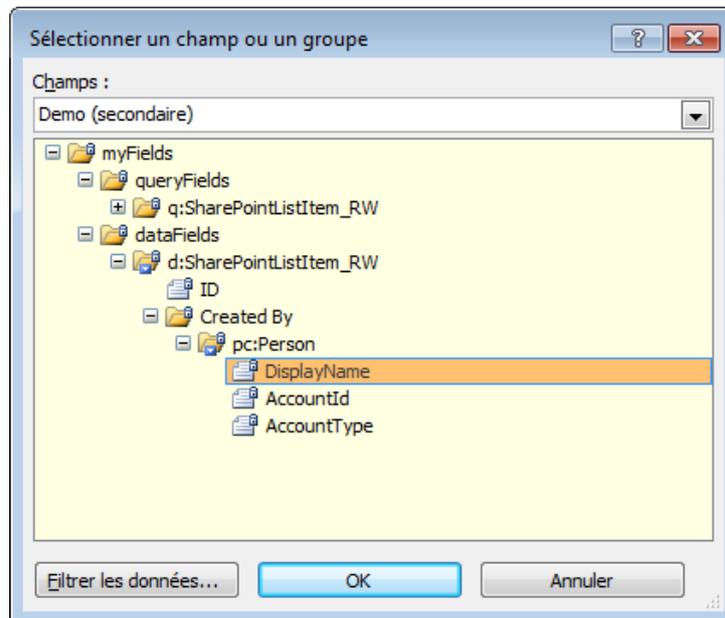
Nous cliquons sur le *fx* de *Valeur* et la boîte de dialogue suivante apparaît:



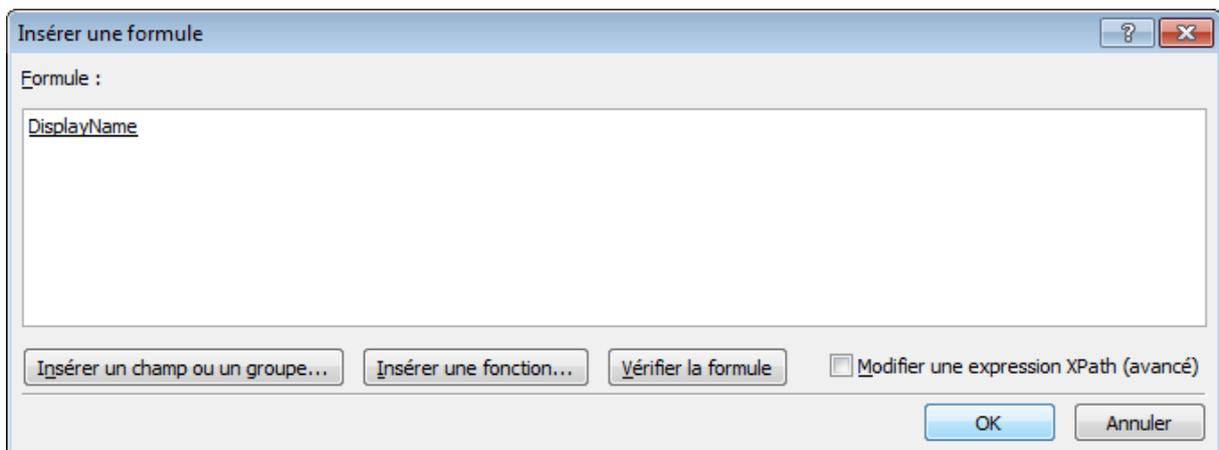
Nous cliquons sur *Insérer un champ ou un groupe...* et de suite nous sélectionnons la source secondaire provenant de SharePoint dans la boîte de dialogue qui apparaît:



et dans cette source nous allons prendre l'attribut *Display name* du noeud *Person*:

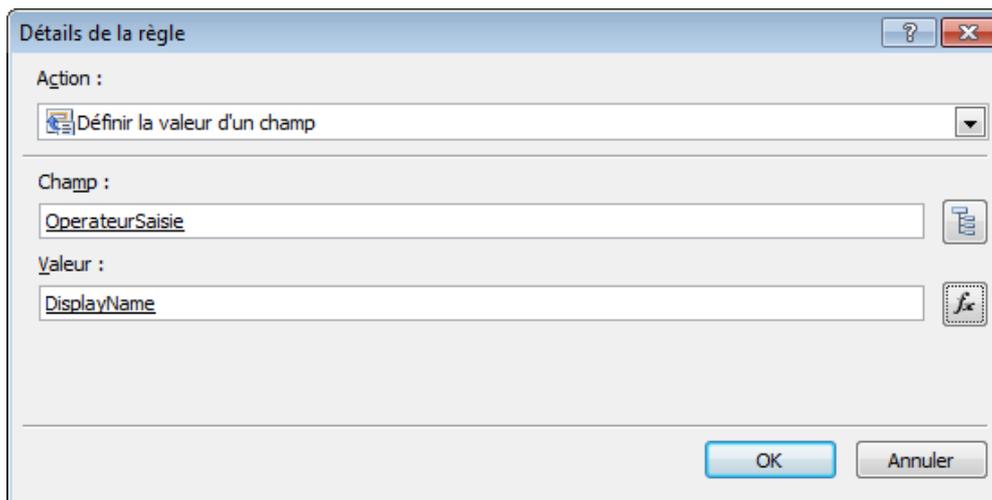


Nous validons par *OK* ce qui nous ramène dans:

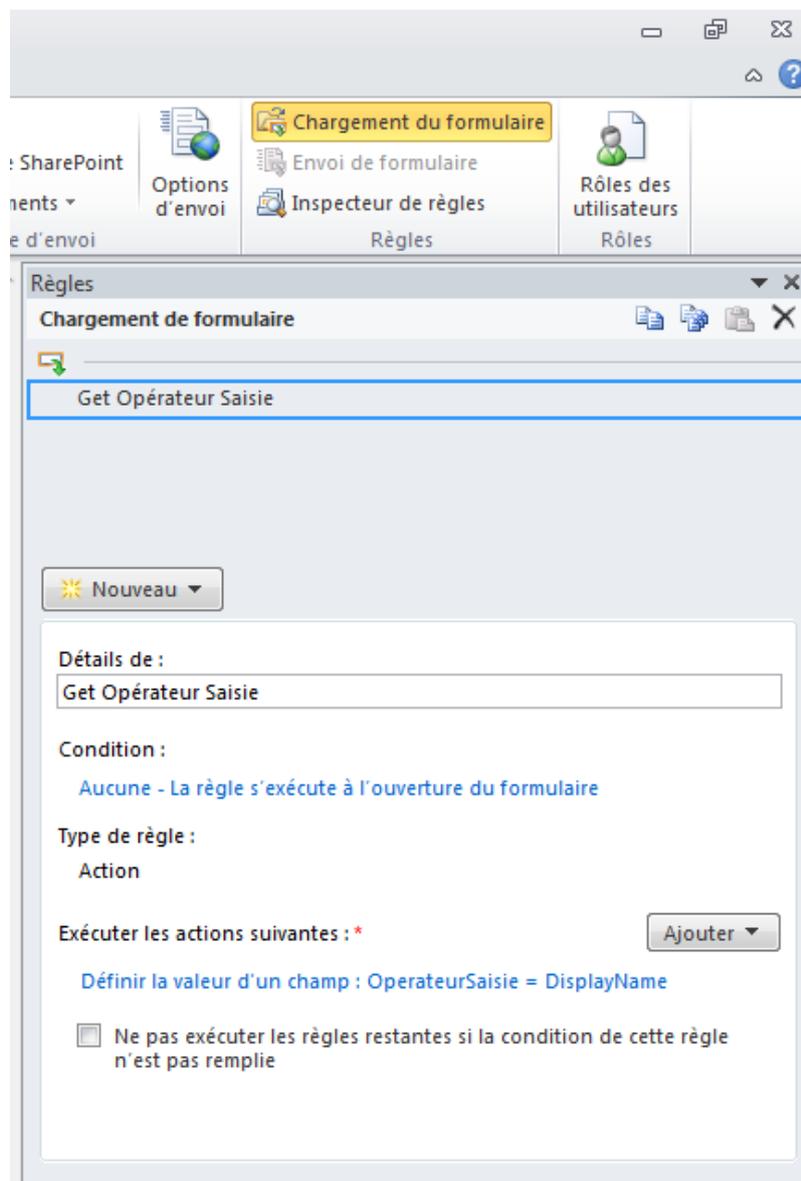


Vincent ISOZ

Nous revalidons par *OK*, ce qui nous ramène dans:



Nous revalidons par *OK* ce qui nous ramène enfin dans le panneau latéral:



Vincent ISOZ

et voilà c'est fini. Il n'y a plus qu'à publier la mise à jour de ce formulaire dans votre bibliothèque de formulaires SharePoint et vous verrez qu'après avoir créé une formulaire, une fois que celui-ci est rouvert le nom du créateur initial apparaîtra dans le champ *OperateurSaisie*.

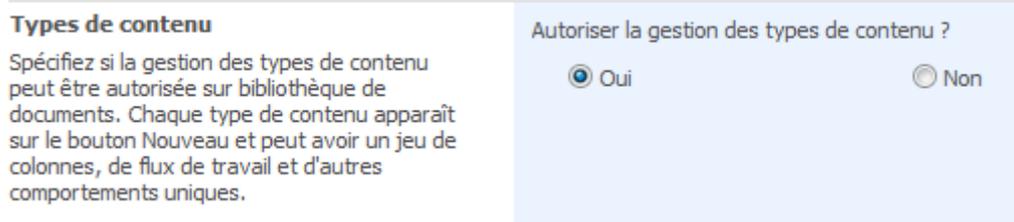
## 11.2 Personnalisation du Data Information Panel (v2007)

Pour des raisons esthétiques principalement, un certain nombre de responsables qualité apprécient de personnaliser le DIP de Word, Excel et PowerPoint. Nous allons voir comment procéder, mais il faudra se rappeler que:

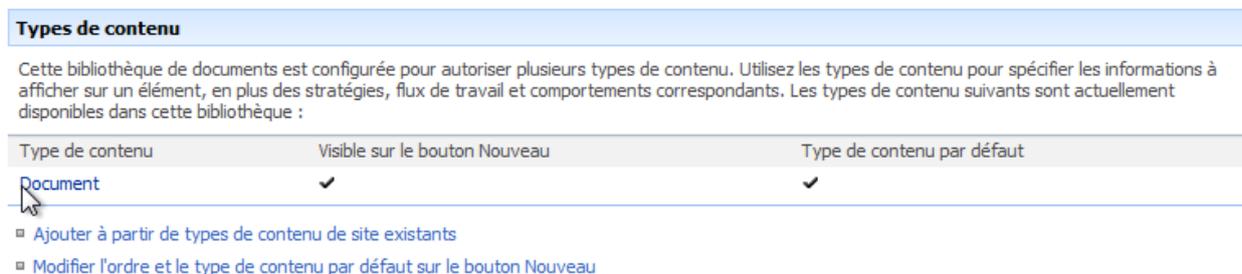
1. Cet outil ne fonctionne pas avec d'autres logiciels que Word, Excel et PowerPoint ce qui réduit considérablement son utilité.
2. Les personnes qui devront faire usage du DIP personnalisé, devront aussi avoir InfoPath installé sur leur ordinateur.
3. Tous les modèles doivent se trouver sur SharePoint.

Nous supposons pour l'exemple que le lecteur maîtrise déjà les bases de SharePoint: création de bibliothèques, définition de modèles, création de colonnes, etc. Par ailleurs, ces notions font partie des prérequis du cours InfoPath.

Donc dans la bibliothèque de votre choix, dans laquelle vous avez pris soin d'activer **l'Autorisation des types de contenu**:



vous cliquez sur le type de contenu dont vous souhaitez personnaliser le DIP:



et vous cliquez sur Paramètres du Panneau information sur le document:

**Informations sur les types de contenu de liste**

Nom : Document  
Description : Crée un document.  
Parent : Document

**Paramètres**

- Nom et description
- Paramètres avancés
- Paramètres du flux de travail
- Supprimer ce type de contenu
- Paramètres de la stratégie de gestion des informations
- Paramètres du panneau Informations sur le document

Viens alors la page suivante:

<p><b>Modèle du panneau Informations sur le document</b></p> <p>Spécifiez le type de modèle du panneau Informations sur le document à afficher dans les applications compatibles avec Microsoft Office.</p> <p>Remarque : la création d'un modèle personnalisé pour le panneau Informations sur le document requiert Microsoft Office InfoPath. <a href="#">Plus d'informations sur la création de modèles du panneau Informations sur le document</a></p>	<p><input checked="" type="radio"/> Utiliser le modèle par défaut pour les applications Microsoft Office</p> <p><input type="radio"/> Utiliser un modèle personnalisé existant (URL, UNC ou URN)</p> <p><input type="text"/></p> <p><a href="#">Modifier ce modèle</a></p> <p><input type="radio"/> Télécharger un modèle personnalisé existant (XSN) à utiliser</p> <p><input type="text"/> <input type="button" value="Parcourir..."/></p> <p><a href="#">Créer un nouveau modèle personnalisé</a></p>
<p><b>Toujours afficher</b></p> <p>Force l'affichage automatique du panneau Informations sur le document pour l'utilisateur dans certaines conditions.</p>	<p><input type="checkbox"/> Toujours afficher le panneau Informations sur le document à l'ouverture d'un document et enregistrer initialement ce type de contenu</p>

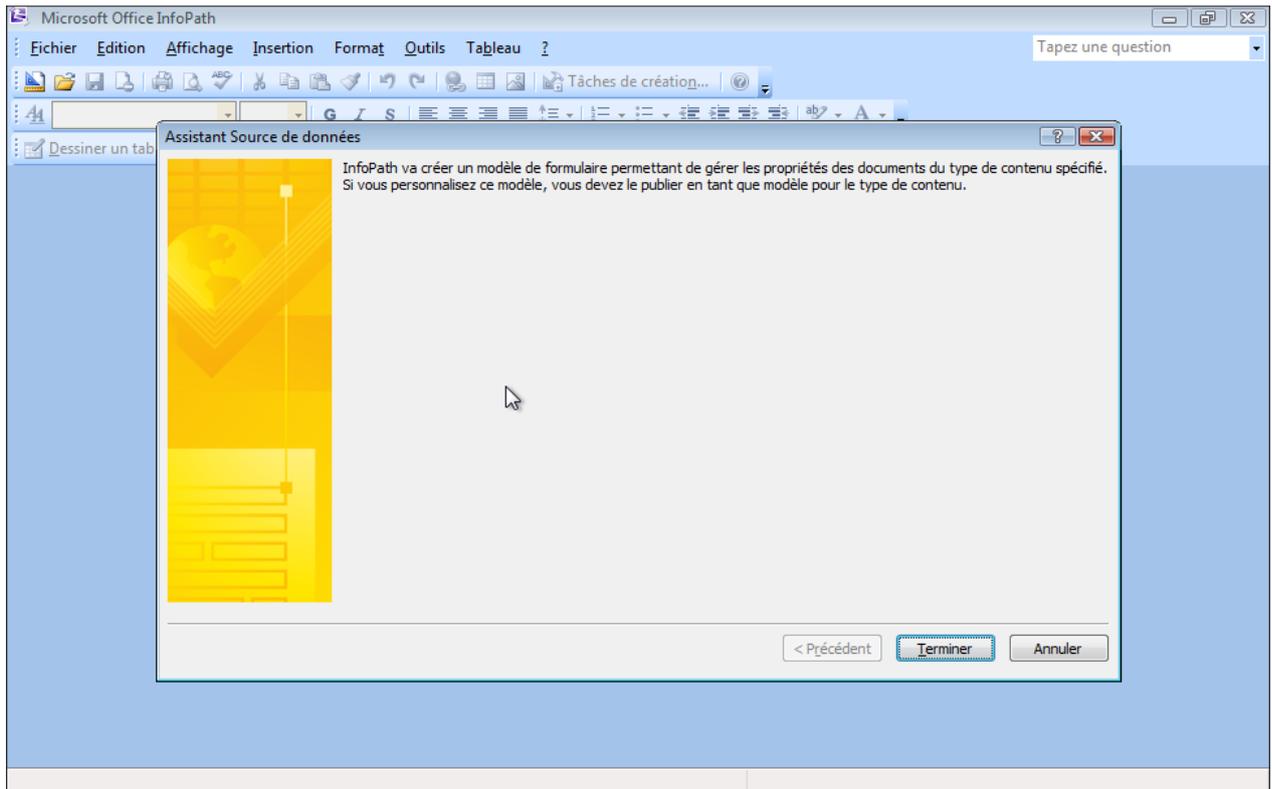
Nous cliquons sur **Télécharger un modèle personnalisé existant XSN à utiliser** et puis sur **Créer un nouveau modèle personnalisé** (c'est le choix le plus simple):

[Modifier ce modèle](#)

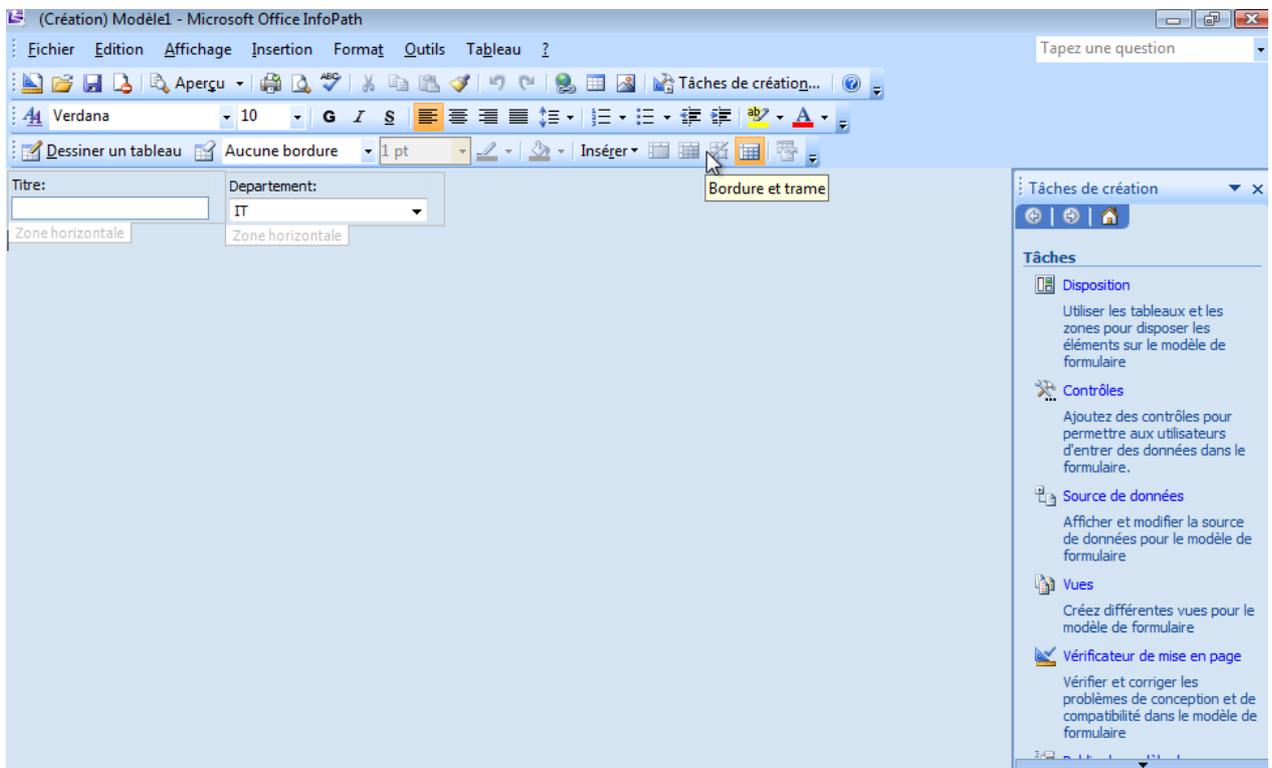
Télécharger un modèle personnalisé existant (XSN) à utiliser

[Créer un nouveau modèle personnalisé](#)

InfoPath s'ouvre:

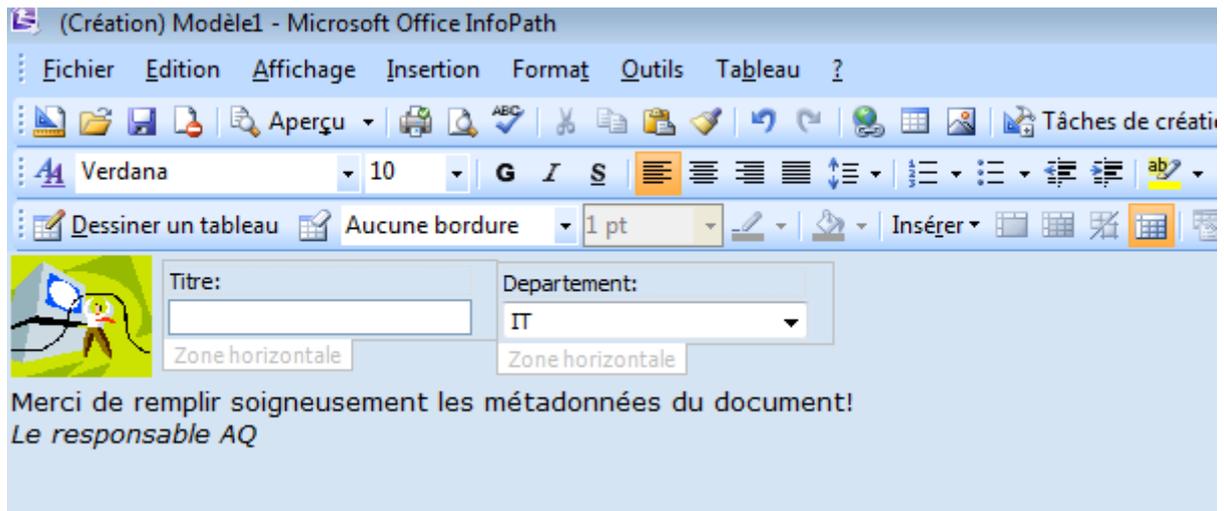


On clique sur Terminer et nous voyons déjà notre DIP avec toutes les colonnes que nous avons pris de créer avant de commencer la procédure:

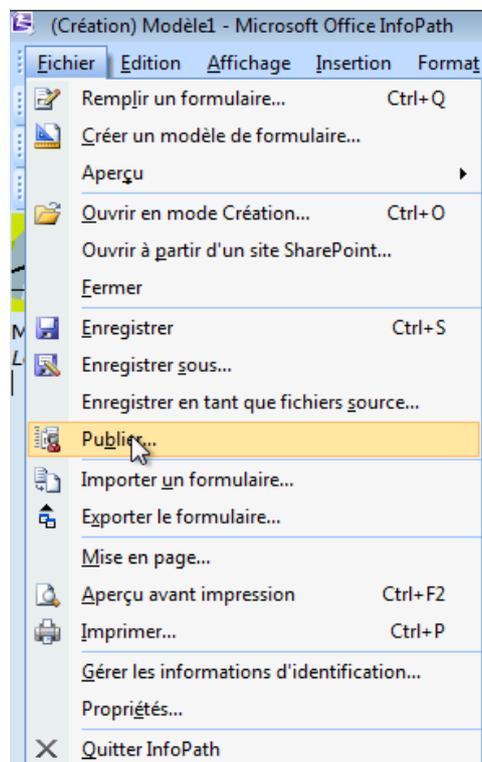


Pour faire simple, nous allons juste ajouter une image et un texte:

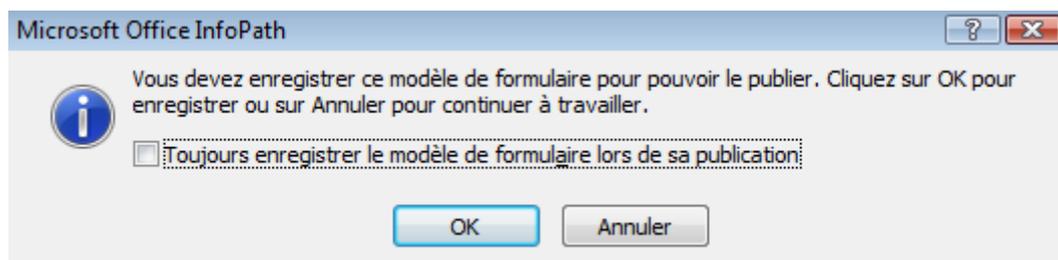
Vincent ISOZ



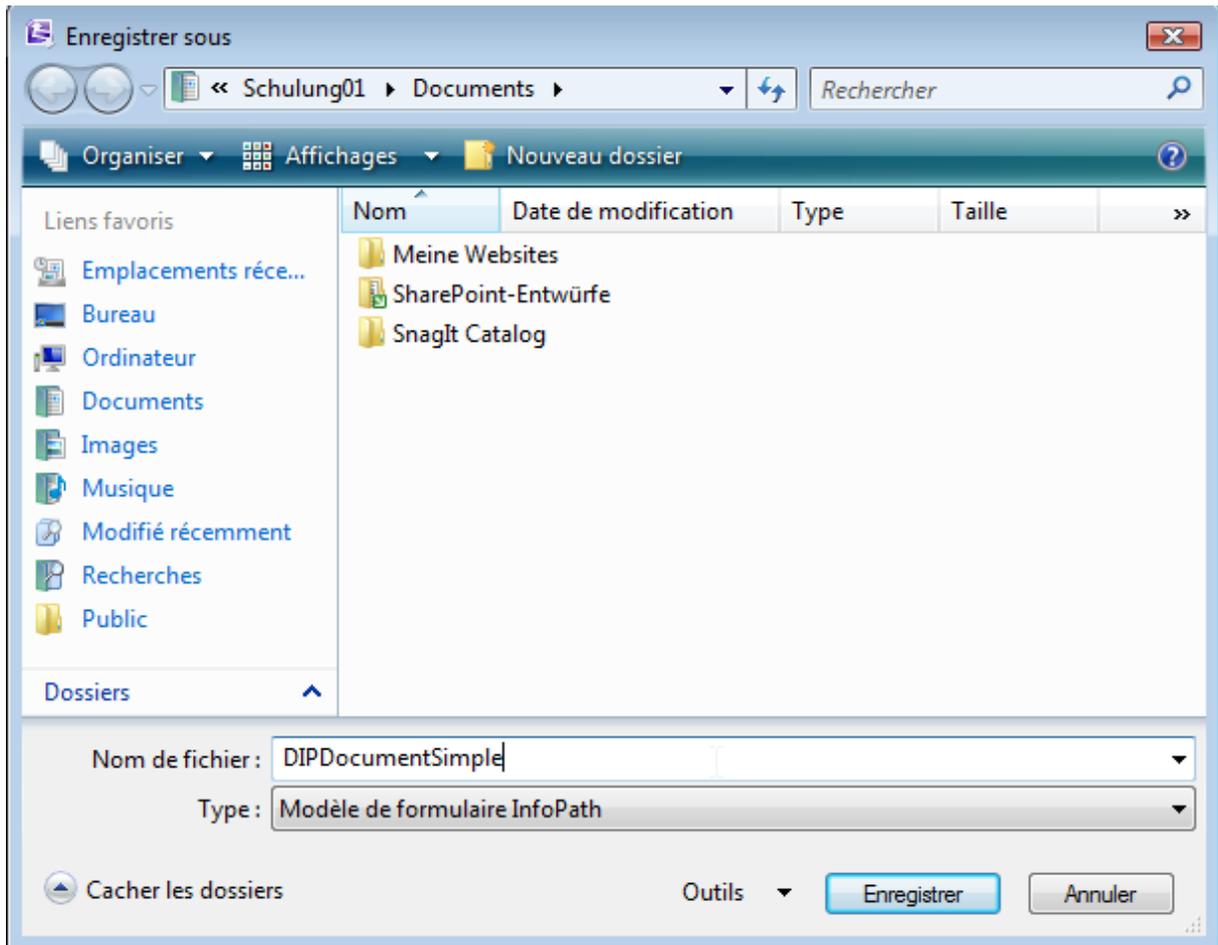
et pour clore, il faut le **Publier**:



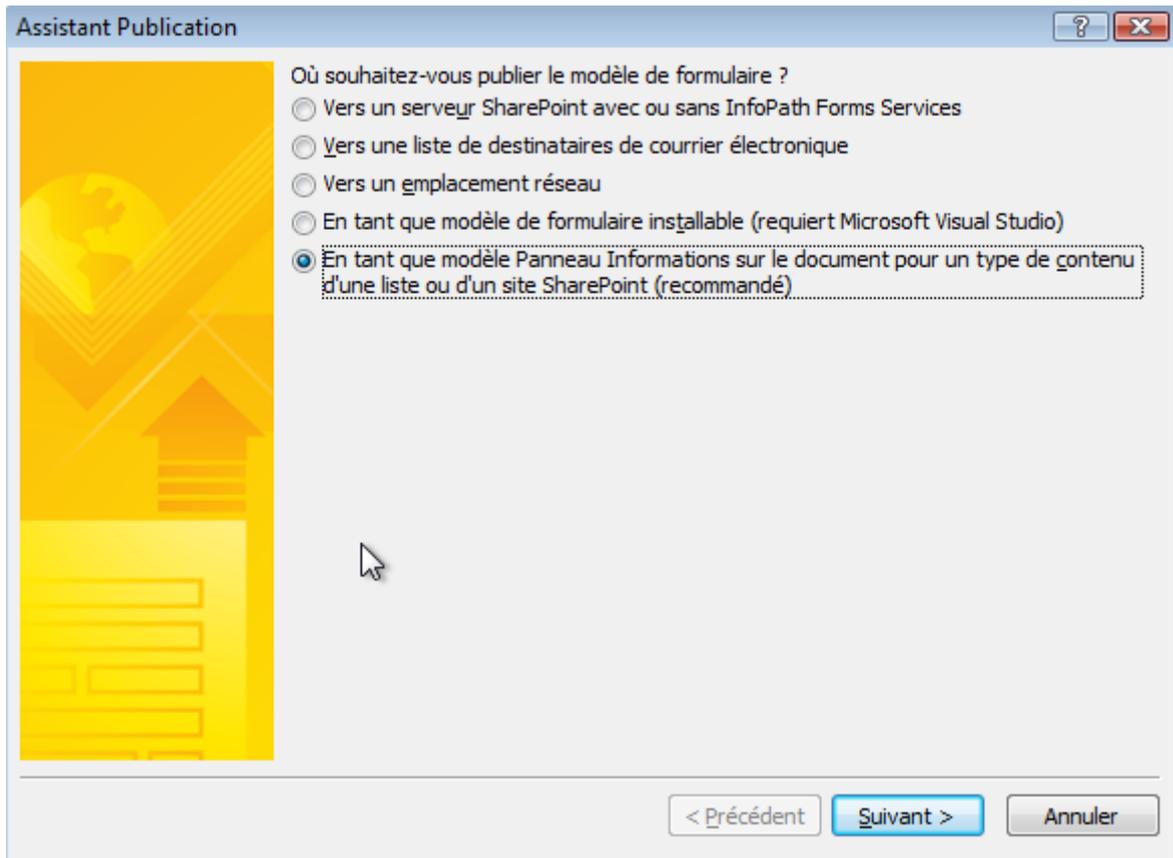
Il vient:



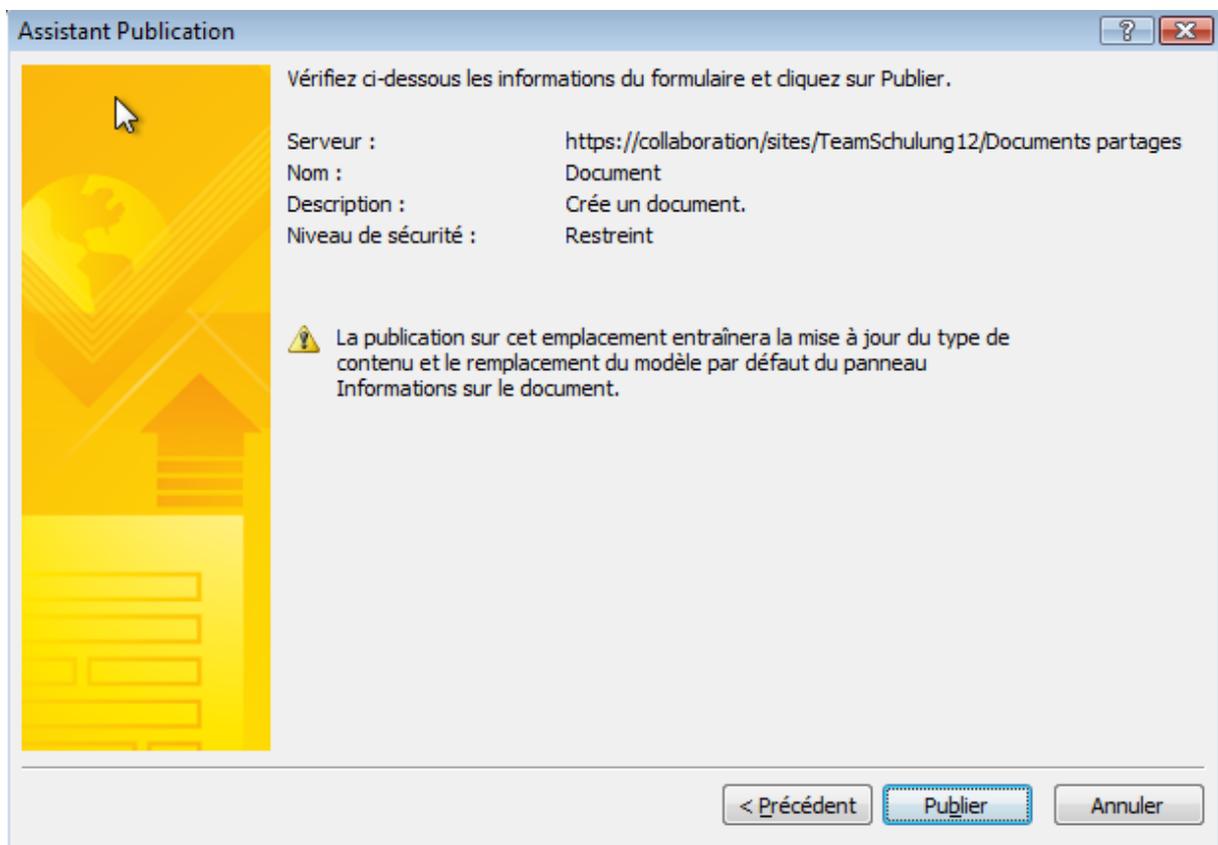
Nous validons par **OK**:



Nous validons par **Enregistrer**:

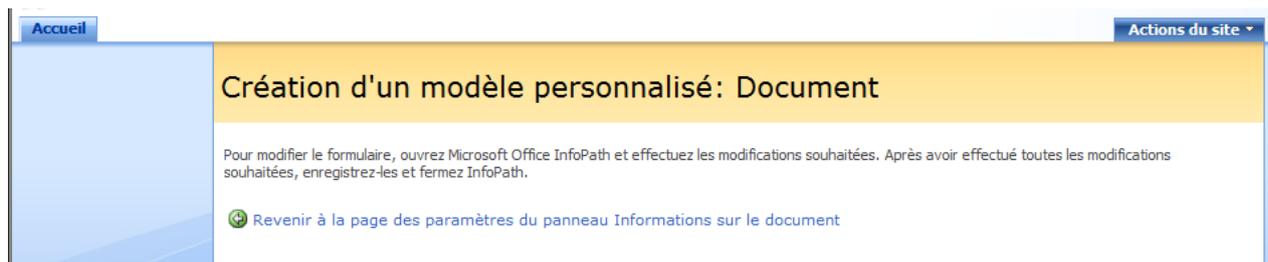


Automatiquement la bonne option apparaît. Nous cliquons sur **Suivant**:



Vincent ISOZ

et enfin sur **Publier**. Quand nous fermons InfoPath, nous revenons sur:

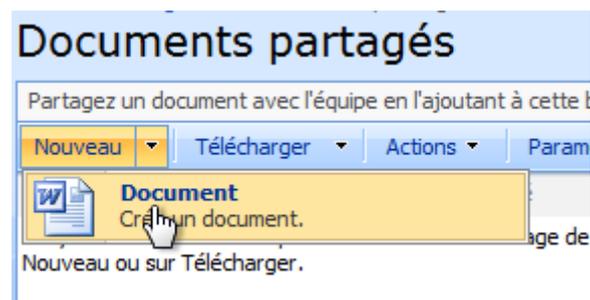


Nous cliquons sur **Revenir à la pages des paramètres du panneau Informations sur le document**:

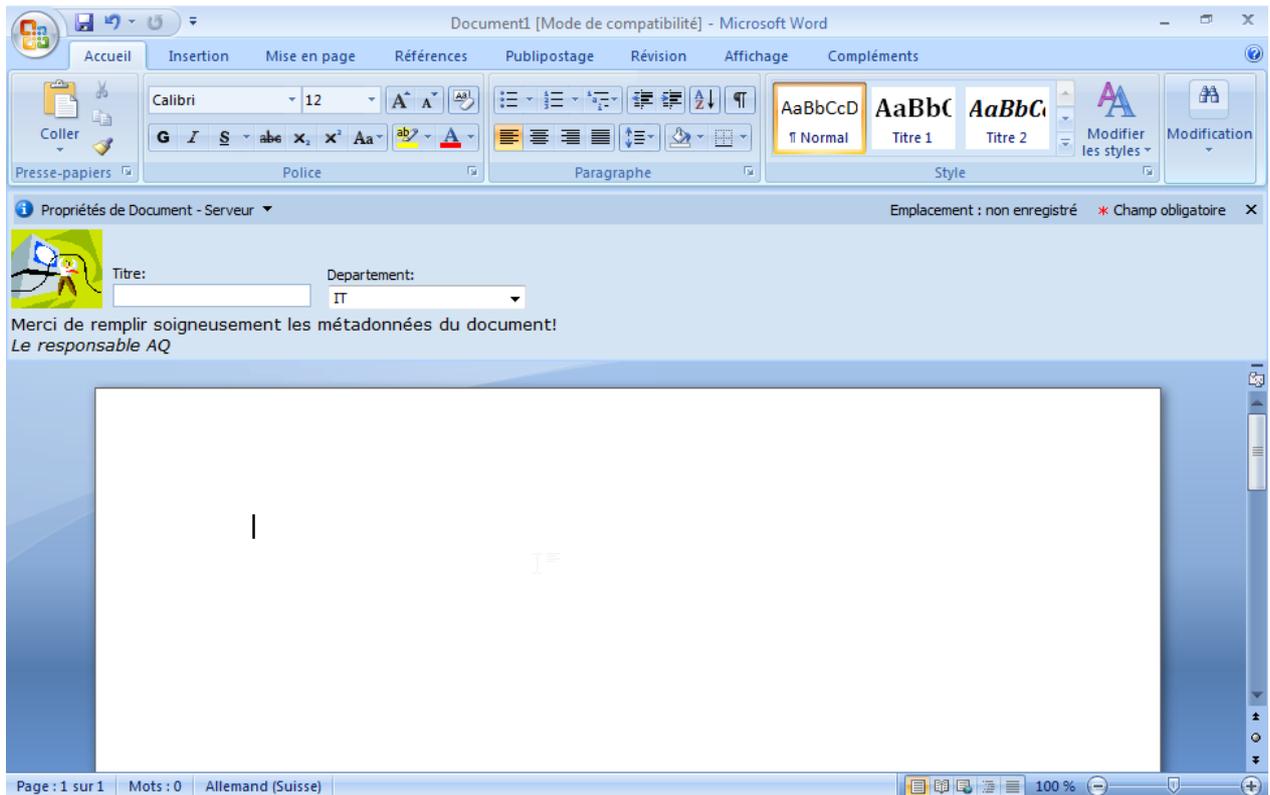
## Paramètres du panneau Informations sur le document: Document

<b>Modèle du panneau Informations sur le document</b> Spécifiez le type de modèle du panneau Informations sur le document à afficher dans les applications compatibles avec Microsoft Office.  Remarque : la création d'un modèle personnalisé pour le panneau Informations sur le document requiert Microsoft Office InfoPath. <a href="#">Plus d'informations sur la création de modèles du panneau Informations sur le document</a>	<input type="radio"/> Utiliser le modèle par défaut pour les applications Microsoft Office <input checked="" type="radio"/> Utiliser un modèle personnalisé existant (URL, UNC ou URN) <input type="text" value="https://collaboration/sites/TeamSchulung12/Do"/> <a href="#">Modifier ce modèle</a> <input type="radio"/> Télécharger un modèle personnalisé existant (XSN) à utiliser <input type="text"/> <input type="button" value="Parcourir..."/> <a href="#">Créer un nouveau modèle personnalisé</a>
<b>Toujours afficher</b> Force l'affichage automatique du panneau Informations sur le document pour l'utilisateur dans certaines conditions.	<input type="checkbox"/> Toujours afficher le panneau Informations sur le document à l'ouverture d'un document et enregistrer initialement ce type de contenu

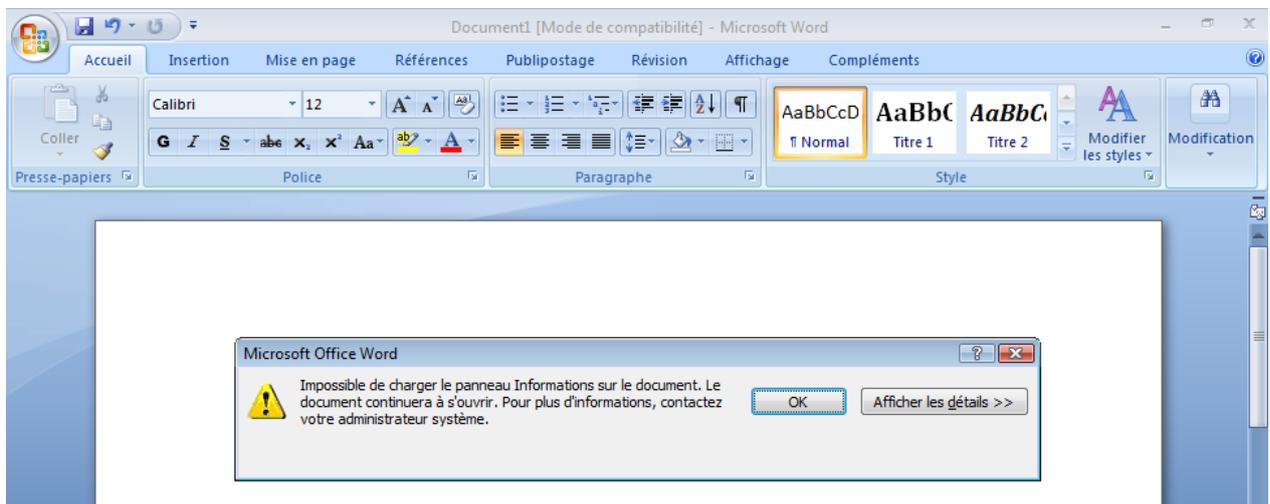
Nous validons par **OK** et si maintenant nous créons un nouveau document:



nous avons:



**Attention!!!** Si vous créez ensuite d'autres colonnes dans la bibliothèque sans les ajouter au DIP, vous aurez le message suivant lors de la création d'un nouveau document:



Il vous faut alors rééditer le formulaire InfoPath en procédant comme avant à la différence que vous cliquerez cette fois-ci sur **Modifier ce modèle**:

## Paramètres du panneau Informations sur le document: Document

### Modèle du panneau Informations sur le document

Spécifiez le type de modèle du panneau Informations sur le document à afficher dans les applications compatibles avec Microsoft Office.

Remarque : la création d'un modèle personnalisé pour le panneau Informations sur le document requiert Microsoft Office InfoPath.  
[Plus d'informations sur la création de modèles du panneau Informations sur le document](#)

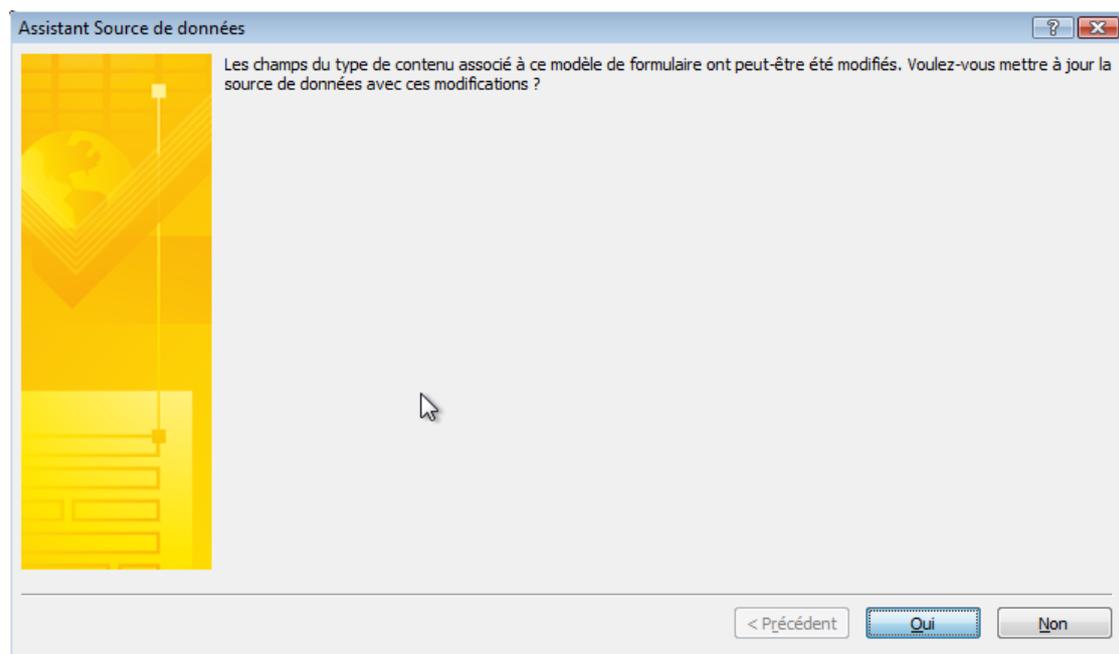
- Utiliser le modèle par défaut pour les applications Microsoft Office
- Utiliser un modèle personnalisé existant (URL, UNC ou URN)

[Modifier ce modèle](#)

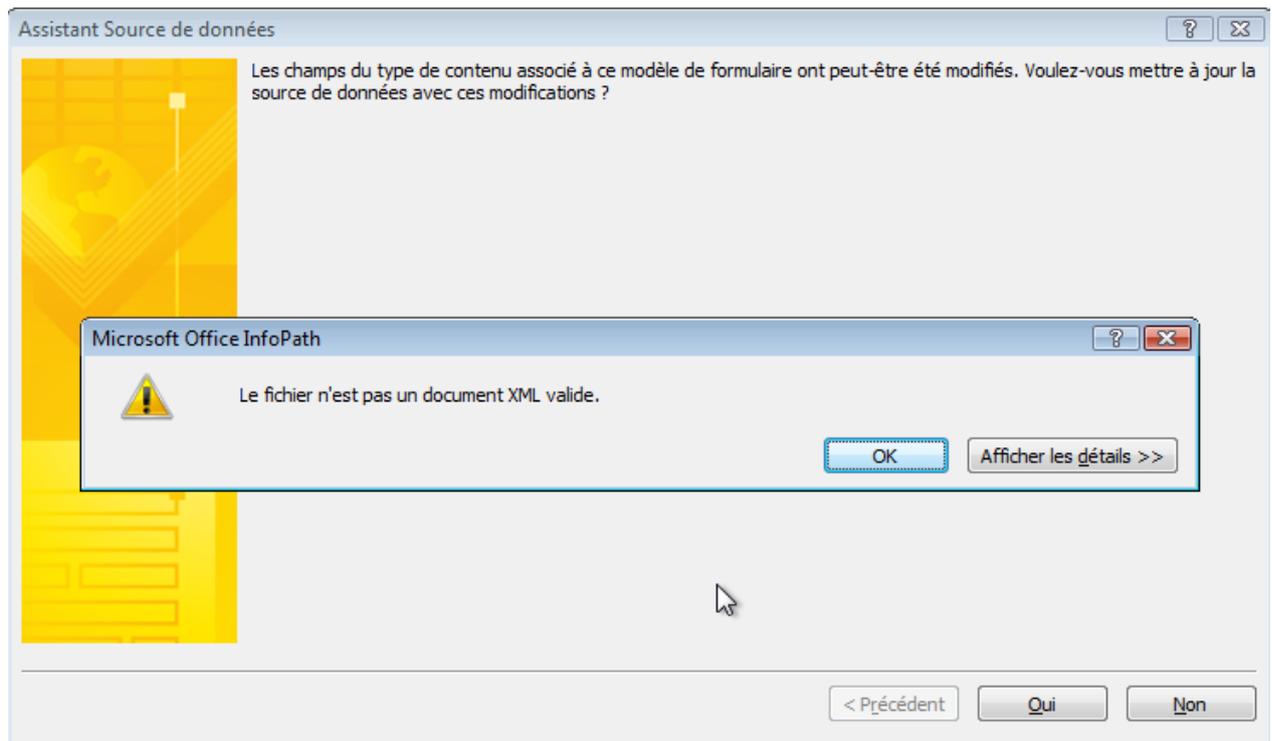
- Télécharger un modèle personnalisé existant (XSN) à utiliser

[Créer un nouveau modèle personnalisé](#)

Quand InfoPath s'ouvrira, vous aurez alors:



Si vous cliquez sur **Oui**, vous risquez d'avoir:



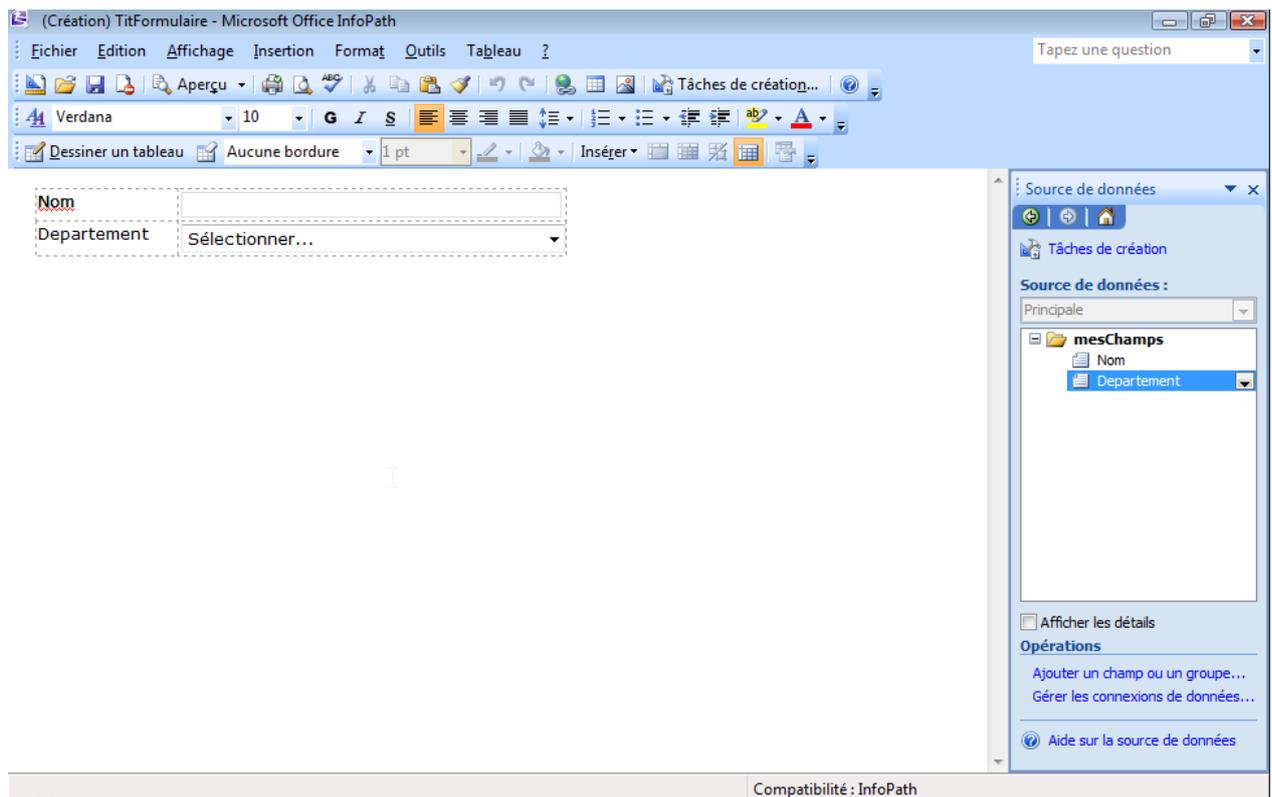
Dès lors, il n'existe pas à ma connaissance d'autres façons de procéder que de tout recommencer (ce qui de toute façon n'est pas bien long).

### ***11.3 Création de formulaires InfoPath avec flux SharePoint Designer (v2007)***

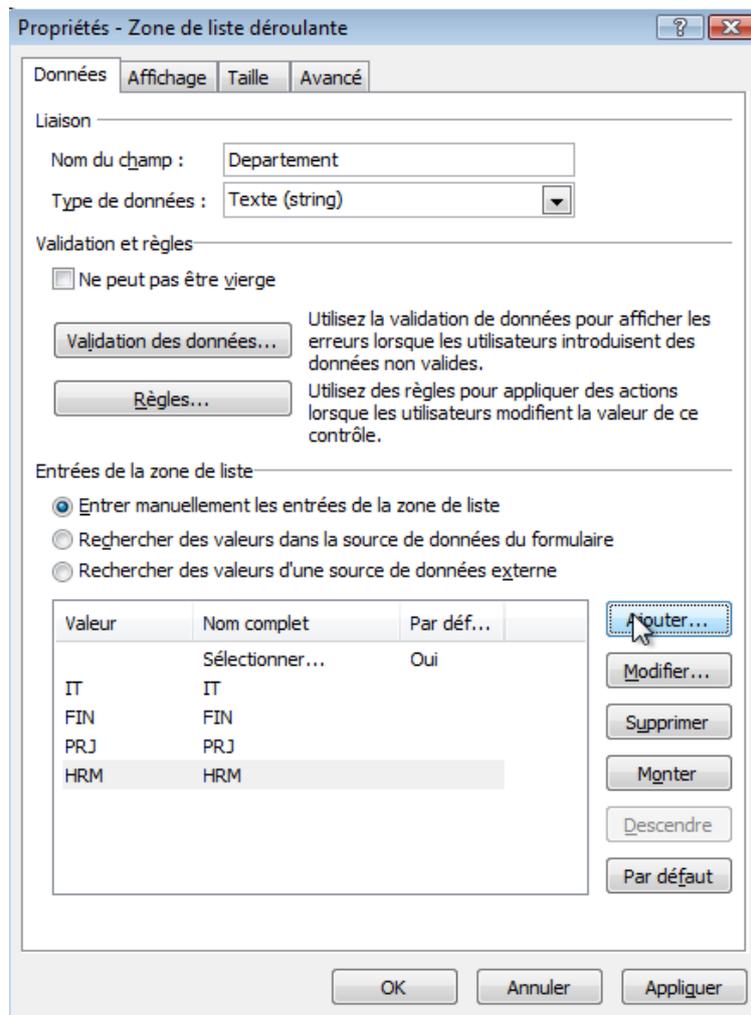
Pour étudier comment générer automatiquement des formulaires InfoPath à partir de flux, nous partirons d'un fichier InfoPath créé sur mesure pour l'exemple (nous ne pouvons pas utiliser les modèles de Microsoft comme point de départ car ils n'ont pas été conçus de manière conventionnelle).

Nous créons donc un formulaire avec un petit champ de texte nommé *Nom* et une liste déroulante nommée *Departement*:

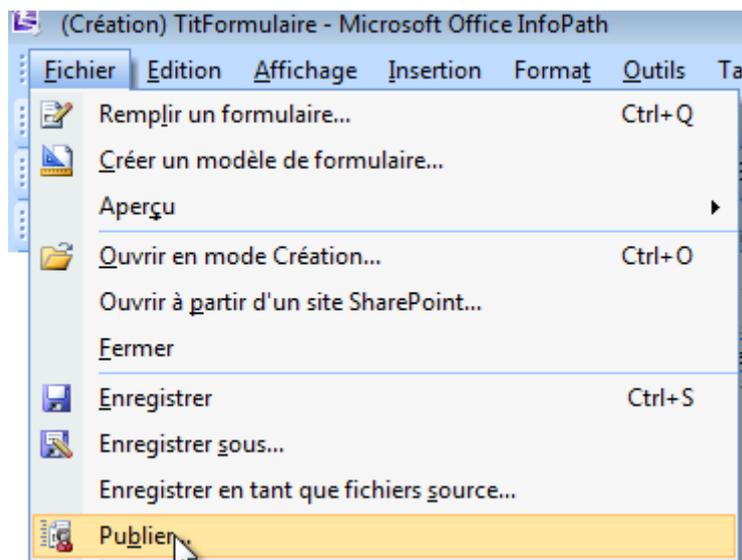
## Vincent ISOZ



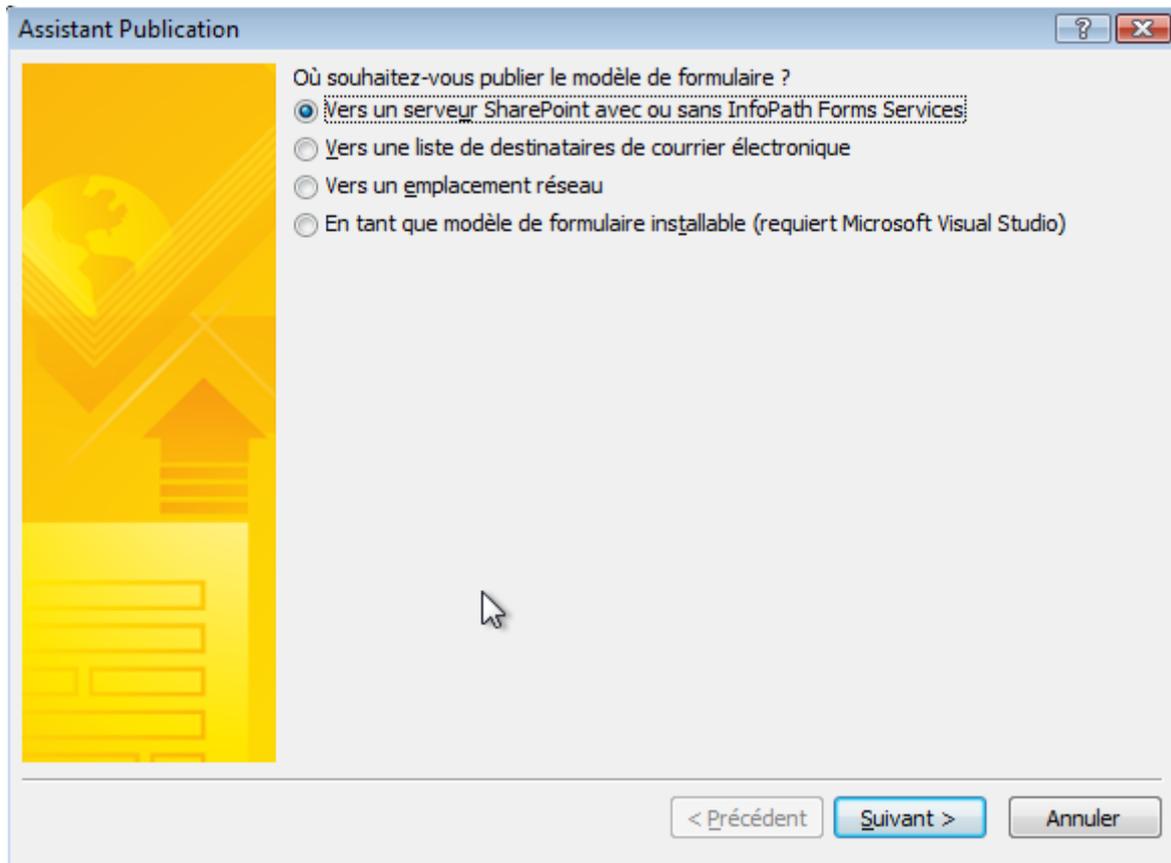
avec pour la liste déroulante *Departement*:



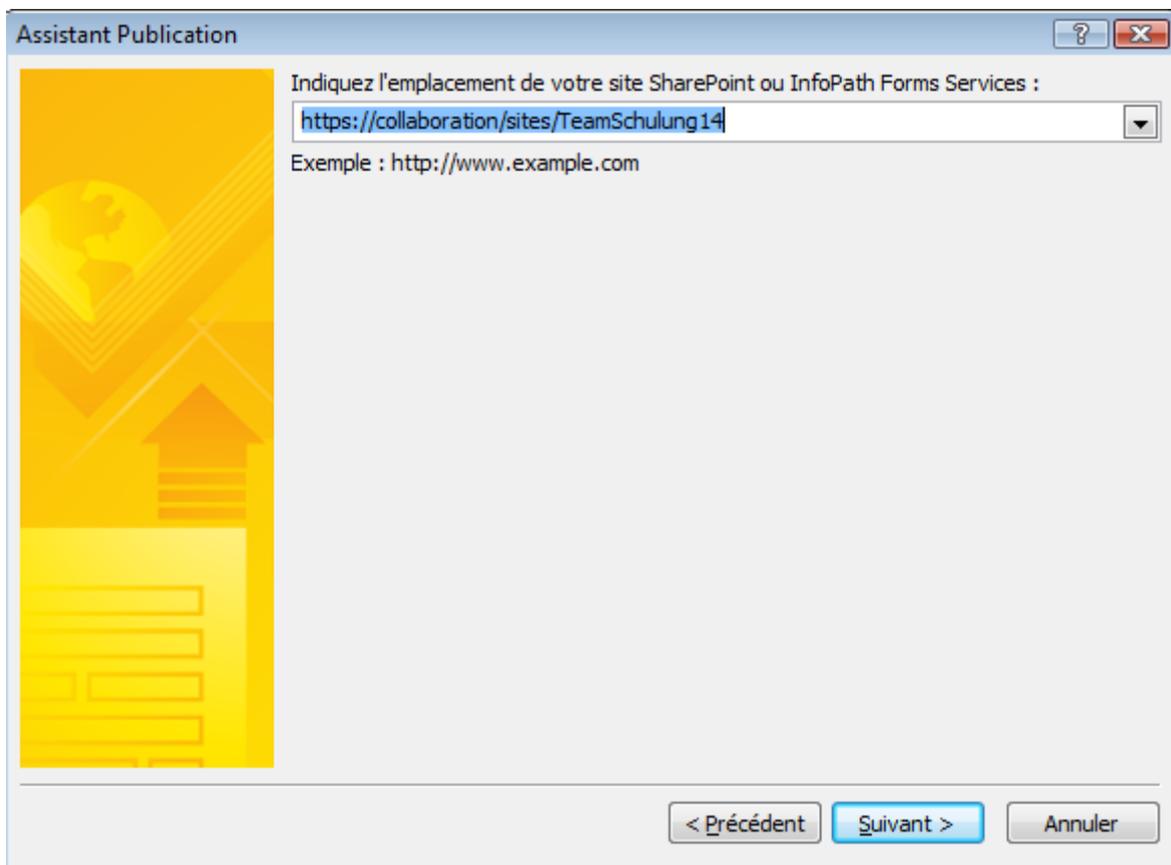
Et nous lançons la publication en supposant que nous n'avons pas InfoPath Services:



Il vient alors:

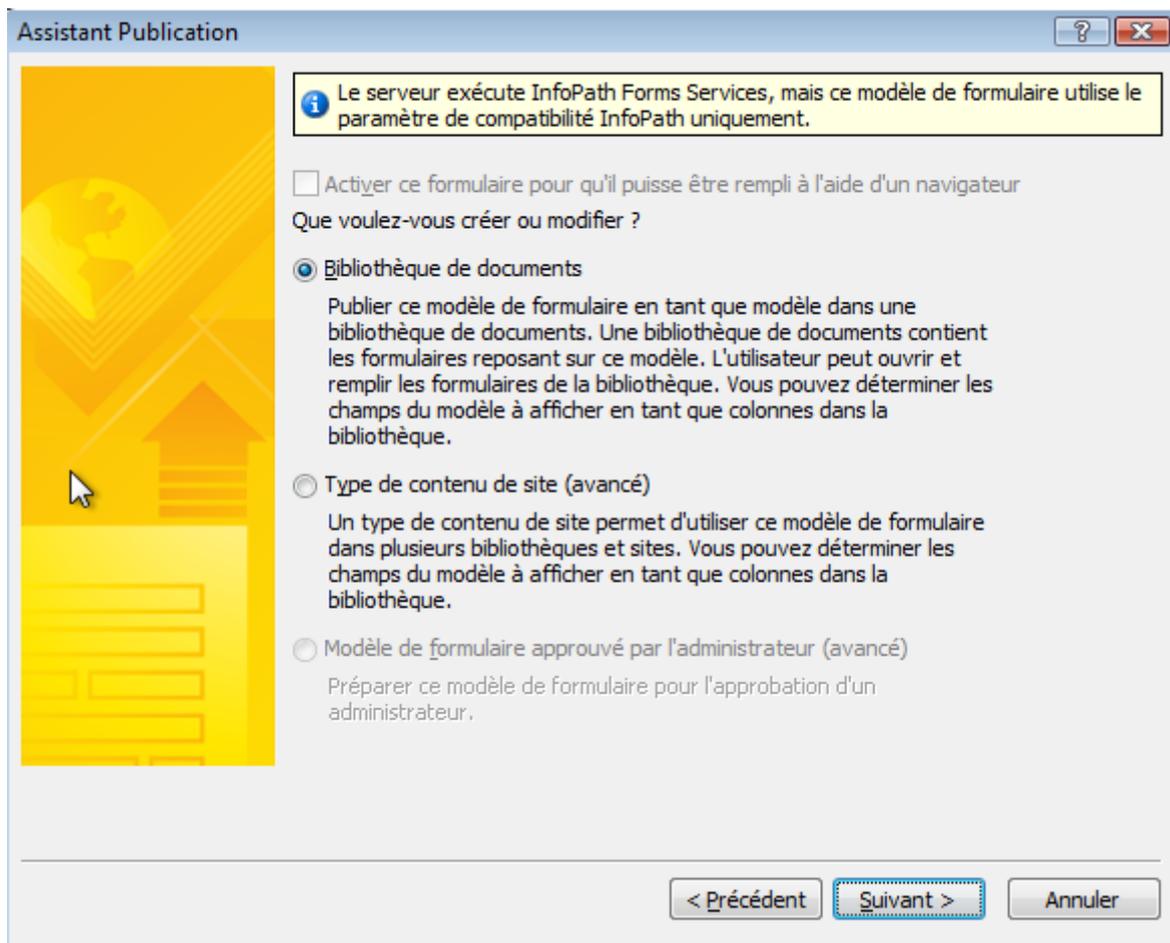


Nous cliquons sur **Suivant**:

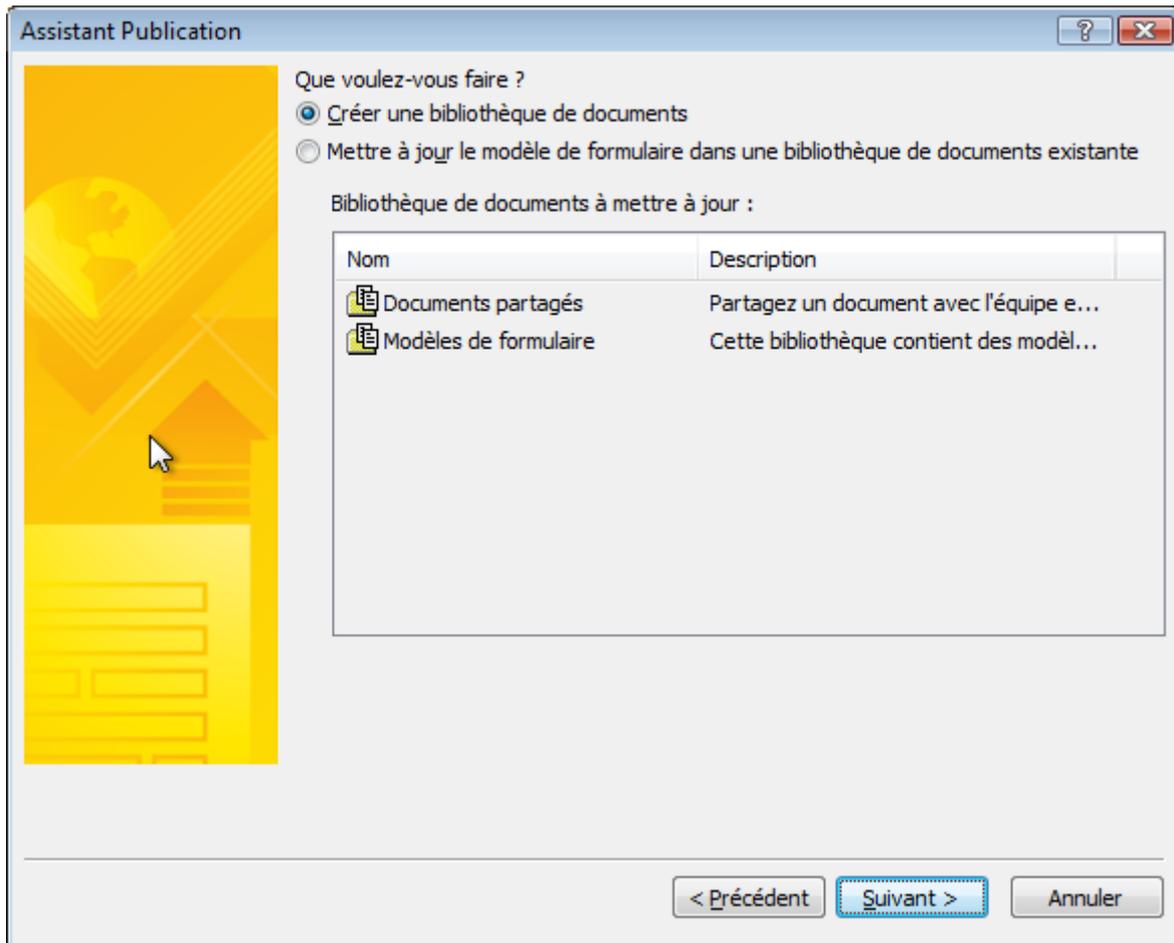


Vincent ISOZ

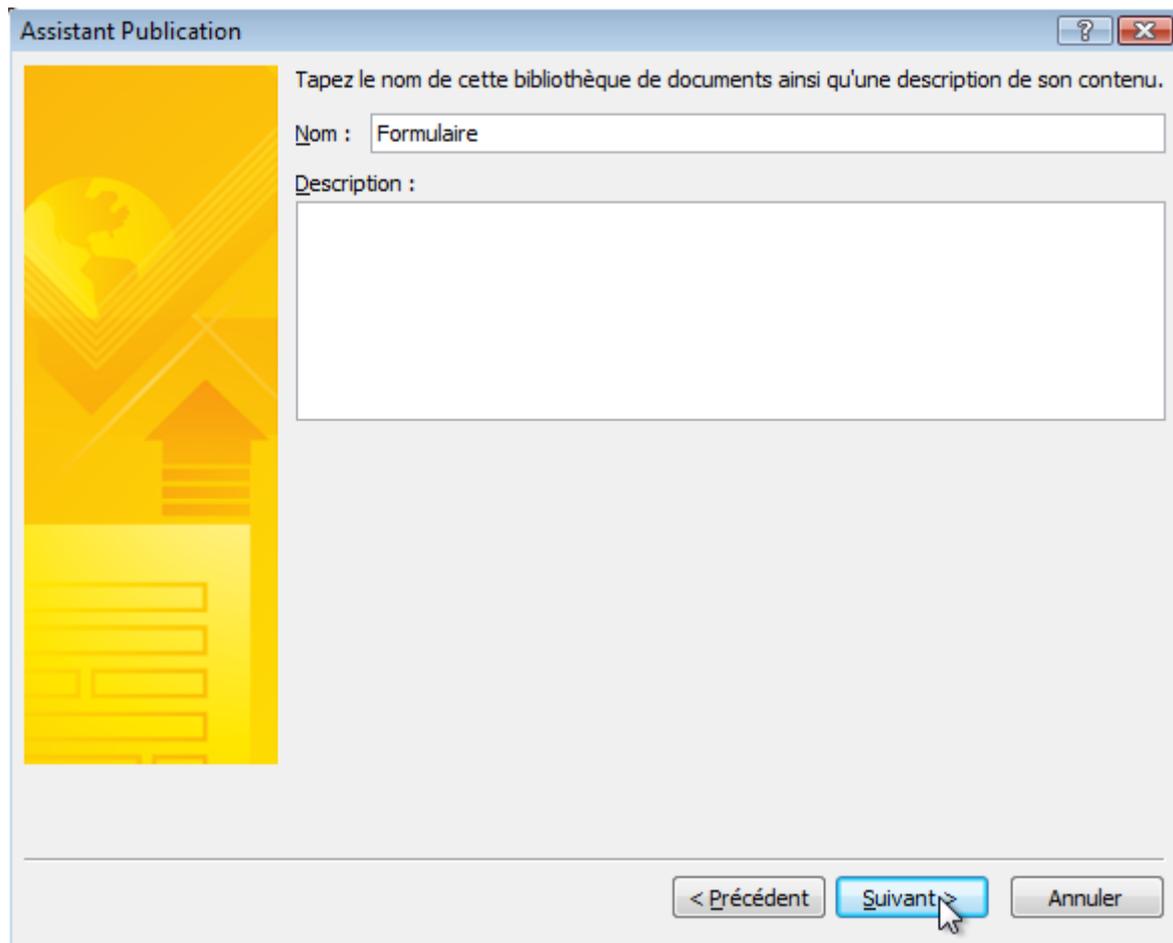
Nous cliquons sur **Suivant**:



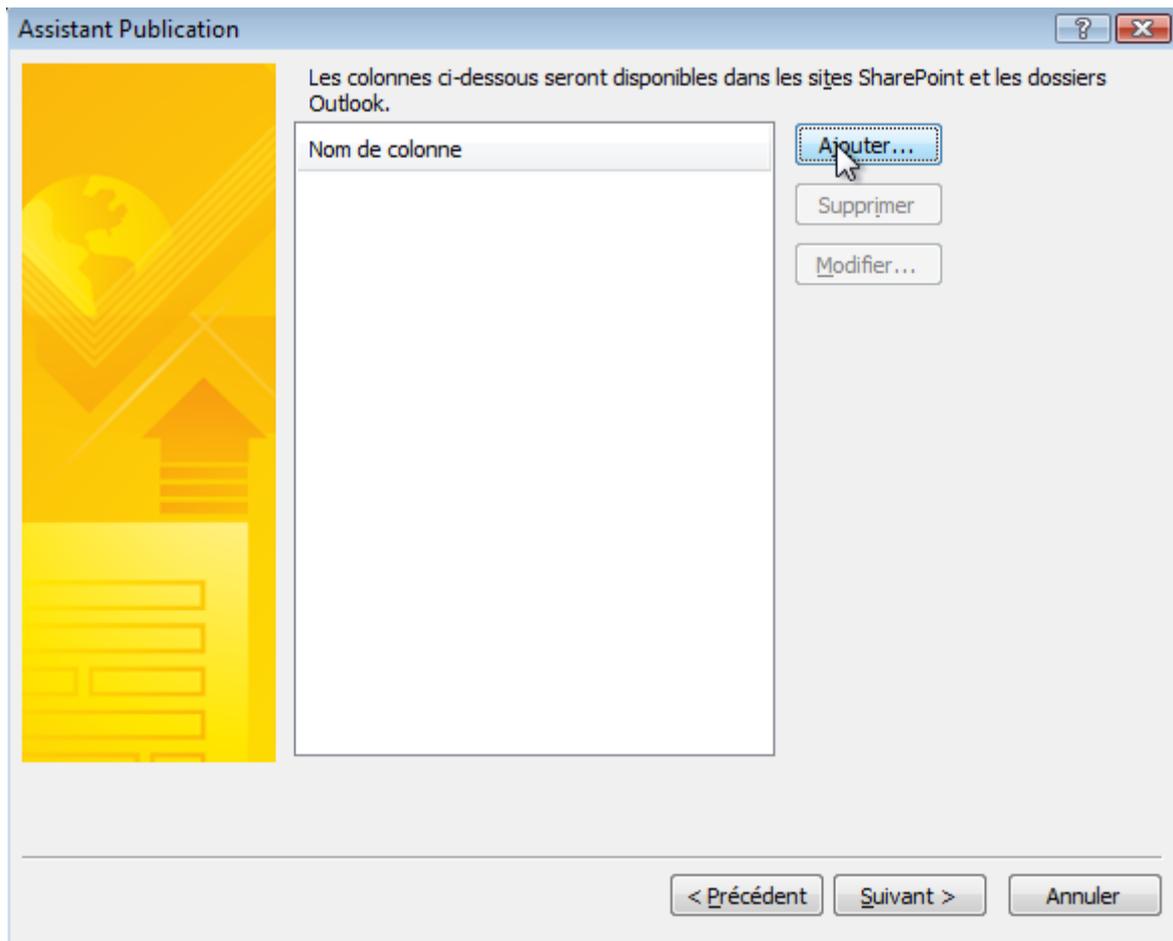
Nous cliquons sur **Suivant**:



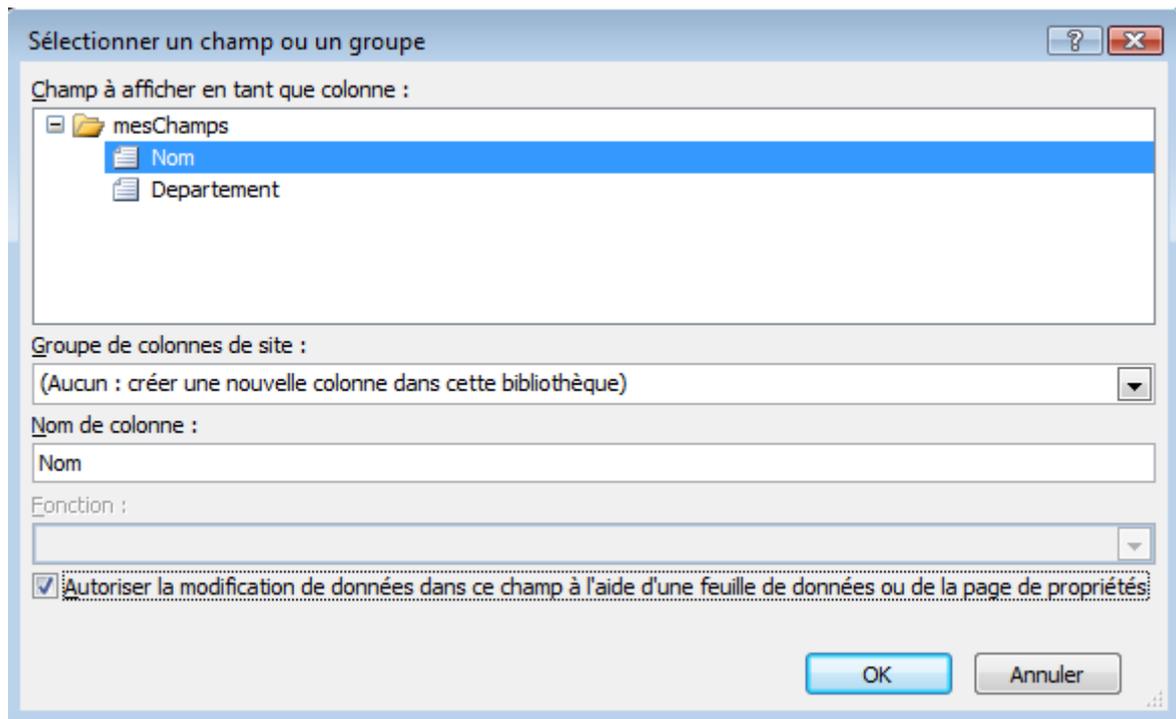
Nous cliquons sur **Suivant** et nous donnons un nom à notre bibliothèque:



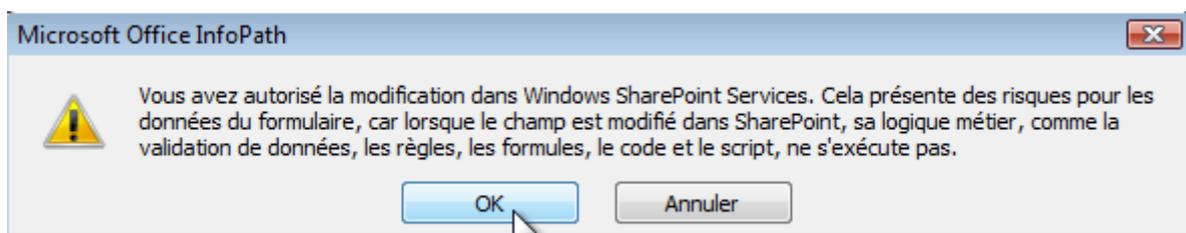
et nous cliquons sur **Suivant**:



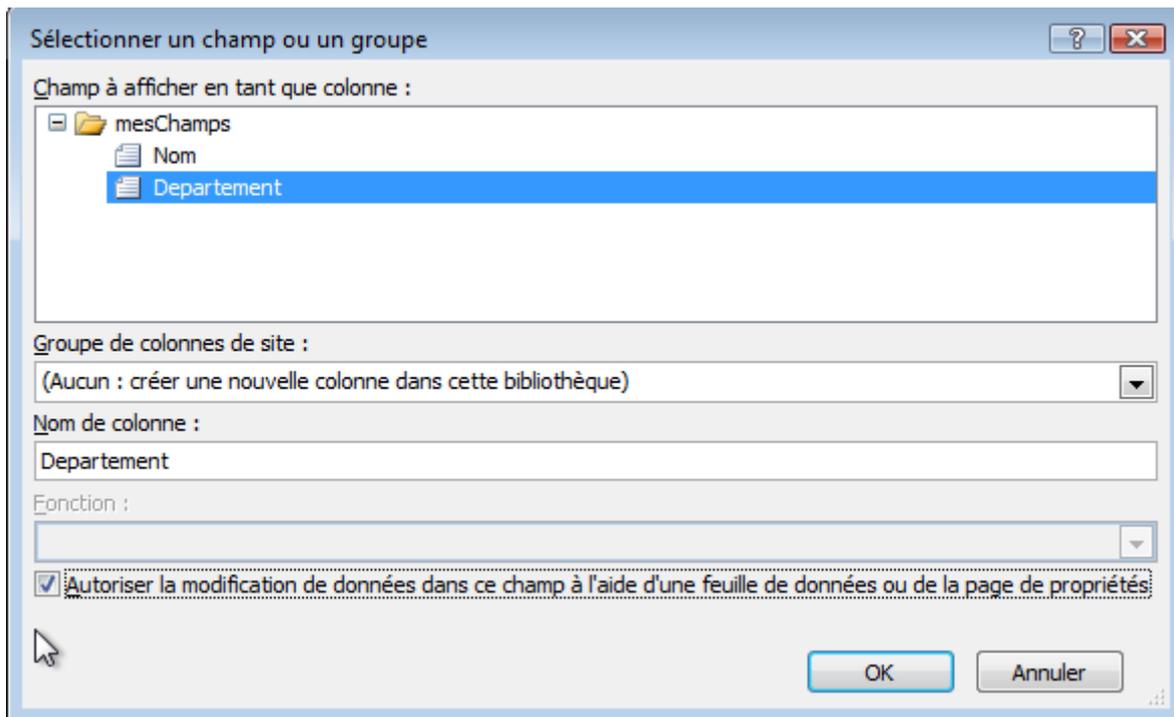
Nous cliquons sur **Ajouter** pour mettre à disposition de la bibliothèque des champs en prenant garde à ne **pas oublier de cocher Autoriser la modification de données dans ce champ à l'aide d'une feuille de données ou de la page de propriétés**:



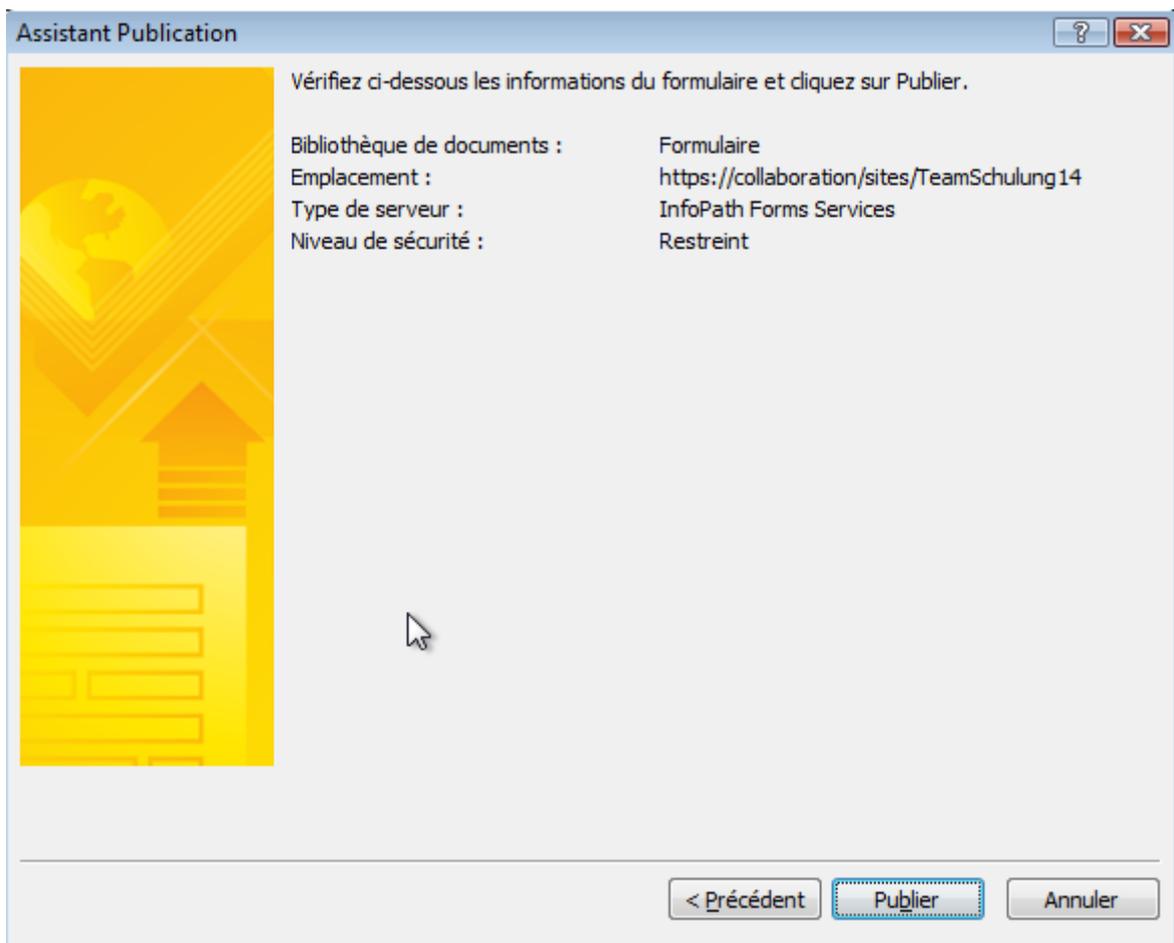
Quand nous validons par **OK**, il vient:



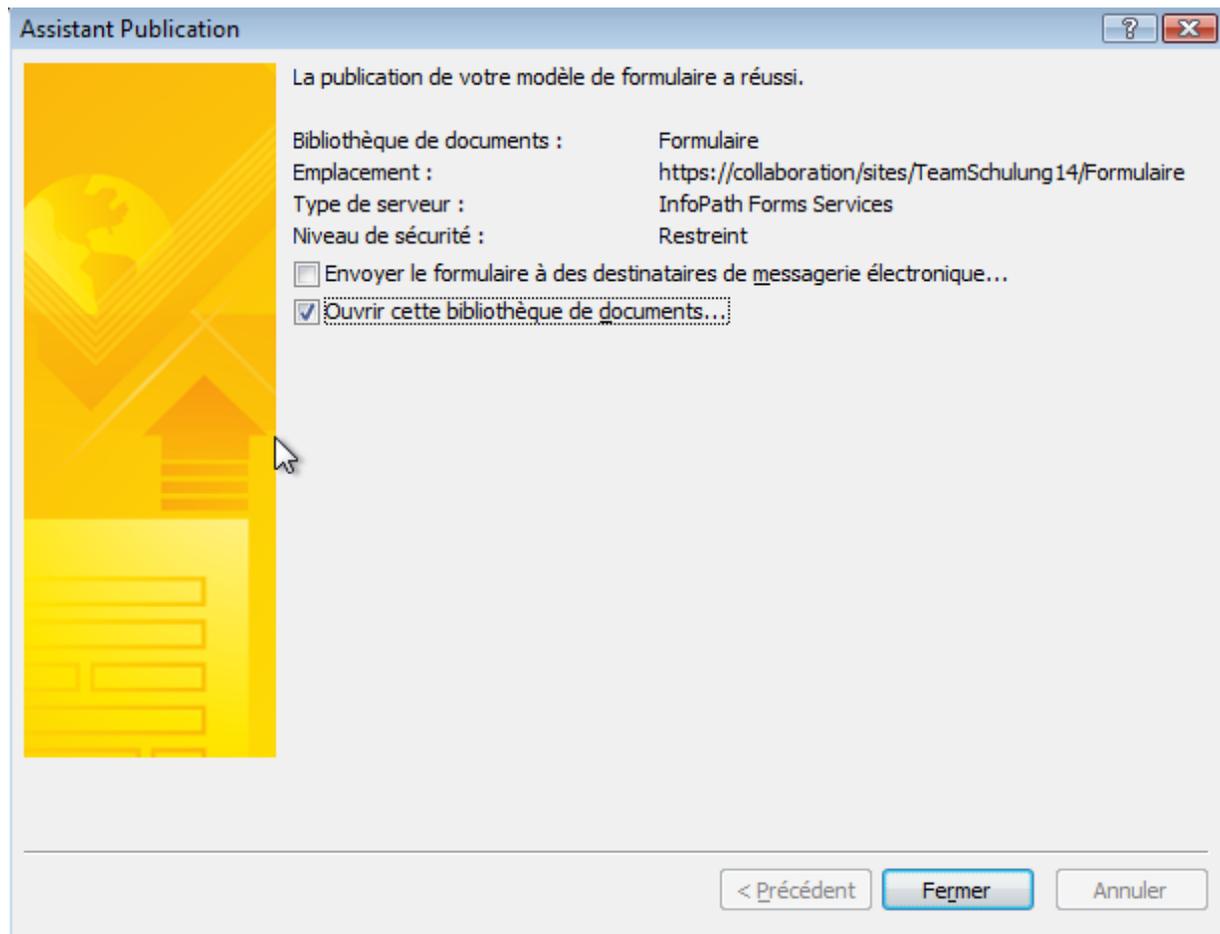
Nous validons par **OK** et faisons de même avec le deuxième champ:



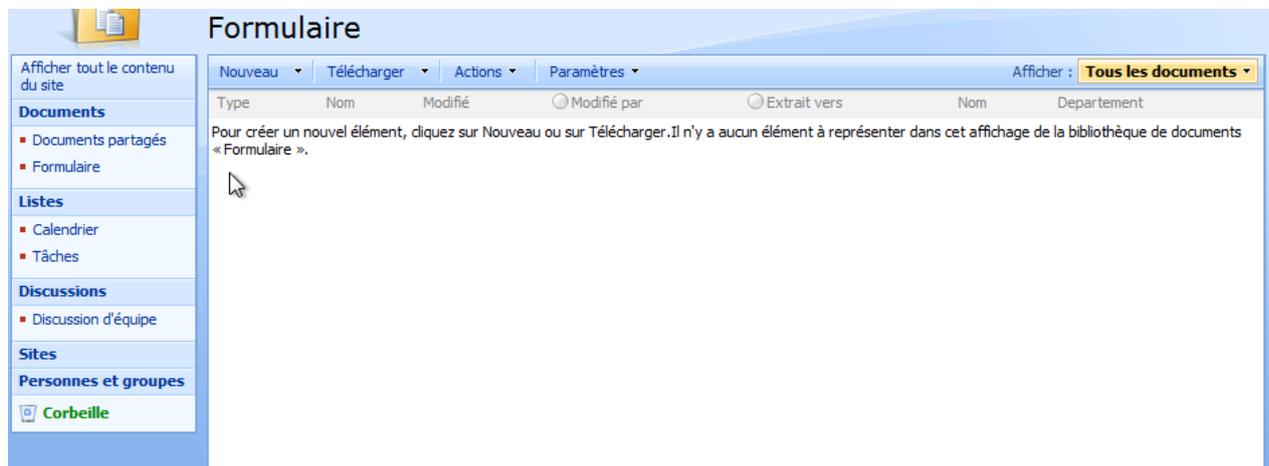
Nous validons par **OK** et cliquons sur **Suivant** pour arriver à:



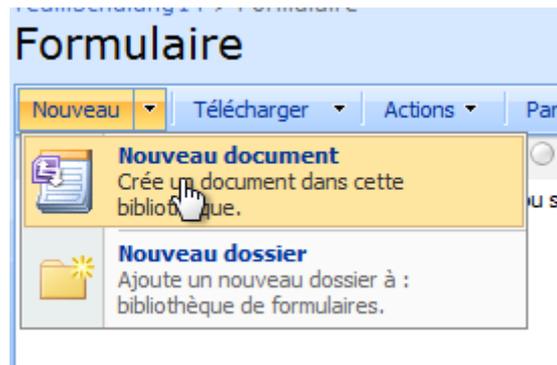
Nous cliquons sur **Publier**:



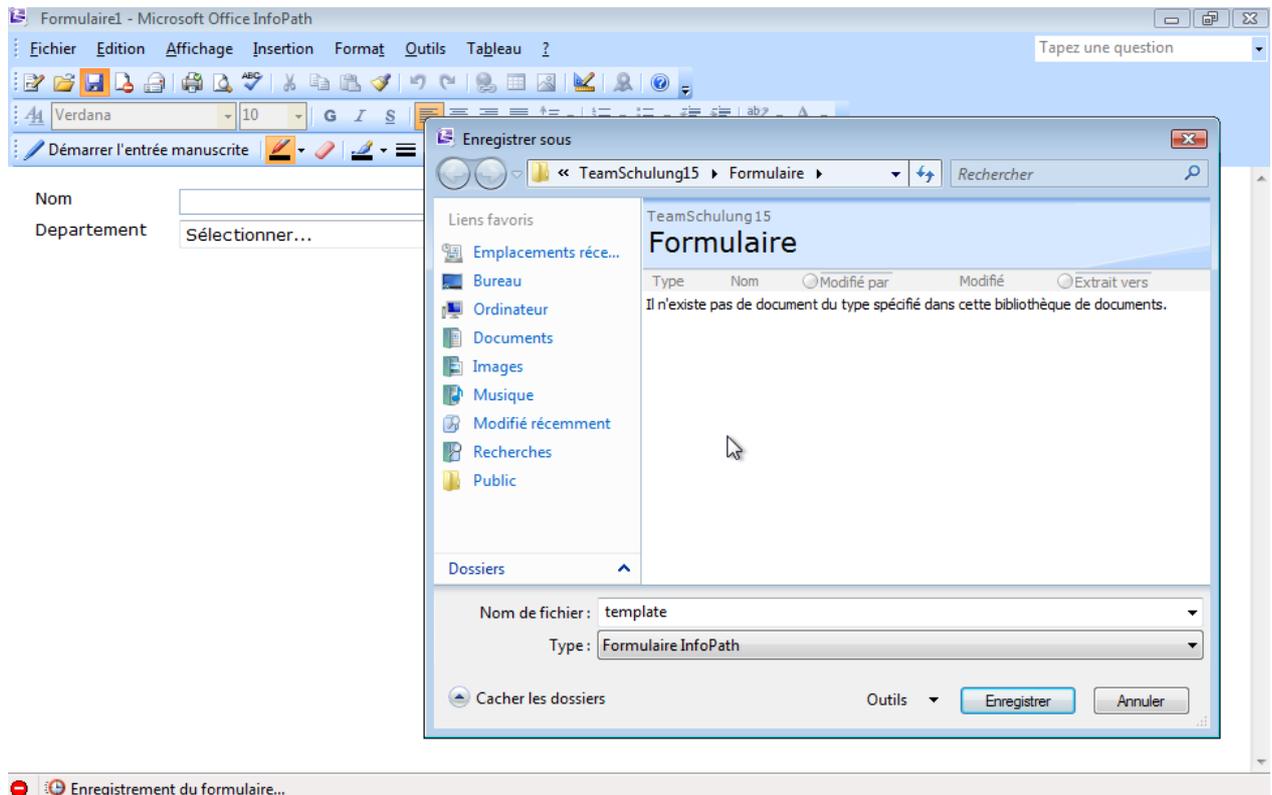
et nous cliquons sur **Fermer**:



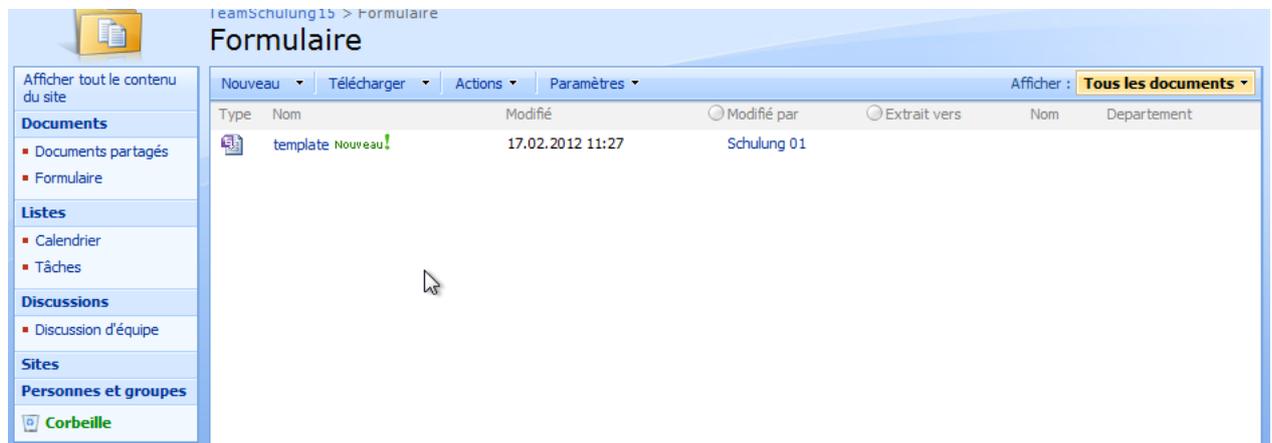
Maintenant vient une astuce très difficile à deviner pour créer un formulaire InfoPath à partir d'un flux. D'abord créez un formulaire vide à partir de notre modèle disponible maintenant dans SharePoint:



enregistrez-le sous le nom *template*:

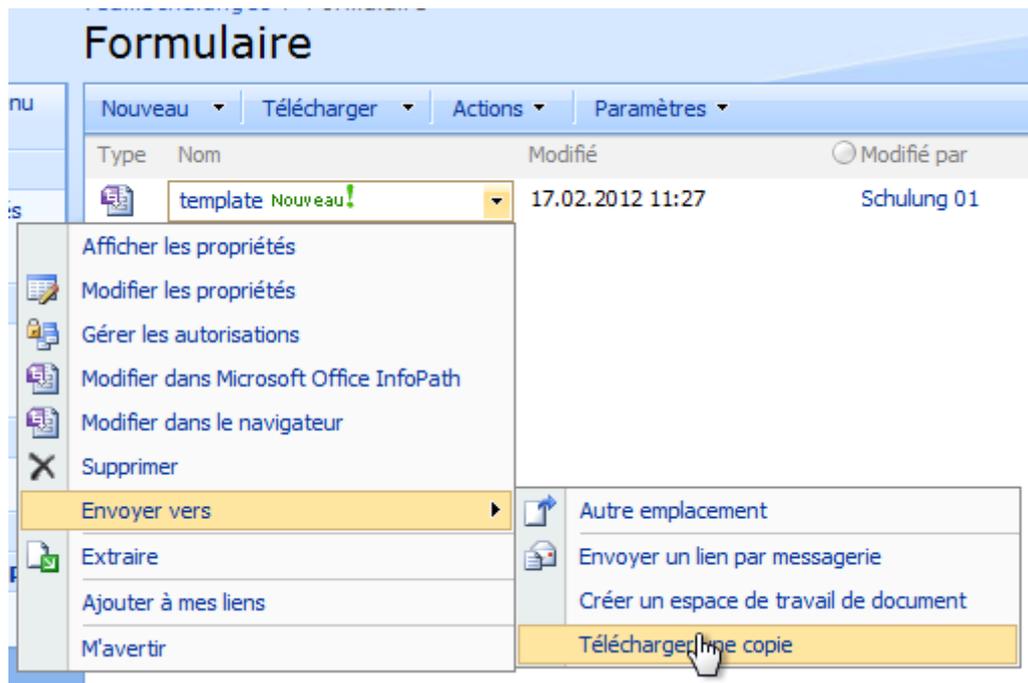


Vous aurez alors:

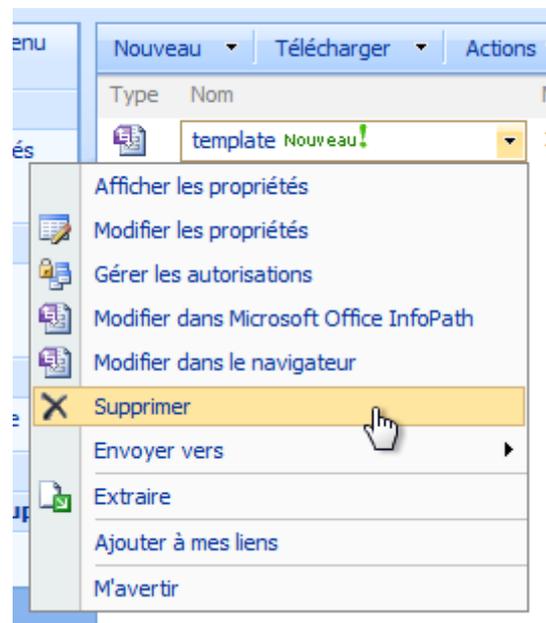


Vincent ISOZ

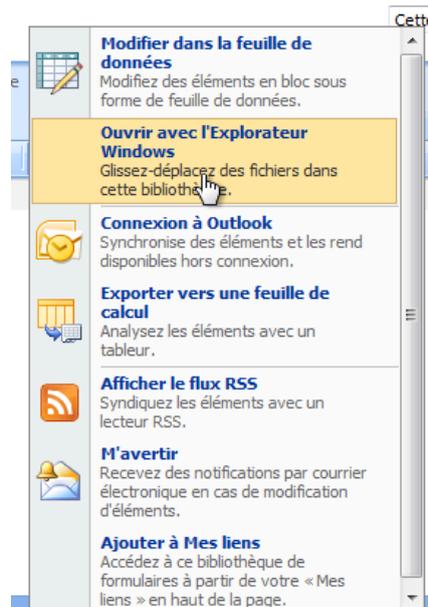
Ensuite, enregistrez-en une copie en local:



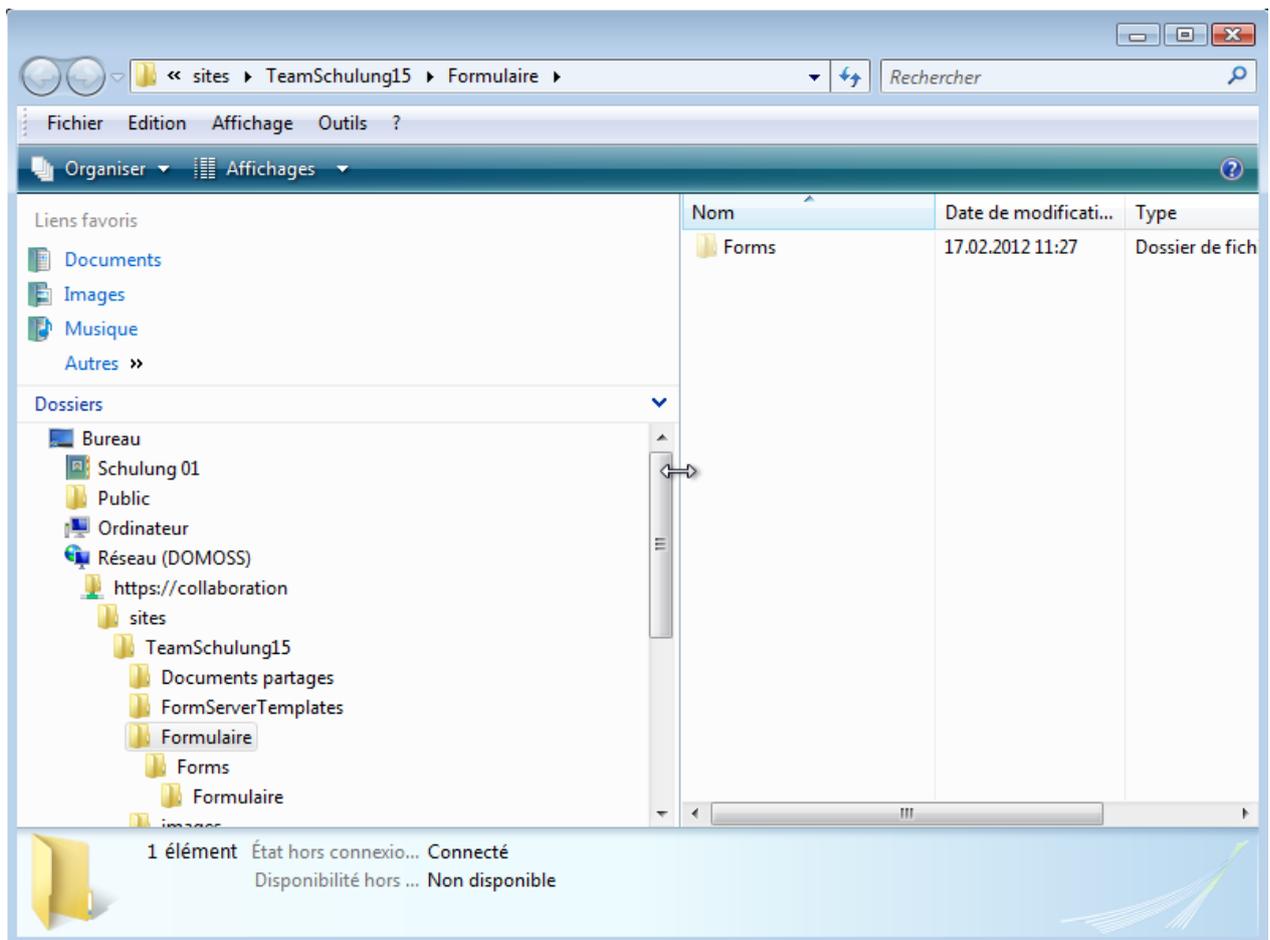
et supprimez le formulaire que vous avez créé dans SharePoint une fois le copie enregistrée en local à l'endroit de votre choix:



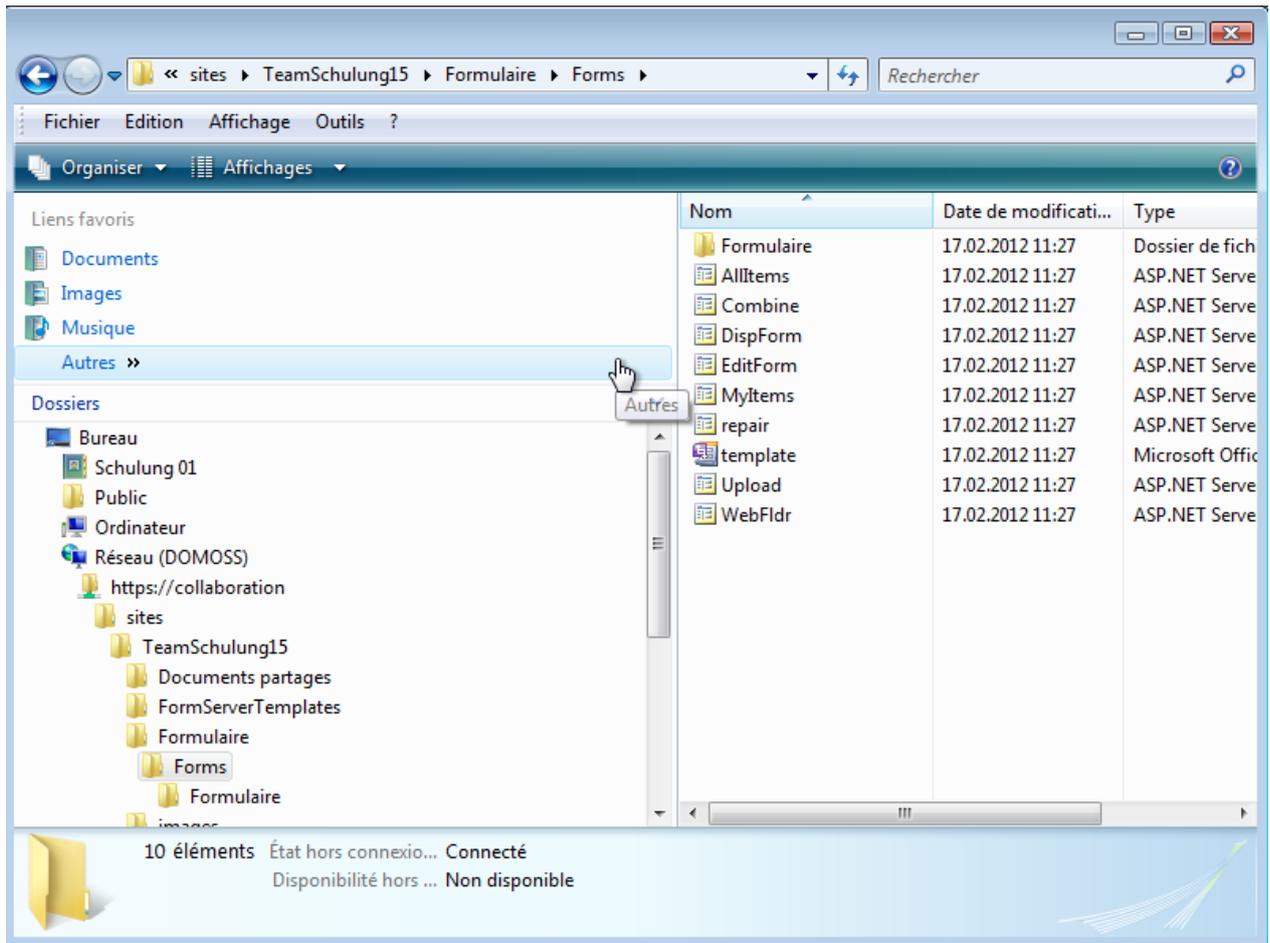
Maintenant, ouvrez la bibliothèque de formulaire en mode Explorateur:



Vous aurez alors un dossier caché *Forms* qui sera visible:



Entrez dans ce dossier et vous y verrez:



copiez-y le fichier *template.xml* que vous aviez enregistré en local précédemment. Vous aurez alors:



Vincent ISOZ

et changez l'URL du modèle de **Formulaire/Forms/template.xsn** à **Formulaire/Forms/template.xml** et validez par **OK** (il faut absolument valider par **OK** avant de faire la suite):

## Bibliothèque de formulaires - Paramètres avancés : Formulaire

<b>Types de contenu</b> Spécifiez si la gestion des types de contenu peut être autorisée sur bibliothèque de formulaires. Chaque type de contenu apparaît sur le bouton Nouveau et peut avoir un jeu de colonnes, de flux de travail et d'autres comportements uniques.	Autoriser la gestion des types de contenu ? <input type="radio"/> Oui <input checked="" type="radio"/> Non
<b>Modèle de document</b> Tapez l'adresse d'un modèle à utiliser comme base pour tous les nouveaux fichiers créés dans cette bibliothèque de documents. Lorsque plusieurs types de contenu sont activés, ce paramètre fonctionne par type de contenu. <a href="#">En savoir plus sur la configuration d'un modèle de bibliothèque.</a>	URL du modèle : <input type="text" value="Formulaire/Forms/template.xml"/> <a href="#">(Modifier le modèle)</a>

Ensuite, retournez dans **Paramètres avancés**:

TeamSchulung15 > Formulaire > Paramètres

## Personnaliser Formulaire

**Informations sur la liste**

Nom : Formulaire  
Adresse Web : <https://collaboration/sites/TeamSchulung15/Formulaire/Forms/AllItems.aspx>  
Description :

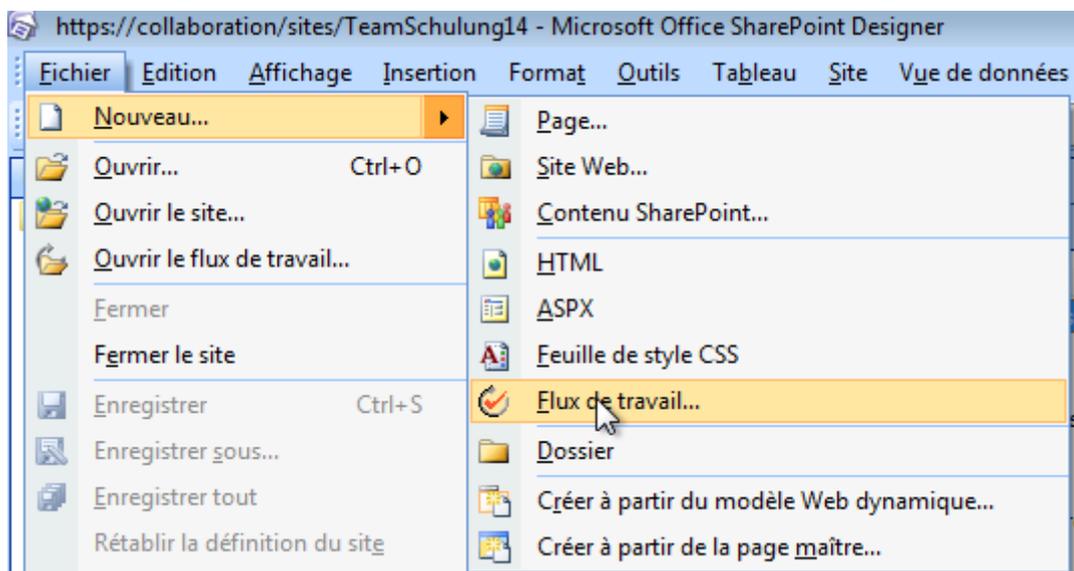
Paramètres généraux	Autorisations et gestion	Communications
<ul style="list-style-type: none"><li>Titre, description et navigation</li><li>Paramètres de contrôle de version</li><li><b>Paramètres avancés</b></li><li>Paramètres de ciblage de l'audience</li></ul>	<ul style="list-style-type: none"><li>Supprimer la liste bibliothèque de formulaires</li><li>Enregistrer la liste bibliothèque de formulaires en tant que modèle</li><li>Autorisations pour : bibliothèque de formulaires</li><li>Gérer les fichiers extraits</li><li>Recréer un lien entre ces documents et cette bibliothèque</li><li>Paramètres du flux de travail</li></ul>	<ul style="list-style-type: none"><li>Paramètres RSS</li></ul>

et cliquez sur **Oui** pour **Autoriser la gestion des types de contenu?** et validez par **OK**:

## Bibliothèque de formulaires - Paramètres avancés : Formulaire

<b>Types de contenu</b> Spécifiez si la gestion des types de contenu peut être autorisée sur bibliothèque de formulaires. Chaque type de contenu apparaît sur le bouton Nouveau et peut avoir un jeu de colonnes, de flux de travail et d'autres comportements uniques.	Autoriser la gestion des types de contenu ? <input checked="" type="radio"/> Oui <input type="radio"/> Non
<b>Modèle de document</b> Tapez l'adresse d'un modèle à utiliser comme base pour tous les nouveaux fichiers créés dans cette bibliothèque de documents. Lorsque plusieurs types de contenu sont activés, ce paramètre fonctionne par type de contenu. <a href="#">En savoir plus sur la configuration d'un modèle de bibliothèque.</a>	URL du modèle : <input type="text" value="Formulaire/Forms/template.xml"/>

Ensuite, nous créons un flux dans SharePoint Designer (voir mes livres électroniques sur SharePoint pour plus d'informations sur le sujet):



Nous donnons un nom à notre flux et nous souhaiterions qu'un formulaire soit créé à chaque fois qu'une nouvelle tâche est créée dans le site:

Concepteur de flux de travail - Créer Formulaire

### Définir votre nouveau flux de travail

Choisissez le type de flux de travail et ses conditions de fonctionnement.

**Nommer ce flux de travail :**  
Creer Formulaire

**À quelle liste SharePoint ce flux de travail doit-il être attaché ?**  
Tâches

**Sélectionnez des options de démarrage du flux de travail pour les éléments dans Tâches :**

- Autoriser le démarrage manuel de ce flux de travail à partir d'un élément
- Démarrer automatiquement ce flux de travail lorsqu'un élément est créé
- Démarrer automatiquement ce flux de travail lorsqu'un élément est modifié

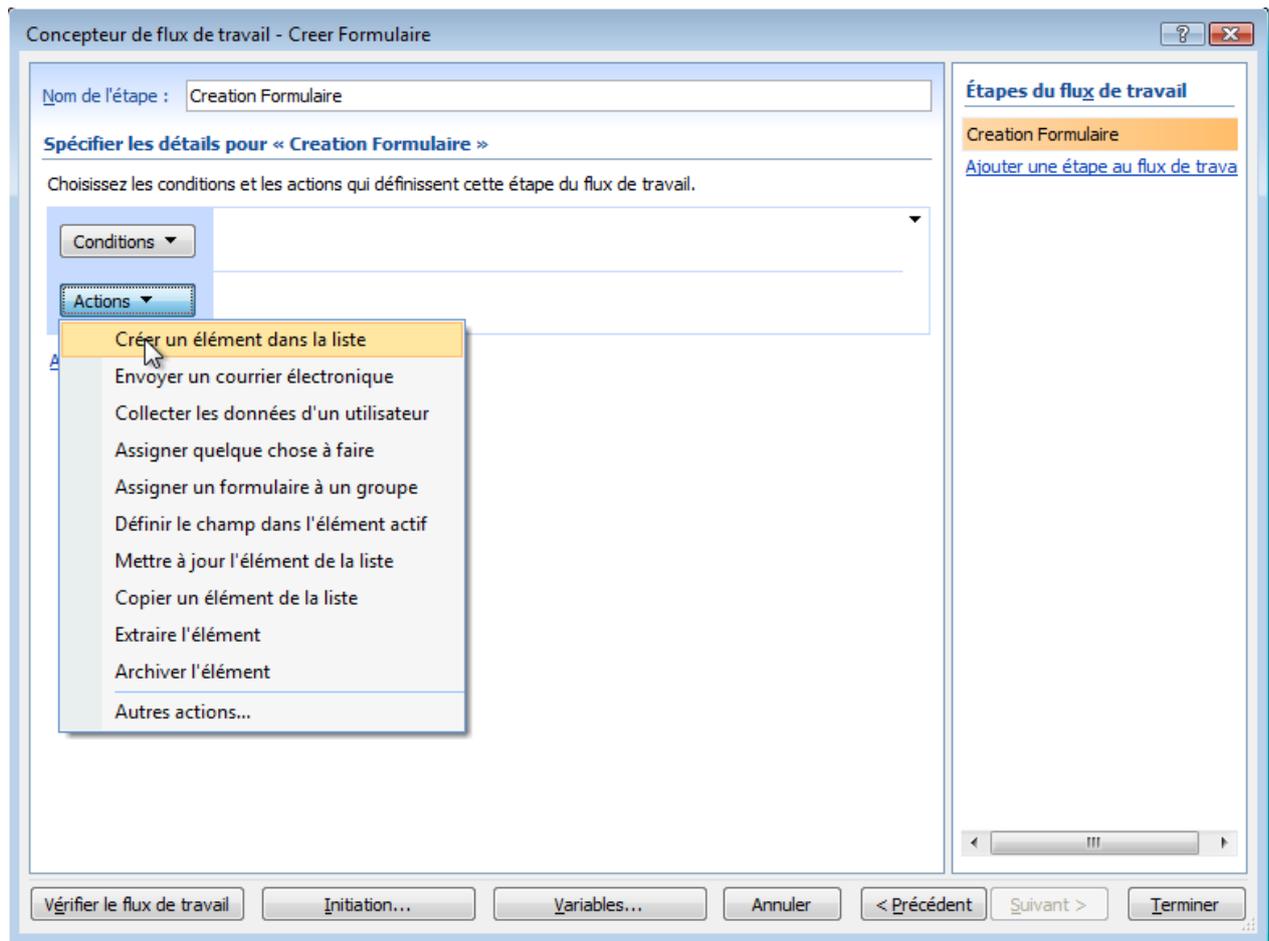
**Aide**

- Le nom de flux de travail indiqué ici permettra aux utilisateurs finaux d'identifier le flux de travail.
- Lorsque vous attachez ce flux de travail à une liste, vous pouvez commencer ce dernier pour les éléments de la liste.
- Les options de démarrage indiquent quand le flux de travail sera exécuté pour les éléments de la liste attachée.

[Cliquez ici pour obtenir une aide supplémentaire...](#)

Vérifier le flux de travail    Initiation...    Variables...    Annuler    < Précédent    Suivant >    Terminer

Nous cliquons sur **Suivant** et prenons l'action **Créer un élément dans la liste**:



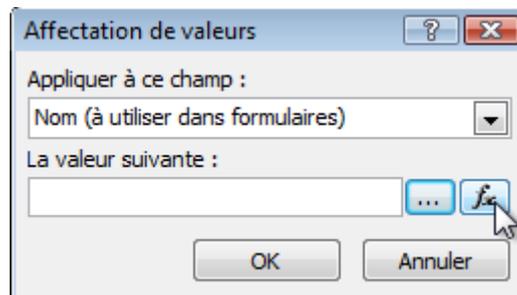
Nous avons alors:



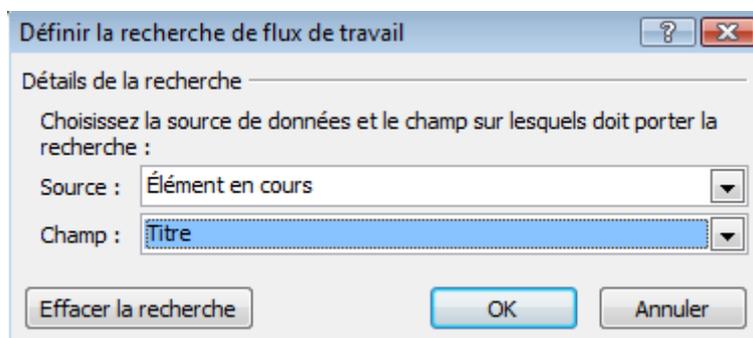
Nous cliquons sur **cette liste** et s'ouvre alors:



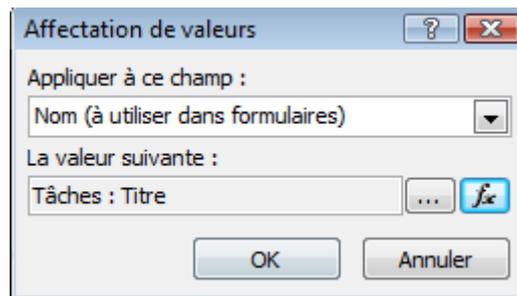
Nous cliquons sur **Modifier**:



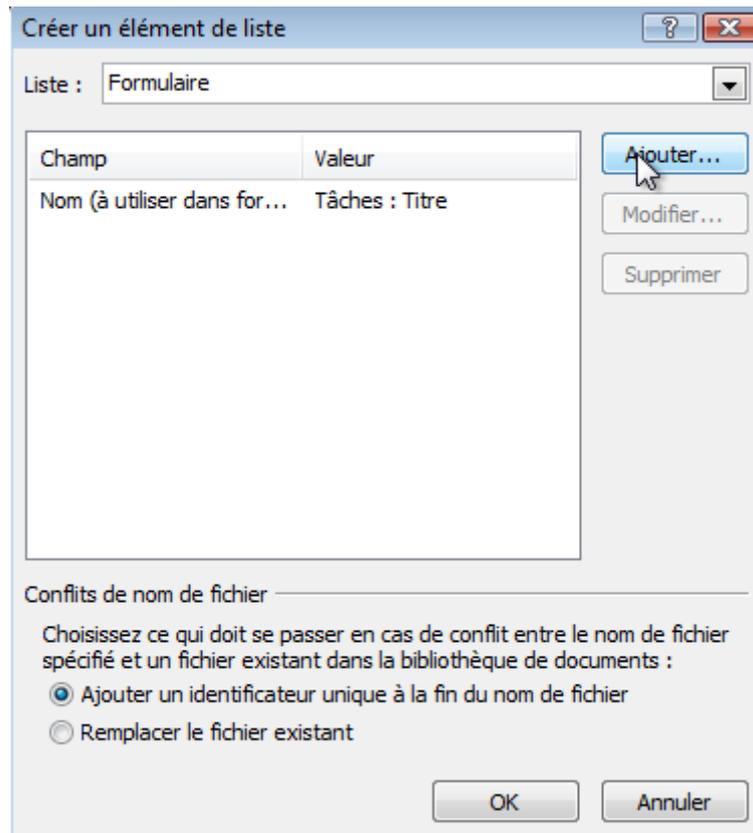
et nous affectons le *Titre* de l'**Élément en cours** au *Nom (à utiliser dans formulaires)*:



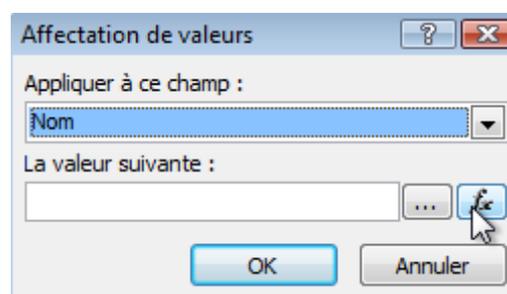
ça c'est pour le nom du fichier. Nous validons par **OK**:

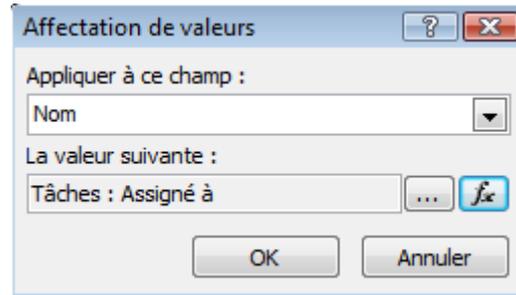
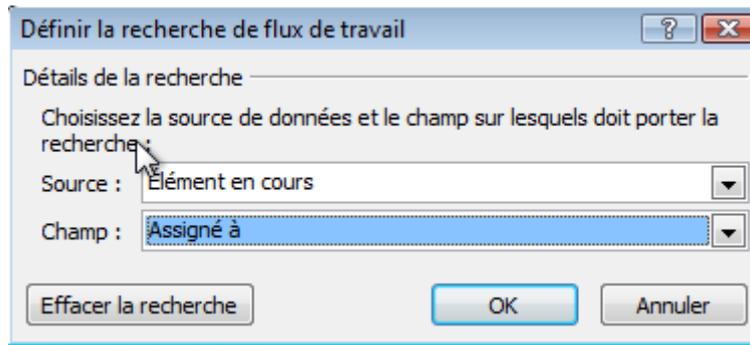


et encore un fois **OK** pour revenir à:

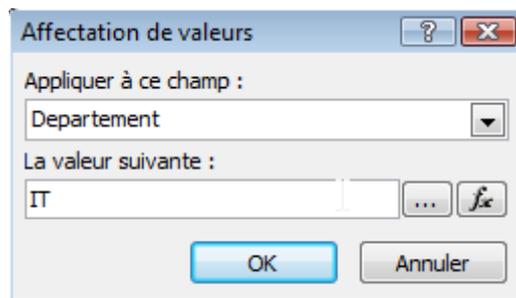


et nous cliquons sur **Ajouter** pour affecter le *Assigné à* au champ *Nom* de notre formulaire. Ce qui donne:

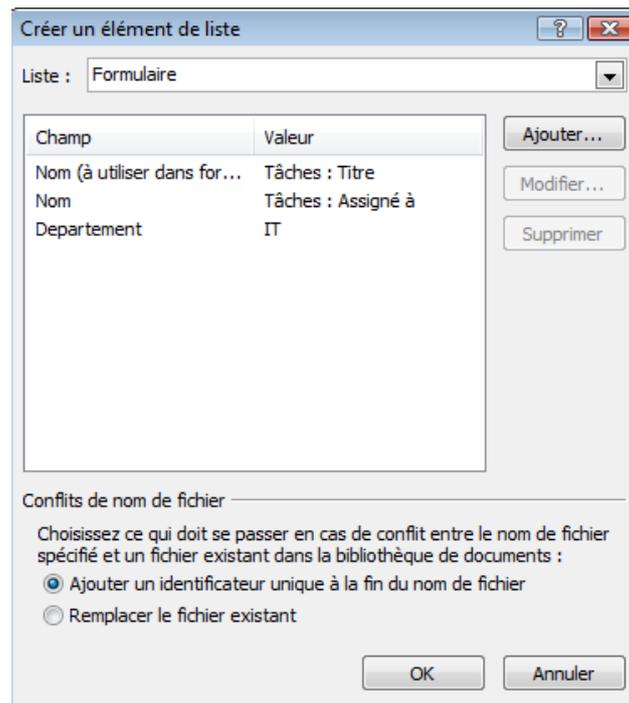




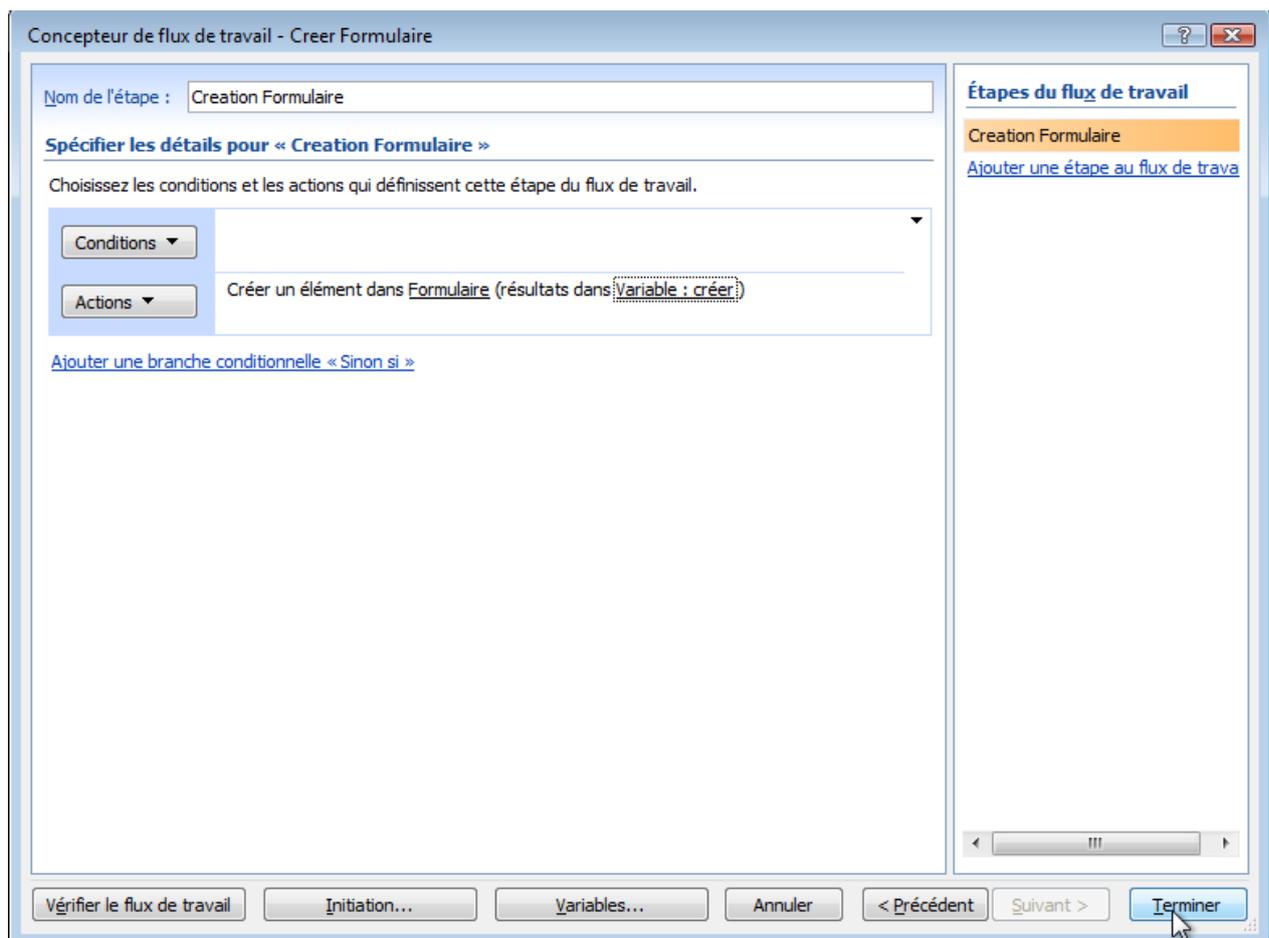
et nous procédons de même avec le champ *Departement* à la différence que nous lui affecterons une valeur constante nommée *IT*:



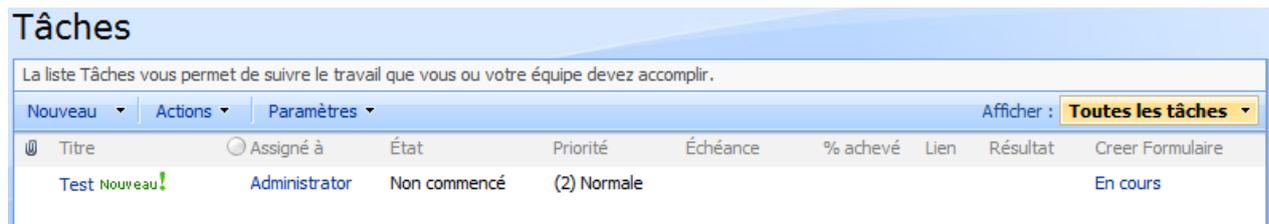
Ce qui donne au final:



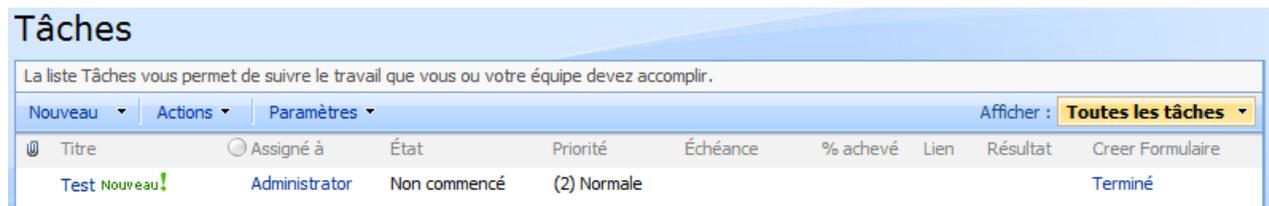
Soit:



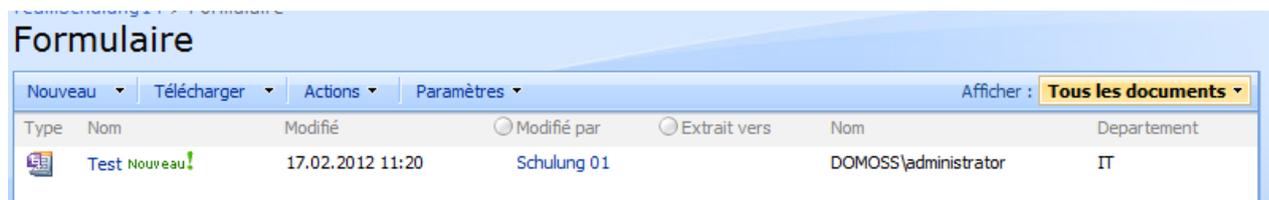
et nous cliquons sur **Terminer**. Ensuite, nous créons une tâche:



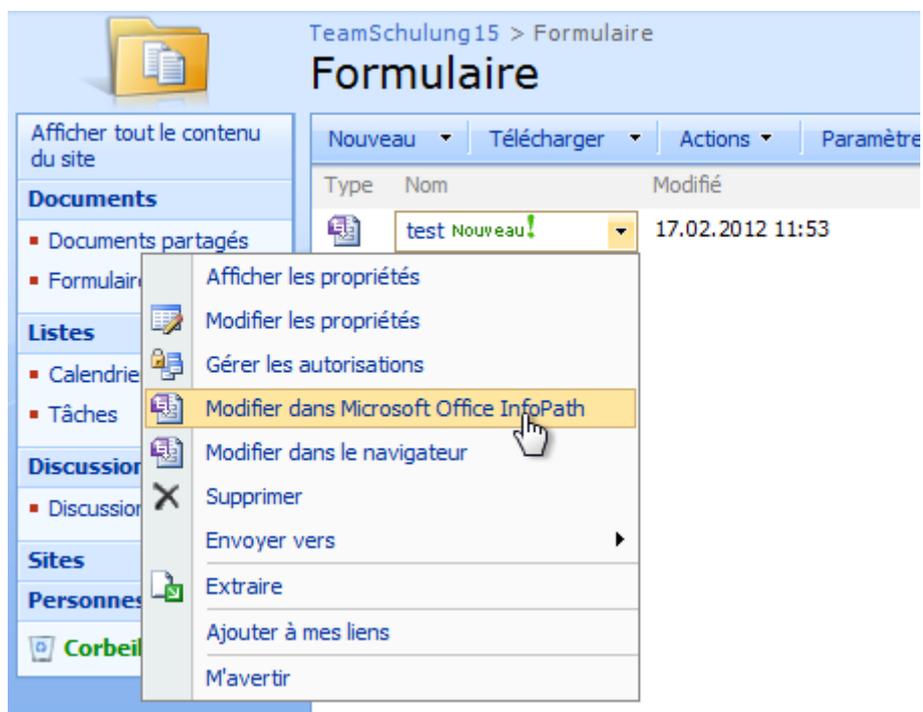
et nous voyons bien dans le colonne *Créer Formulaire* que le flux s'est lancé. Après quelques secondes ou minutes, il suffit de faire d'actualiser la page pour voir que le flux est *Terminé*:



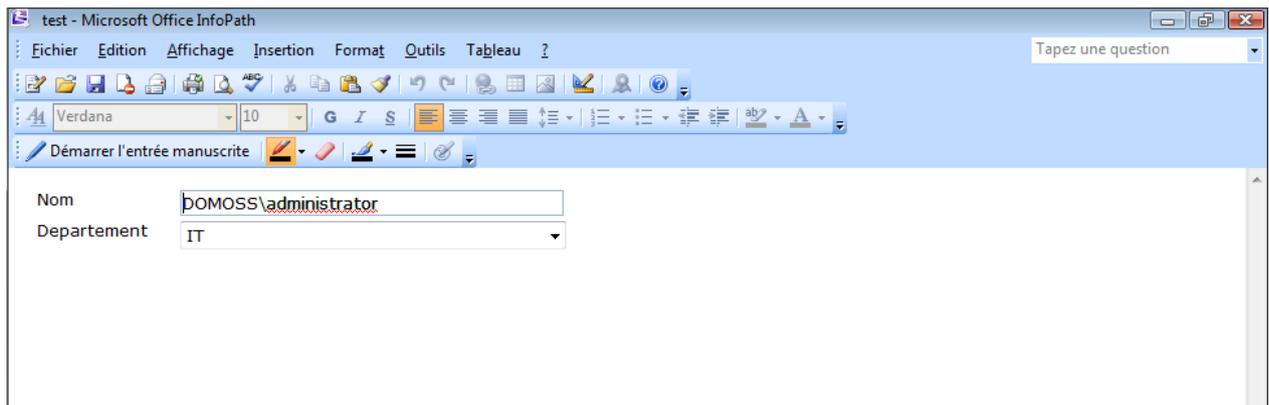
Une fois ceci fait, si nous retournons dans note bibliothèque, nous observons qu'un formulaire a bien été créé:



et nous le modifions pour bien vérifier que les données ont été rapatriées dedans:



et bingo!:



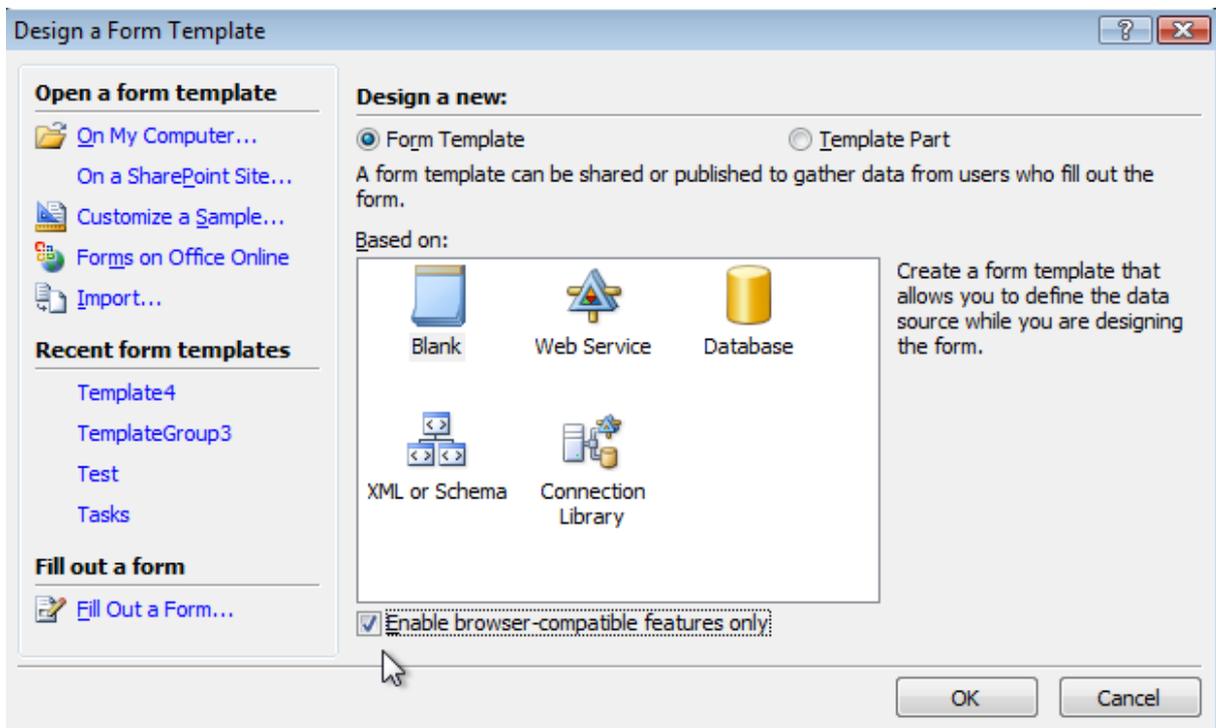
### 11.4 Publier sur InfoPath Services (v2007)

Une nouveauté depuis la version 2007 d'InfoPath est la publication d'un formulaire sur la version 1.0 d'InfoPath Services.

Évidemment cette dernière à quelques contraintes étant donné que c'est la toute première version. La plus embêtante étant le fait qu'elle ne supporte pas les filtres entre listes déroulantes et de nombreux contrôles. Mais ceci sera certainement corrigé dans une version ultérieure (ce qui est corrigé maintenant depuis la version 2010).

D'abord avant de pouvoir publier sur InfoPath Services il faut s'assurer que le formulaire est compatible avec cette technologie. Pour ce faire, il y a basiquement deux solutions.

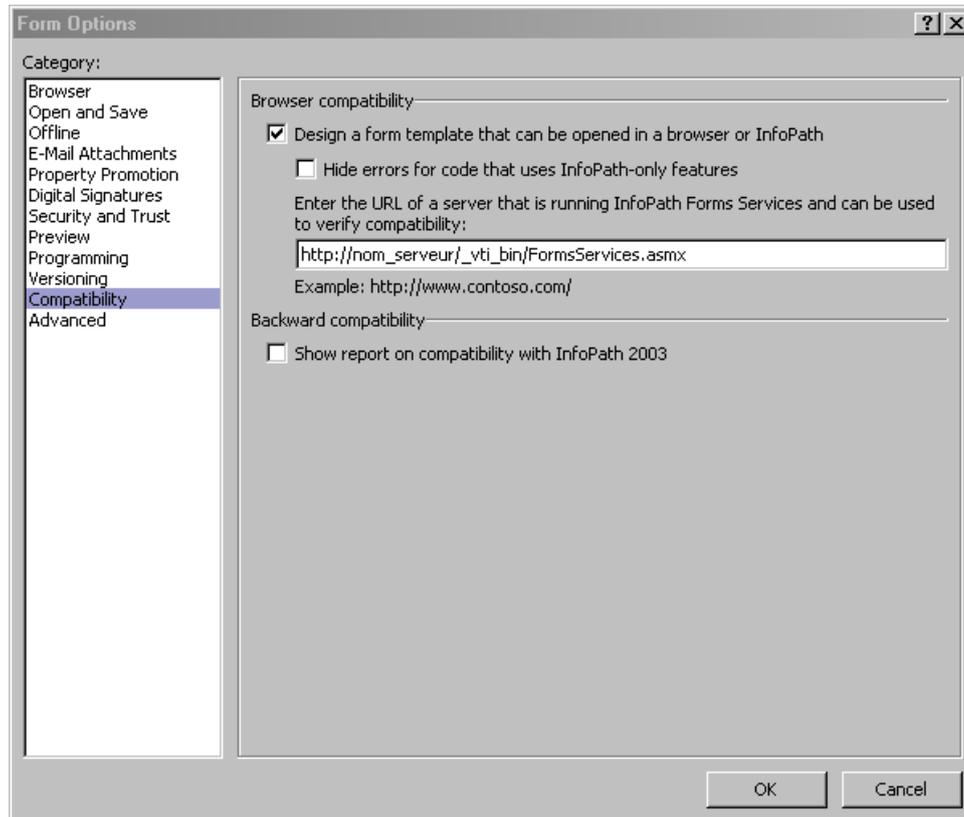
1. Soit on doit créer un nouveau formulaire et dès lors c'est très simple. Il suffit pour éviter des problèmes de cocher *Enable browser-compatible features only* lors de la création du nouveau formulaire:



Vincent ISOZ

et dès lors, lorsque vous cliquerez sur *OK*, tout ce qui ne sera pas compatible avec InfoPath Services sera désactivé.

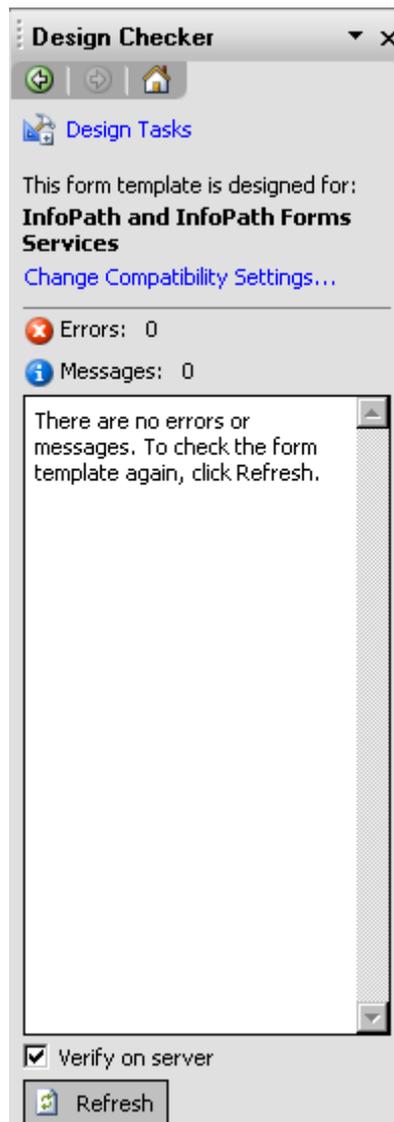
2. Vous avez déjà un formulaire et vous voulez contrôler ce qui ne fonctionnera pas avec InfoPath Services. Dès lors, soit vous faites semblant de le publier et à la fin de la publication vous aurez un message d'erreur qui vous affichera la liste des problèmes (c'est la technique la plus utilisée), soit vous allez dans le menu *Tools/Form options...* et ensuite dans la catégorie *Compatibility*:



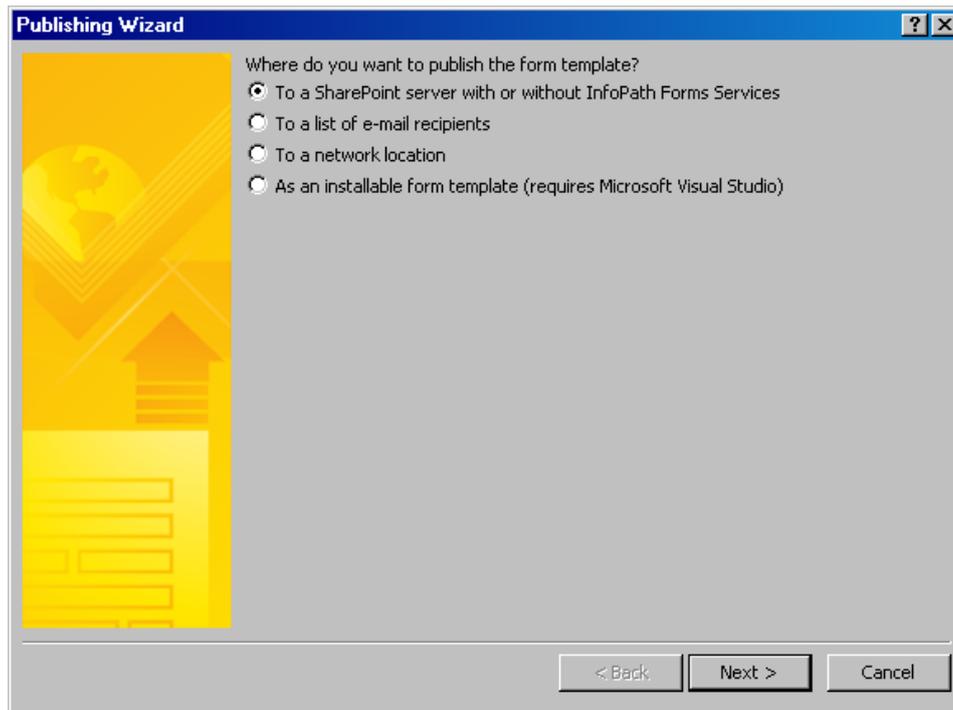
où il vous faudra saisir l'adresse du service InfoPath sous la forme:

`http://<nom du serveur>/_vti_bin/FormsServices.aspx`

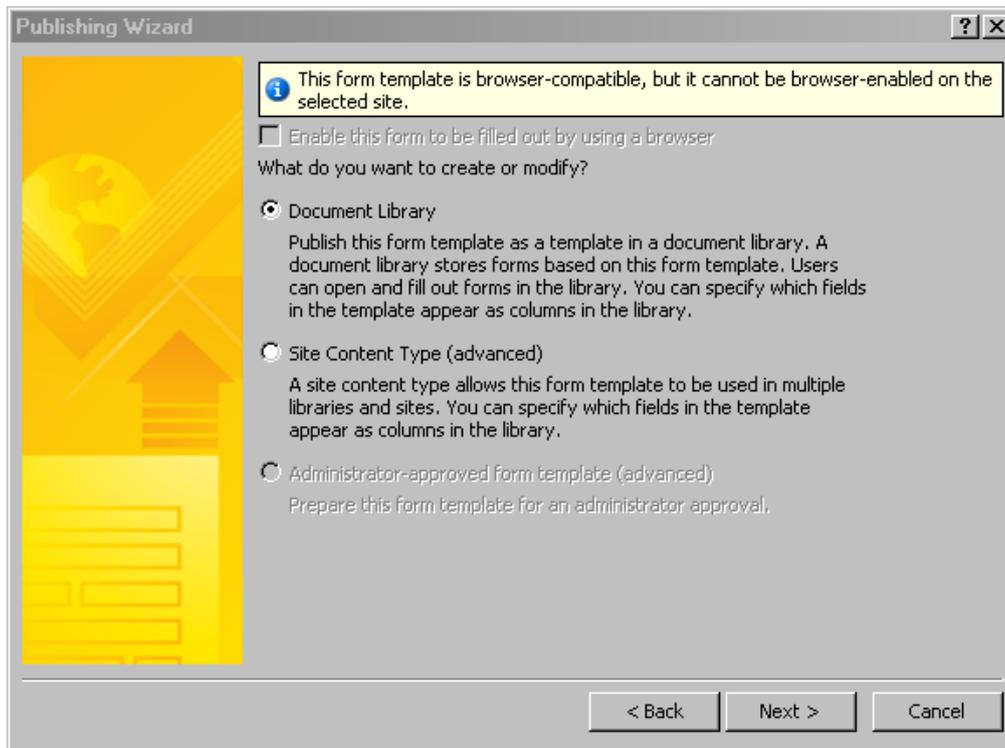
Une fois ceci fait il est possible de lancer le *Design Checker* en passant par exemple via le volet des tâches d'InfoPath:



Si vous essayez de publier l'assistant suivant démarrera:



et si vous cliquez sur *Next* et que le serveur n'est pas configuré correctement, vous aurez le message *This form template is browser-compatible, but it cannot be browser-enabled on the selected site*:



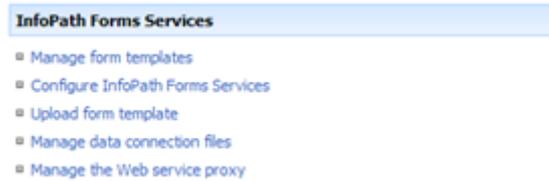
Il faut alors arrêter l'assistant et dans un premier temps, sur le site SharePoint 2007/2010 aller dans *Site Actions/Site settings/Site features* et il faut d'abord activer la *feature* suivante:



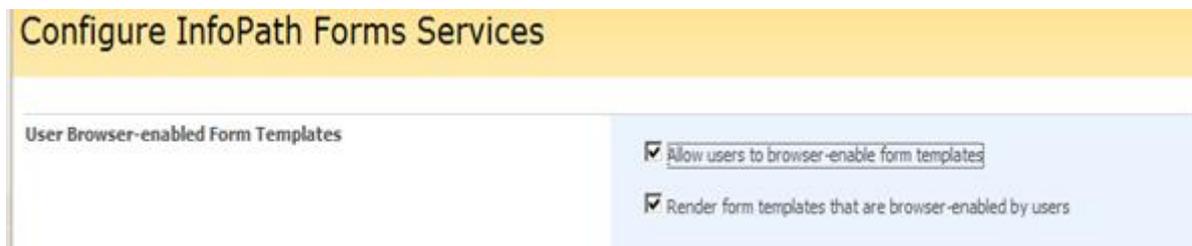
Vincent ISOZ

et réessayer la publication.

Si cela ne marche toujours pas il faut aller alors sur le serveur SharePoint 2007/2010 dans la *Central Administration/Application Management* et s'assurer que dans la partie:



en allant dans *Configure InfoPath Forms Services* les deux cases supérieures suivantes soient bien cochées:



et réessayer la publication.

Si cela ne marche toujours pas... il faut sur le serveur SharePoint 2007/2010 taper les lignes de commandes suivantes dans l'ordre indiqué:

```
C:\WINDOWS\system32\cmd.exe

C:\Program Files\Fichiers communs\Microsoft Shared\web server extensions\12\BIN>
stsadm -o deactivatefeature -filename IPFSSiteFeatures\feature.xml -force -url http://sps.excelcia.ch/customer/demo
Feature 'c88c4ff1-dbf5-4649-ad9f-c6c426ebcbf5' is not activated at this scope.

C:\Program Files\Fichiers communs\Microsoft Shared\web server extensions\12\BIN>
stsadm -o deactivatefeature -filename IPFSWebFeatures\feature.xml -force -url http://sps.excelcia.ch/customer/demo
Feature 'a0e5a010-1329-49d4-9e09-f280cdbed37d' is not activated at this scope.

C:\Program Files\Fichiers communs\Microsoft Shared\web server extensions\12\BIN>
stsadm -o activatefeature -filename IPFSSiteFeatures\feature.xml -force -url http://sps.excelcia.ch/customer/demo
Operation completed successfully.

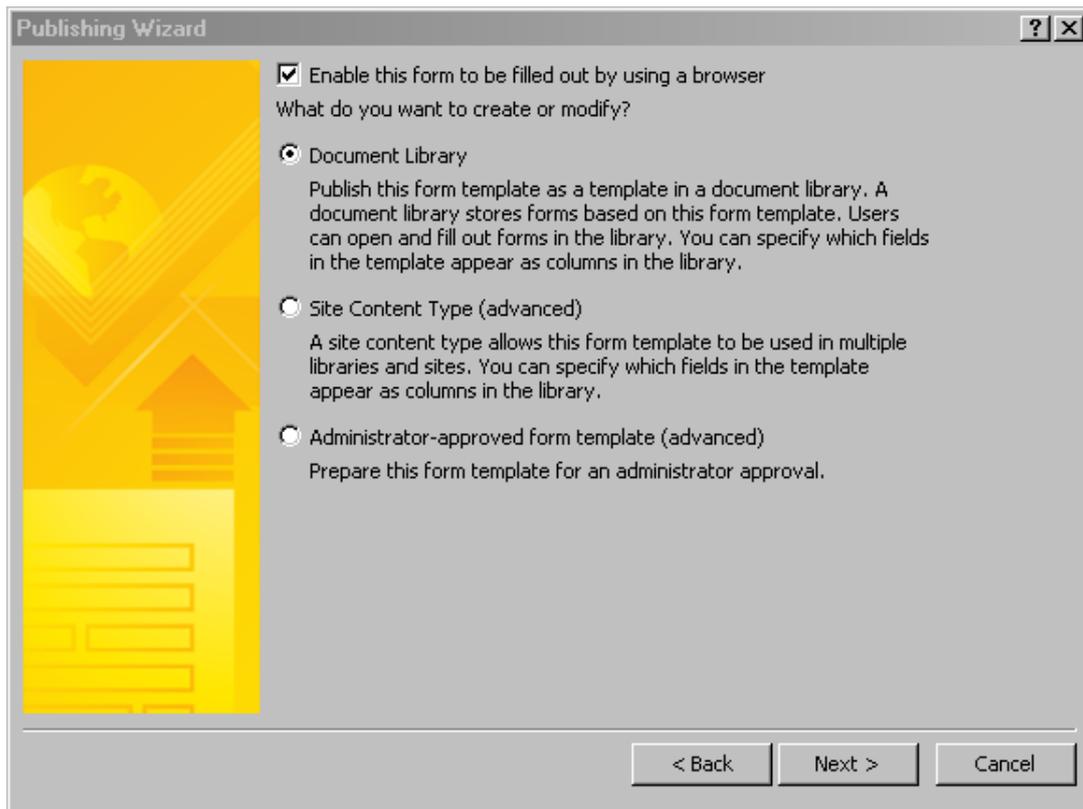
C:\Program Files\Fichiers communs\Microsoft Shared\web server extensions\12\BIN>
stsadm -o activatefeature -filename IPFSWebFeatures\feature.xml -force -url http://sps.excelcia.ch/customer/demo
Operation completed successfully.

C:\Program Files\Fichiers communs\Microsoft Shared\web server extensions\12\BIN>
```

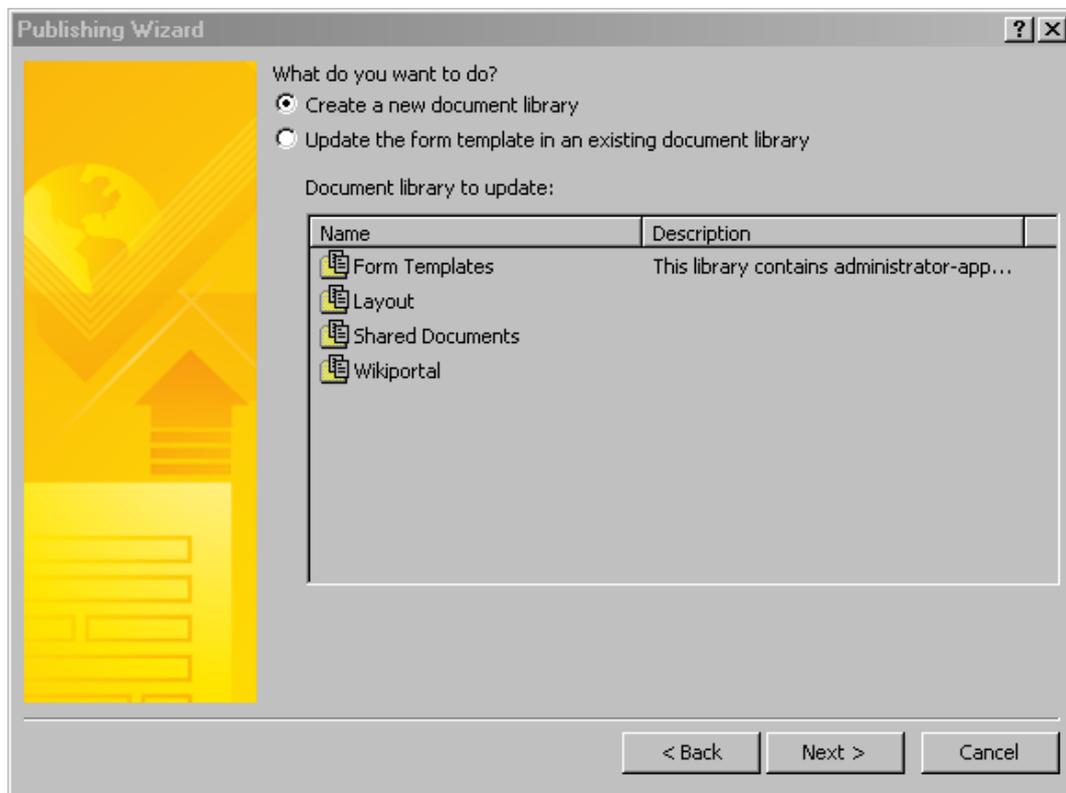
et là... cela DOIT marcher à tous les coups (excepté bien évidemment si vous utilisez dans votre formulaire un élément nom compatible avec InfoPath Services)!

Vincent ISOZ

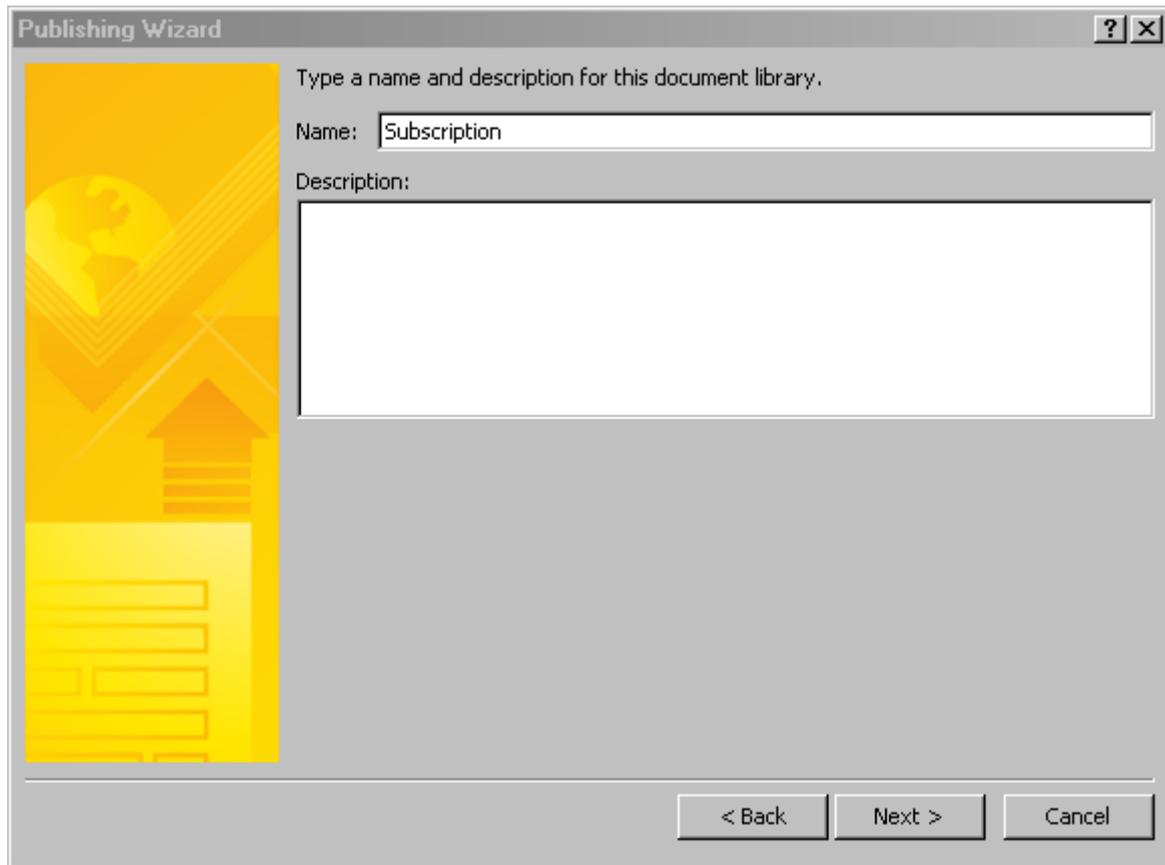
Vous aurez alors dans l'assistant:



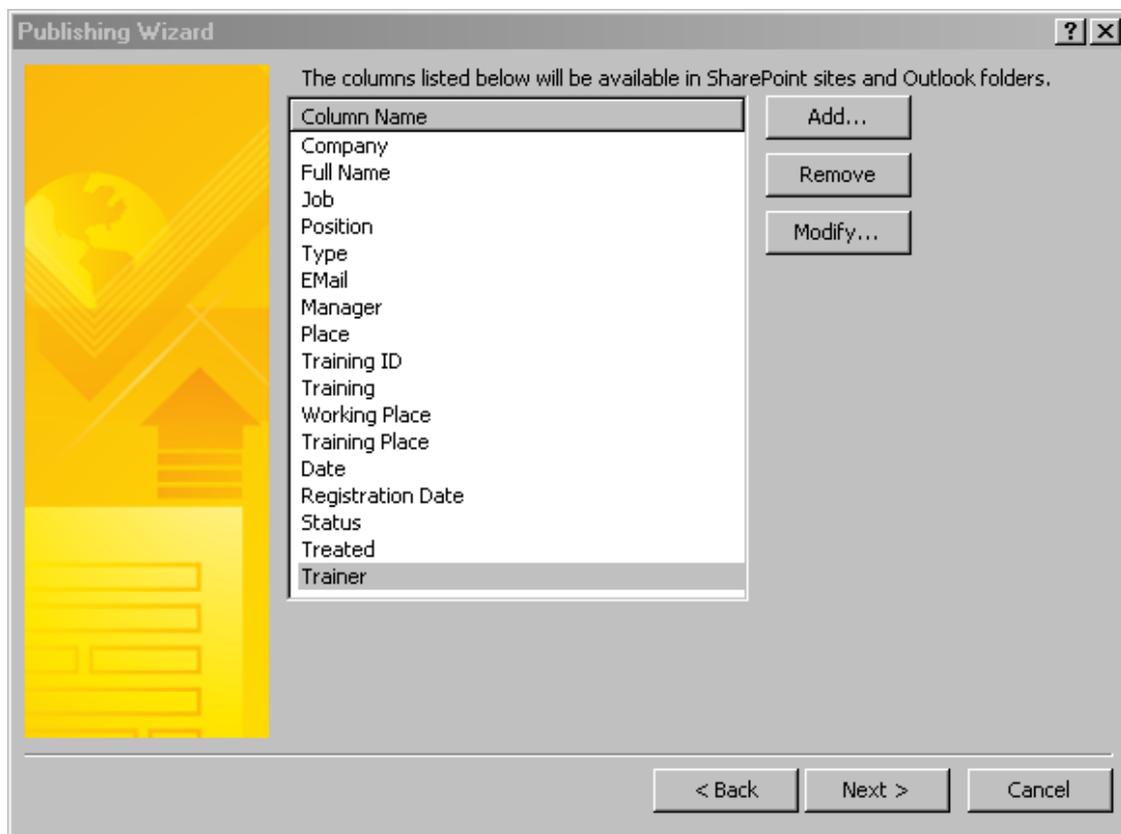
Ensuite:



Ensuite:

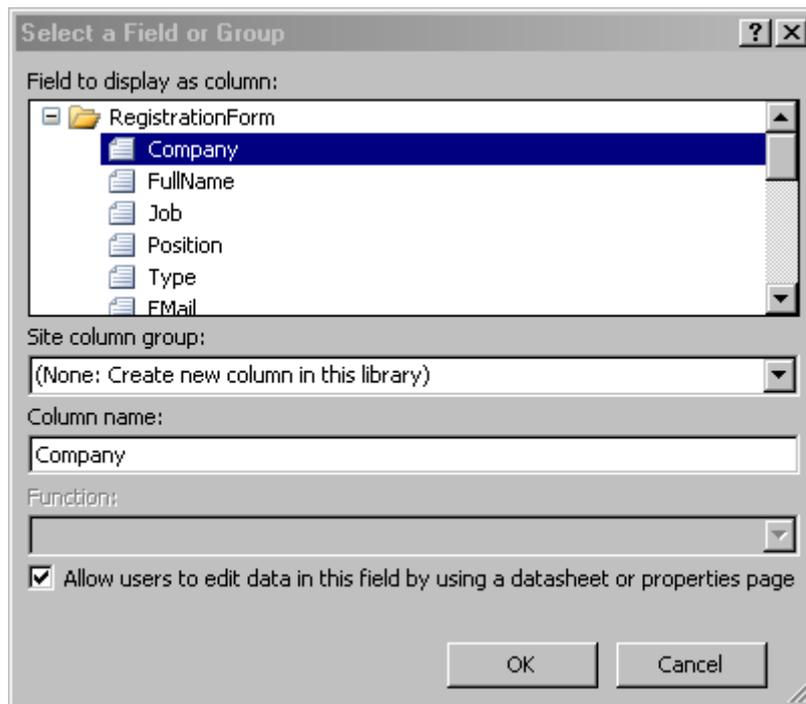


Ensuite:

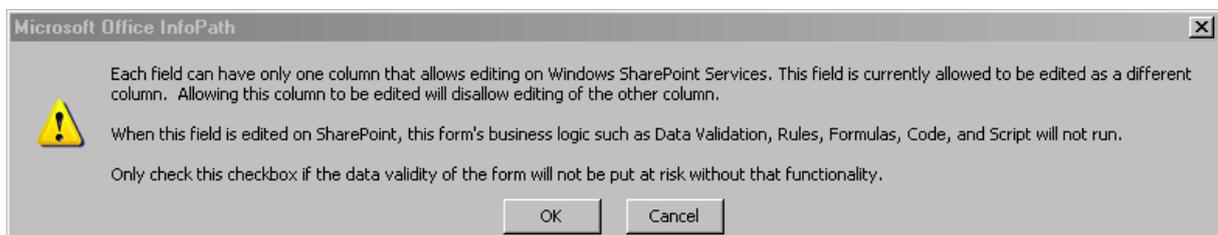


Vincent ISOZ

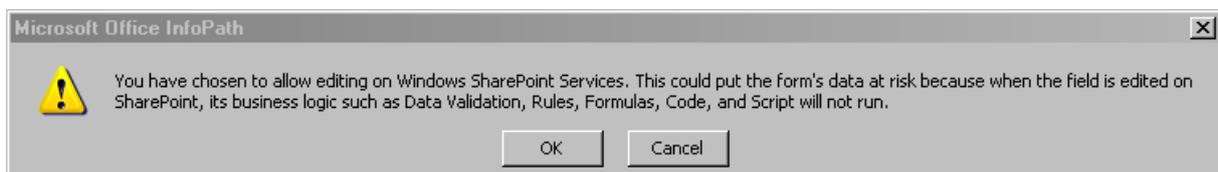
avec la possibilité de choisir en allant dans *Modify* quel champ sera éditable ou dans directement dans SharePoint sans passer par le formulaire InfoPath:



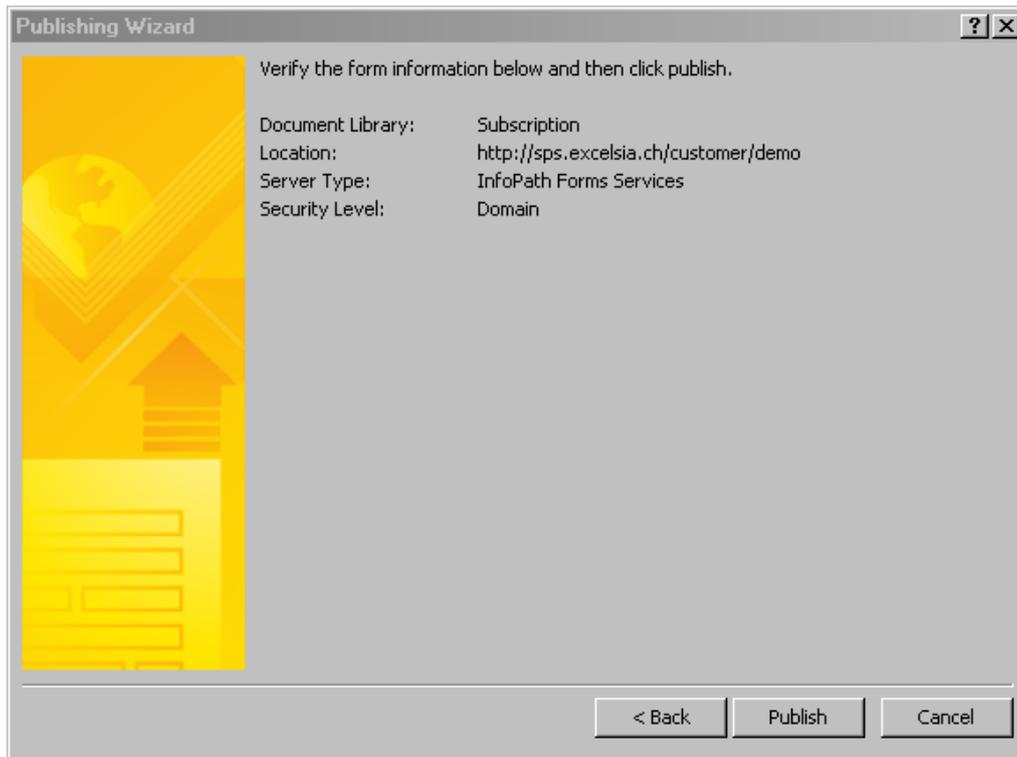
et lorsqu'on valide *Allow users to edit data in this field by using a datasheet or properties page* vient le message d'avertissement suivant:



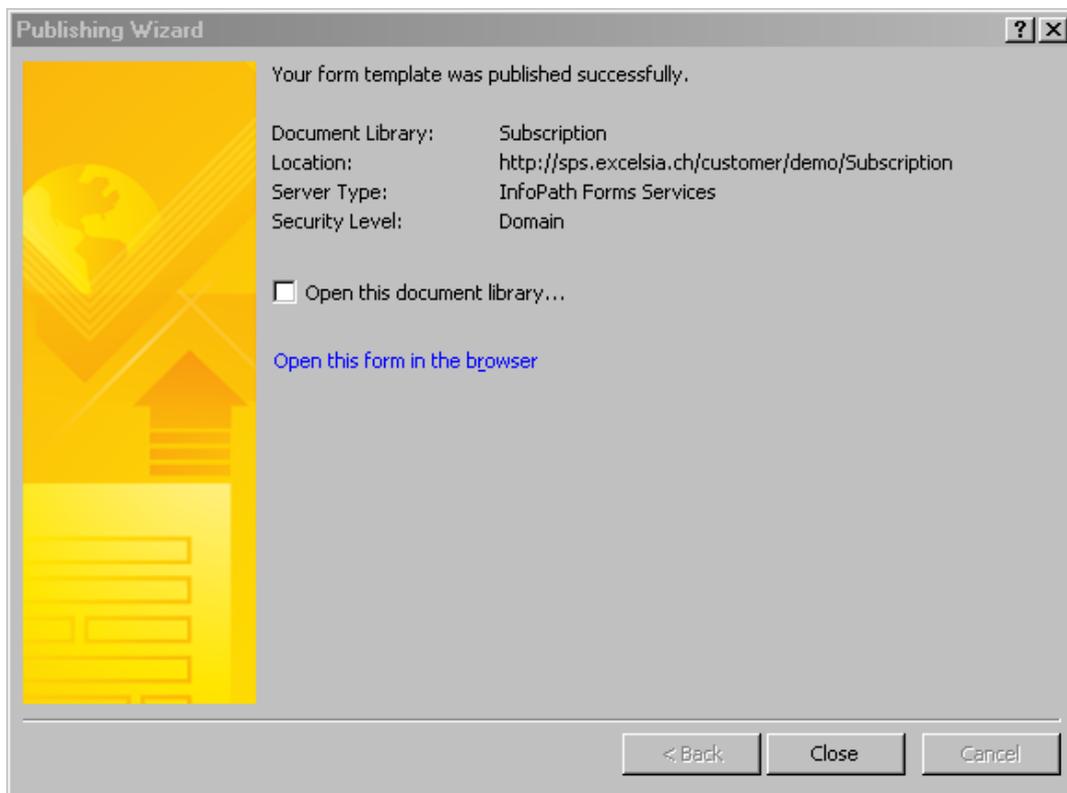
et quand on valide le tout:



En faisant *Next*:



et en validant la publication il vient enfin:



Et c'est fini....

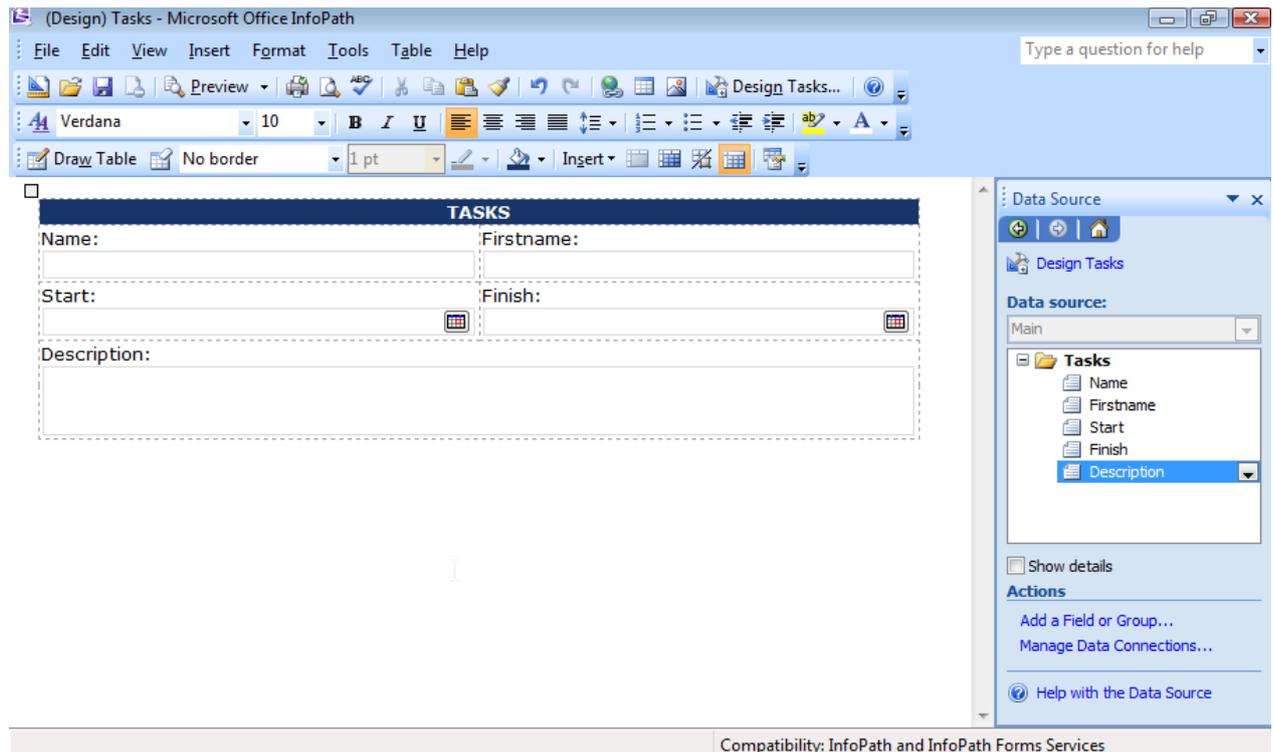
Vincent ISOZ

## 11.5 Boîte de message à la soumission dans InfoPath Services (v2007)

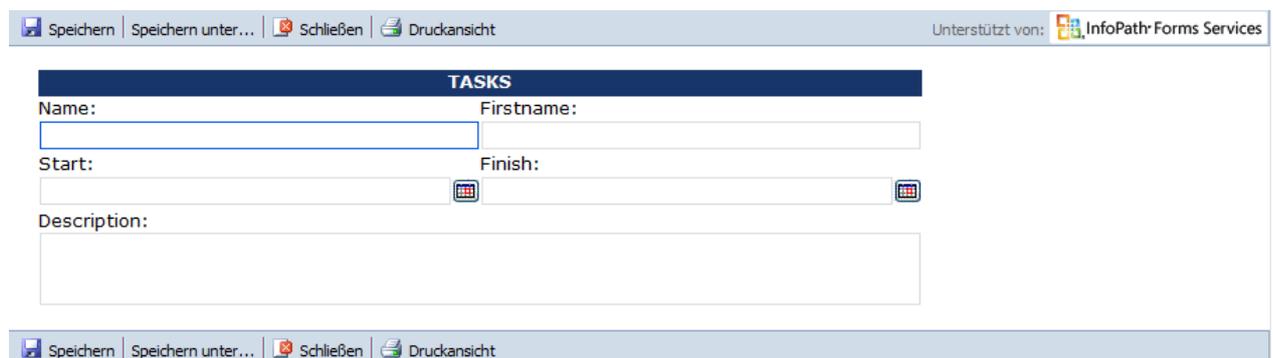
Il arrive (très) régulièrement dans les entreprises que des employés qui doivent remplir des formulaires, oublient leurs obligations après avoir soumis celui-ci. Une bonne manière de corriger un peu le tir est d'afficher une boîte de message à la soumission.

Bien que la création et l'affichage d'un message avec le client lourd InfoPath soit simple, le faire avec InfoPath Services 2007 est moins évident.

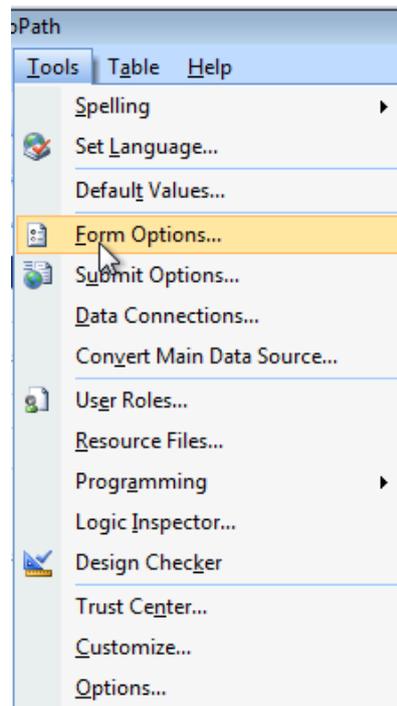
Pour l'exemple, nous allons considérer le petit formulaire suivant:



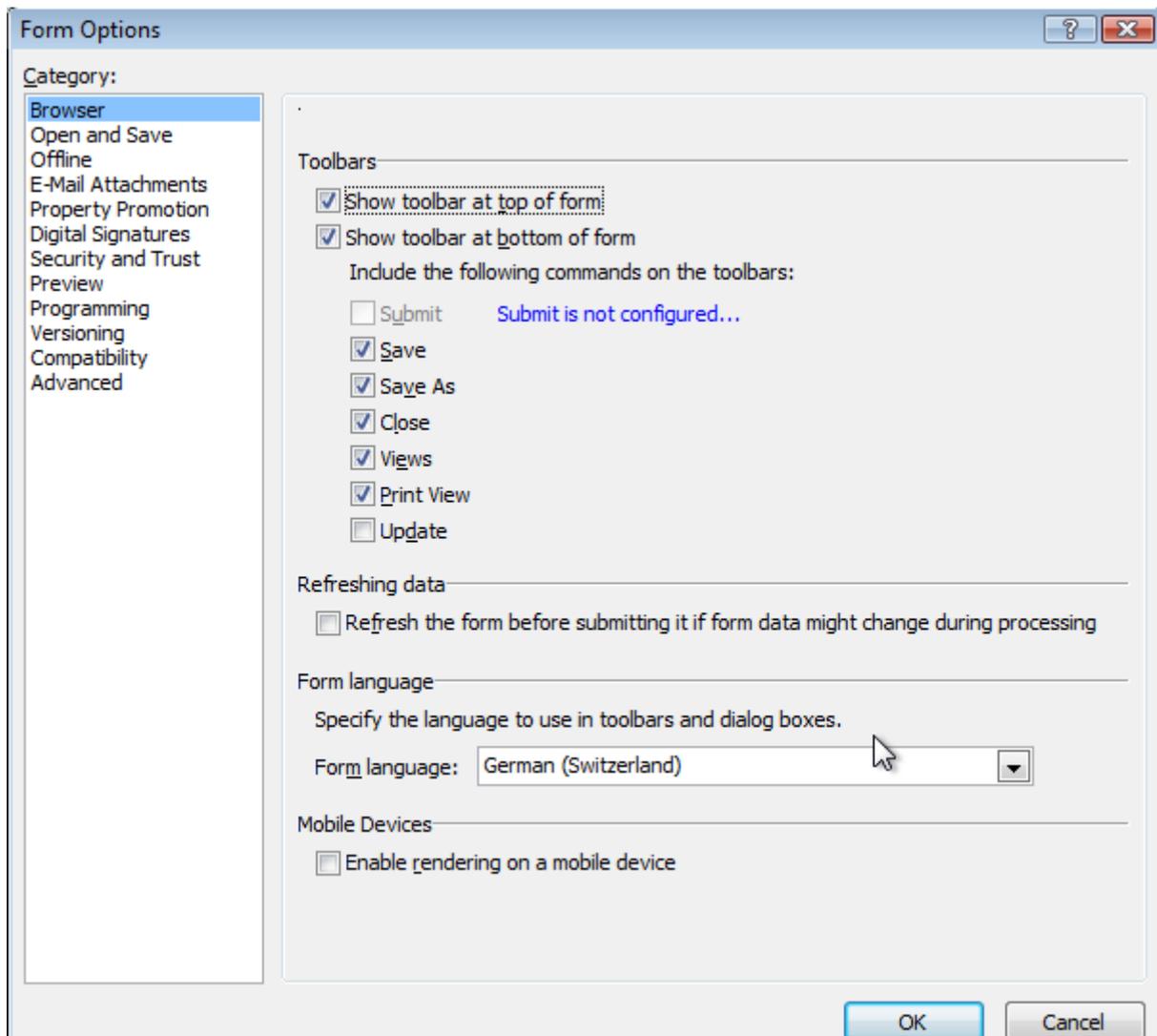
qui sous InfoPath Services ressemblera donc après publication à:



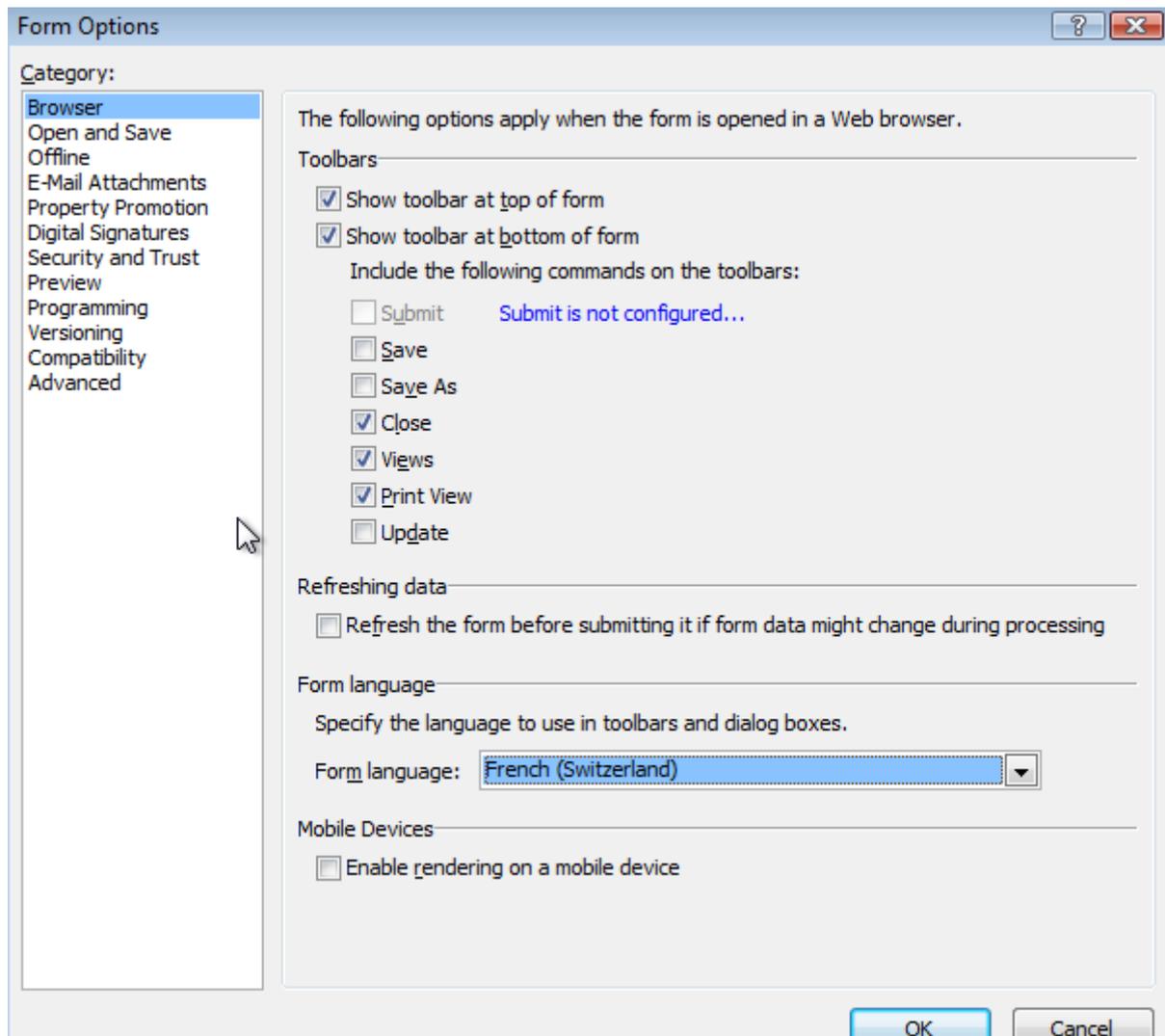
Nous souhaitons déjà changer les langues de InfoPath et empêcher les utilisateur de pouvoir enregistrer en utilisant les boutons des barres qui ne nous permettent pas sans code complexe d'afficher une message d'avertissement. Pour cela, dans InfoPath, nous allons dans le menu **Tools/Form Options**:



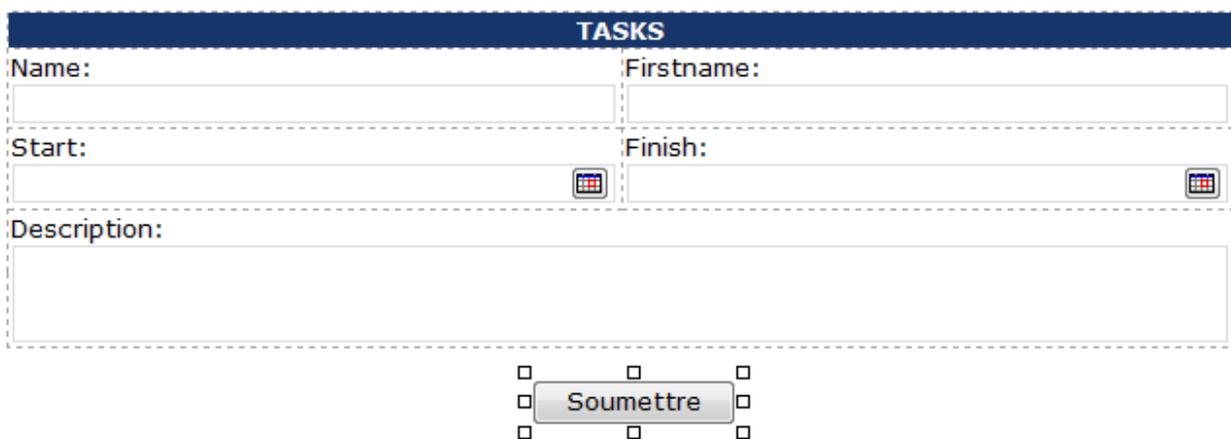
et dans la partie Browser nous décochons **Save** et **Save As** et changeons la **Form Language** à *Français*:



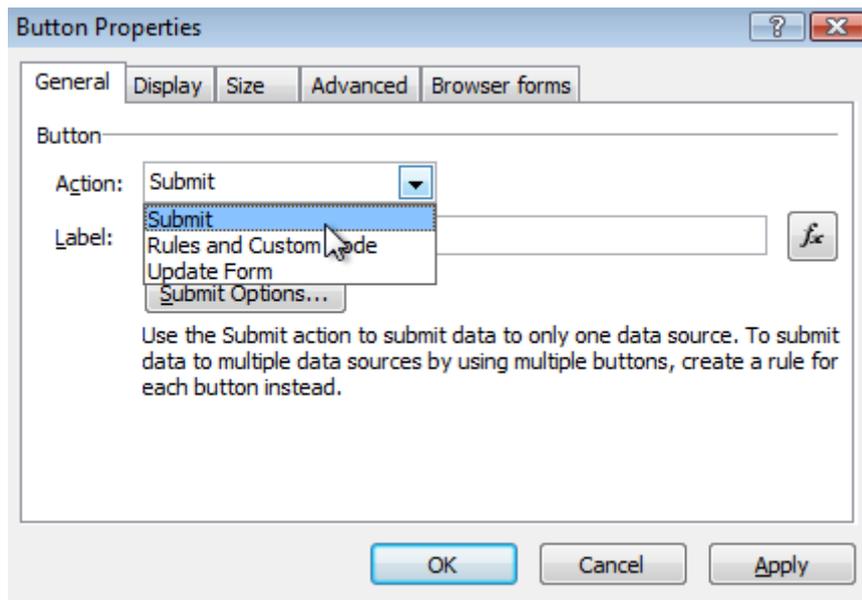
Pour avoir:



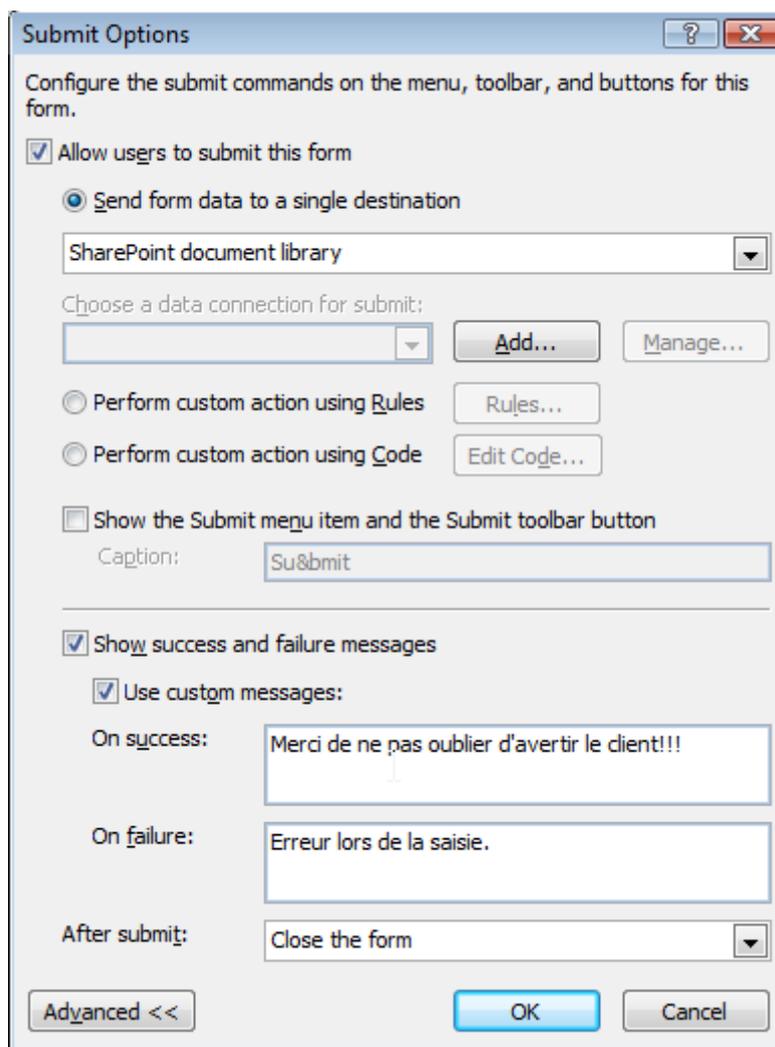
ensuite, nous créons un simple et bête bouton *Soumettre* comme ci-dessous:



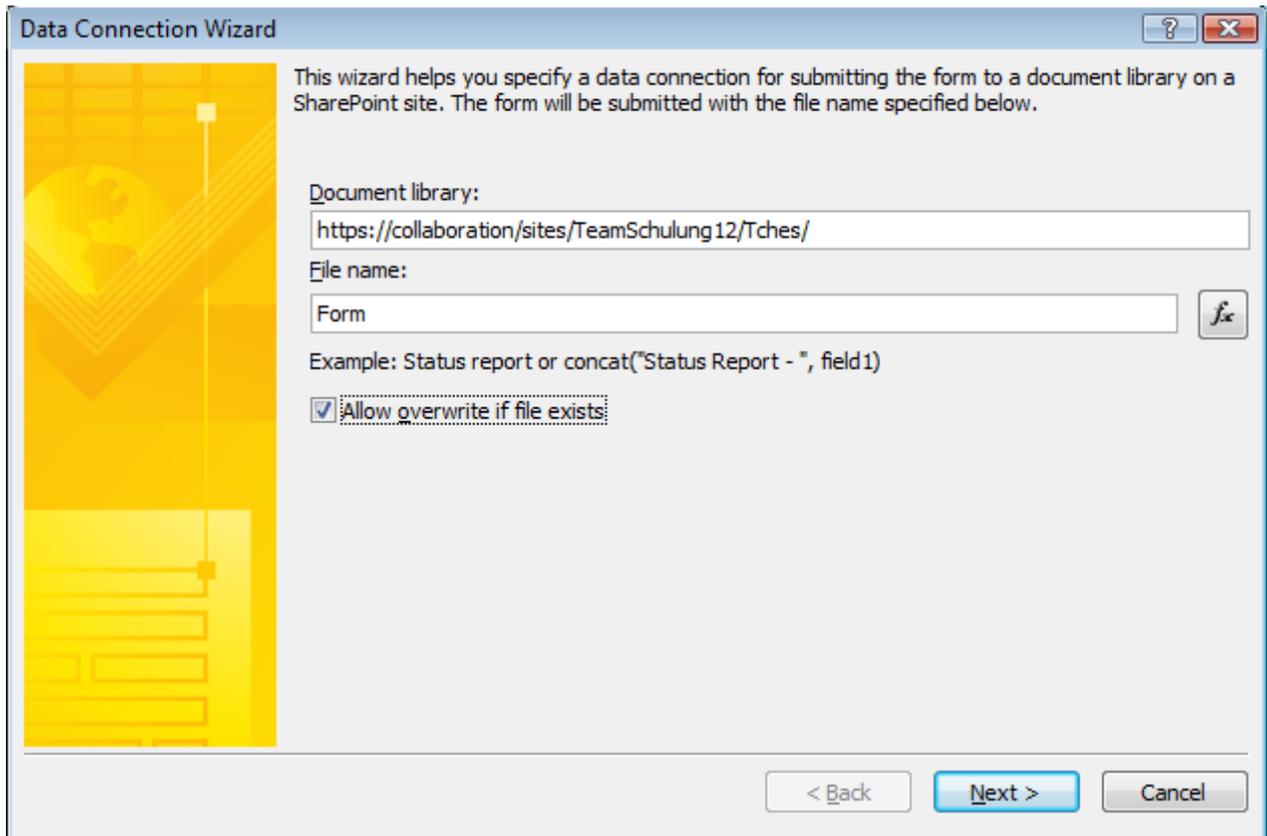
on va dans les propriétés du bouton et dans **Action** on choisi l'option **Submit**:



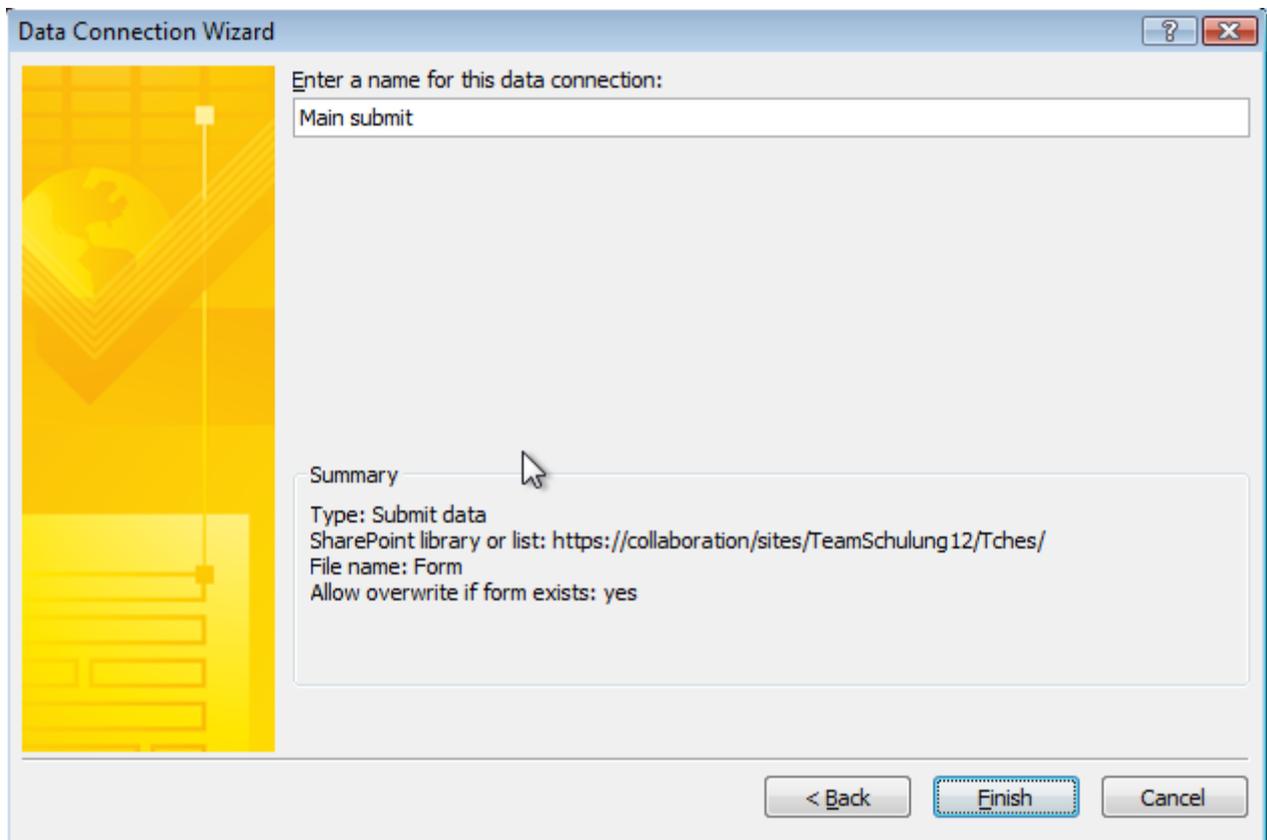
et on clique ensuite sur **Submit Options...** Viens la boîte suivante qu'il s'agit de paramétrer comme ci-dessous:



et nous cliquons sur **Add...**:



pour le **File Name** nous laissons la valeur standard car nous savons déjà faire cela proprement en y mettant des formules. Et nous cliquons sur **Next...**:



Vincent ISOZ

et sur **Finish** et validons ensuite le tout en cliquant deux fois sur **OK**. Nous publions une mise à jour du formulaire comme nous savons bien le faire et obtenons alors:

The screenshot shows the 'TASKS' form with the following fields:

- Name: [Empty text box]
- Firstname: [Empty text box]
- Start: [Empty date picker]
- Finish: [Empty date picker]
- Description: [Empty text area]

A 'Submit' button is located at the bottom center of the form.

et si nous cliquons sur **Submit**:

The screenshot shows the 'TASKS' form with the following fields:

- Name: Isoz
- Firstname: [Empty text box]
- Start: 23.02.2012
- Description: toto

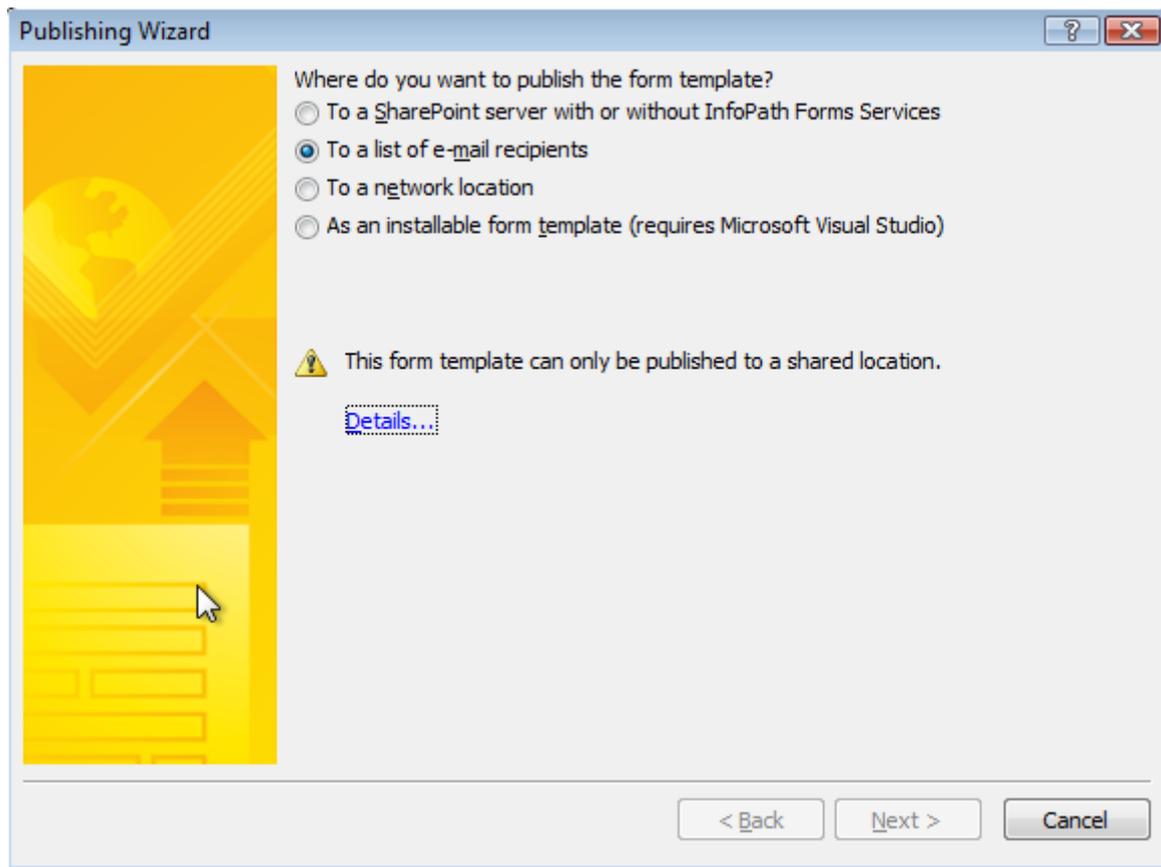
A warning dialog box is overlaid on the form, titled 'Windows Internet Explorer'. It contains a warning icon and the text: 'Merci de ne pas oublier d'avertir le client!'. An 'OK' button is visible in the dialog box.

et le formulaire se ferme automatiquement aussi:

*Le formulaire a été fermé.*

bon libre à vous ensuite de lancer des flux SharePoint...

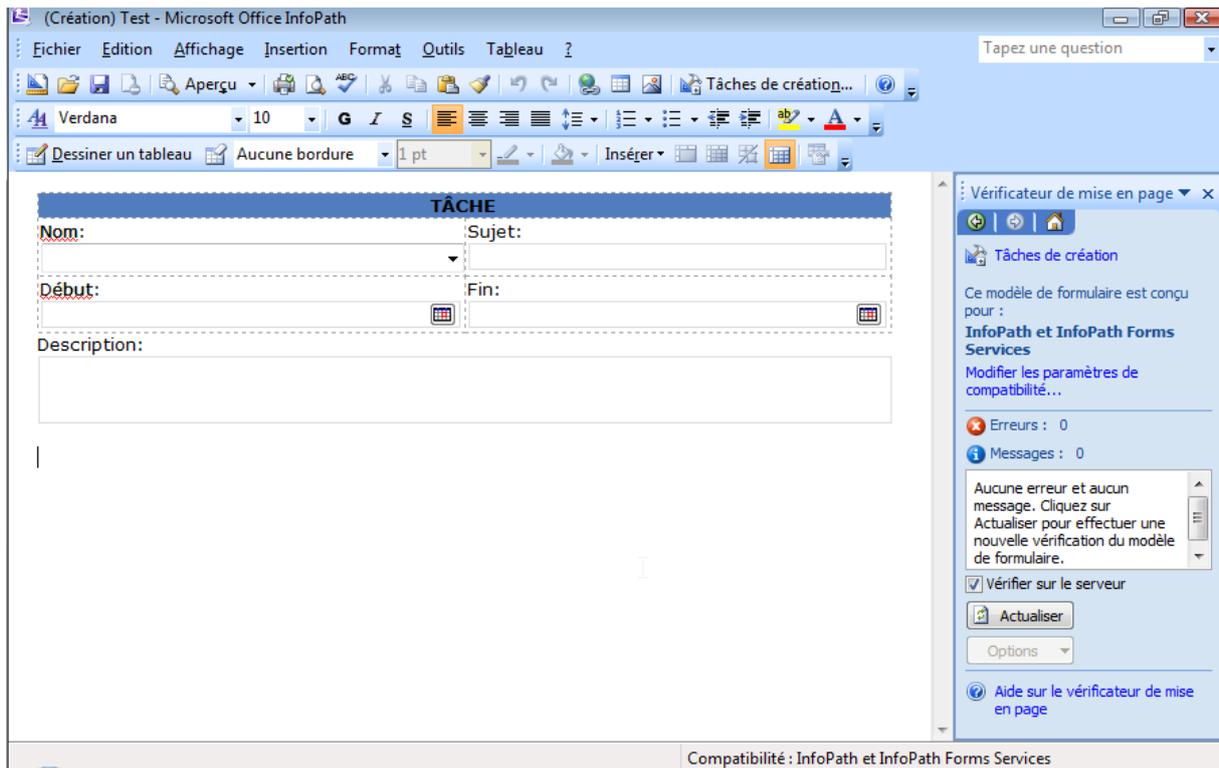
Beaucoup d'utilisateurs souhaitent pouvoir envoyer ce type de formulaire directement par Outlook car le bouton Submit permettrait éventuellement de gagner du temps. Malheureusement, si vous lancez l'assistant d'envoi (déjà vu plus haut à maintes reprises) vous aurez dans InfoPath:



## ***11.6 Mise à jour liste SharePoint suite à modification de structure d'un formulaire InfoPath***

### **11.6.1 InfoPath/SharePoint 2007**

Partons du petit formulaire suivant pour faire simple:



et publions-le avec tout ces champs dans une nouvelle bibliothèque.

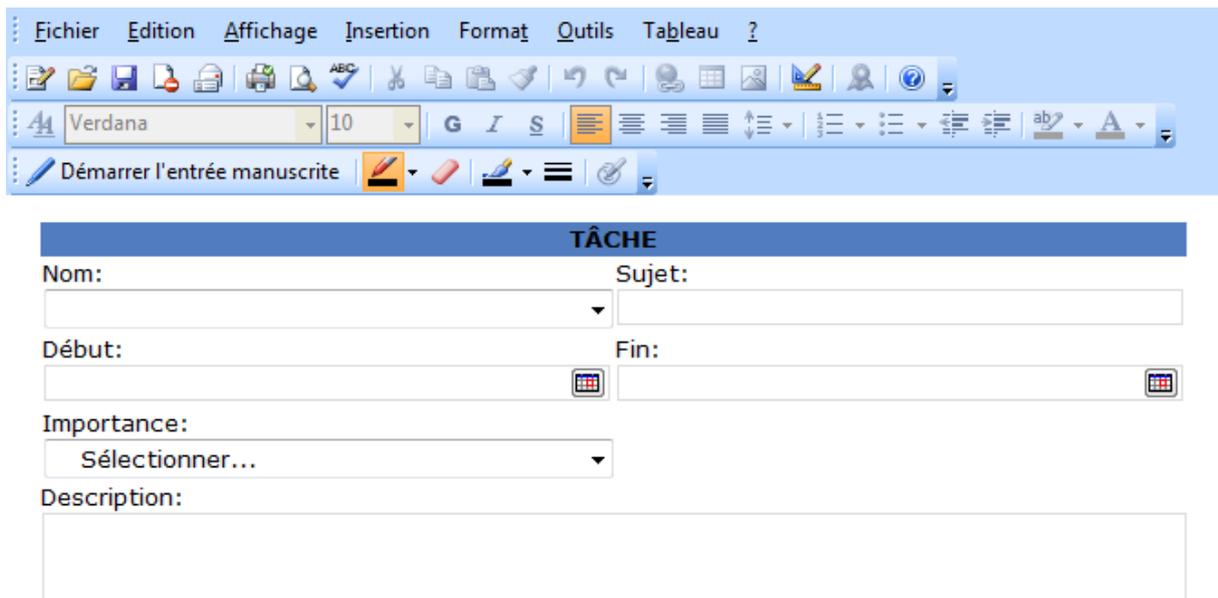
Remplissons quelques formulaires (au moins trois pour l'exemple). Cela donne naturellement:

Tasks						
Nouveau		Télécharger		Actions		Paramètres
						Afficher : <b>Tous les documents</b>
Type	Nom	Utilisateurs	Sujet	Debut	Fin	
	Formulaire 1 Nouveau!	DOMOSS\schulung01	Tache 1	15.03.2011	30.08.2011	
	Formulaire 2 Nouveau!	DOMOSS\schulung11	Tache 2	13.07.2010	24.09.2010	
	Formulaire 3 Nouveau!	DOMOSS\schulung03	Tache 3	31.01.2009	24.03.2009	

Maintenant, ajoutons un champ à notre modèle de formulaire dans InfoPath et republions en mettant à jour la librairie:

Tasks						
Nouveau		Télécharger		Actions		Paramètres
						Afficher : <b>Tous les documents</b>
Type	Nom	Utilisateurs	Sujet	Debut	Fin	Importance
	Formulaire 1 Nouveau!	DOMOSS\schulung01	Tache 1	15.03.2011	30.08.2011	
	Formulaire 2 Nouveau!	DOMOSS\schulung11	Tache 2	13.07.2010	24.09.2010	
	Formulaire 3 Nouveau!	DOMOSS\schulung03	Tache 3	31.01.2009	24.03.2009	

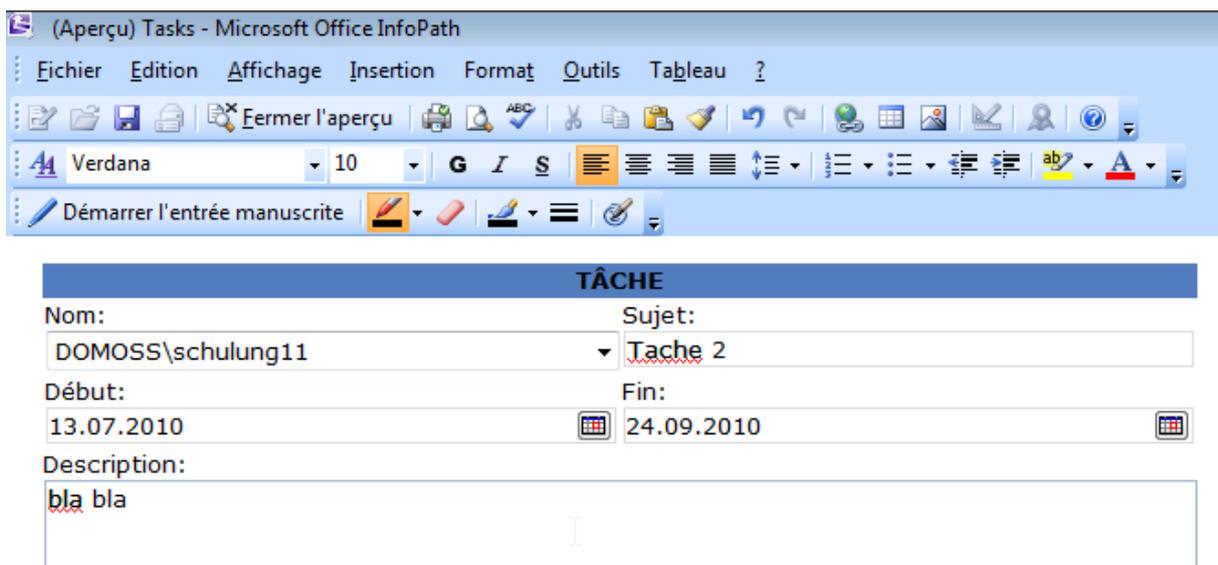
Si nous créons une nouvelle tâche dès lors nous avons:



The screenshot shows the Microsoft Office InfoPath interface. The menu bar includes 'Fichier', 'Edition', 'Affichage', 'Insertion', 'Format', 'Outils', and 'Tableau'. The ribbon shows various icons for file operations and editing. The font is set to Verdana, size 10. The task form is titled 'TÂCHE' and contains the following fields:

TÂCHE	
Nom:	Sujet:
<input type="text"/>	<input type="text"/>
Début:	Fin:
<input type="text"/>	<input type="text"/>
Importance:	
<input type="text" value="Sélectionner..."/>	
Description:	
<input type="text"/>	

Pas de problèmes c'est logique. Mais si nous éditons une des anciennes tâches. Il se peut que nous ayons alors (c'est rare que cela arrive pour des simples formulaires mais dès que c'est plus complexe c'est presque systématique):

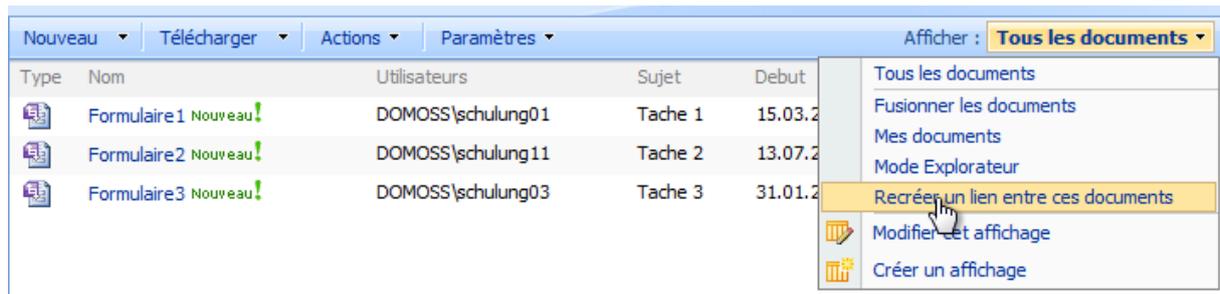


The screenshot shows the Microsoft Office InfoPath interface with the task form populated. The menu bar and ribbon are the same as in the previous screenshot. The task form is titled 'TÂCHE' and contains the following fields:

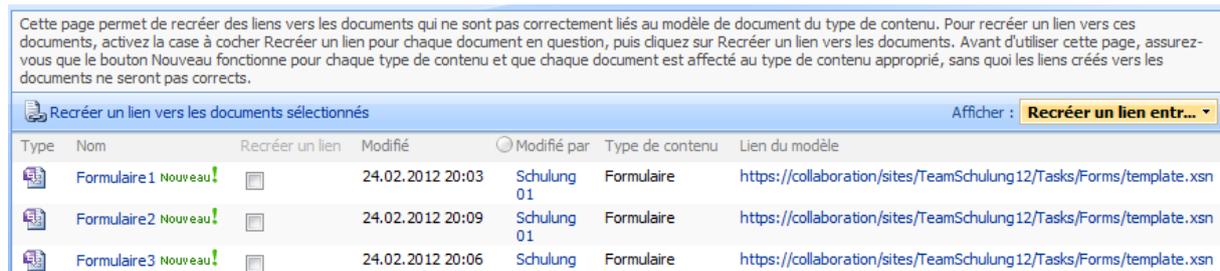
TÂCHE	
Nom:	Sujet:
<input type="text" value="DOMOSS\schulung11"/>	<input type="text" value="Tache 2"/>
Début:	Fin:
<input type="text" value="13.07.2010"/>	<input type="text" value="24.09.2010"/>
Description:	
<input type="text" value="bla bla"/>	

hummm.... il manque quelque chose...

Pour corriger cela, il vous suffit dans SharePoint d'aller cliquer sur le lien *Recréer un lien entre ces documents*:



ce qui donnera:



il suffit de cocher les formulaires de cliquer sur *Recréer un lien vers les documents sélectionnés*.

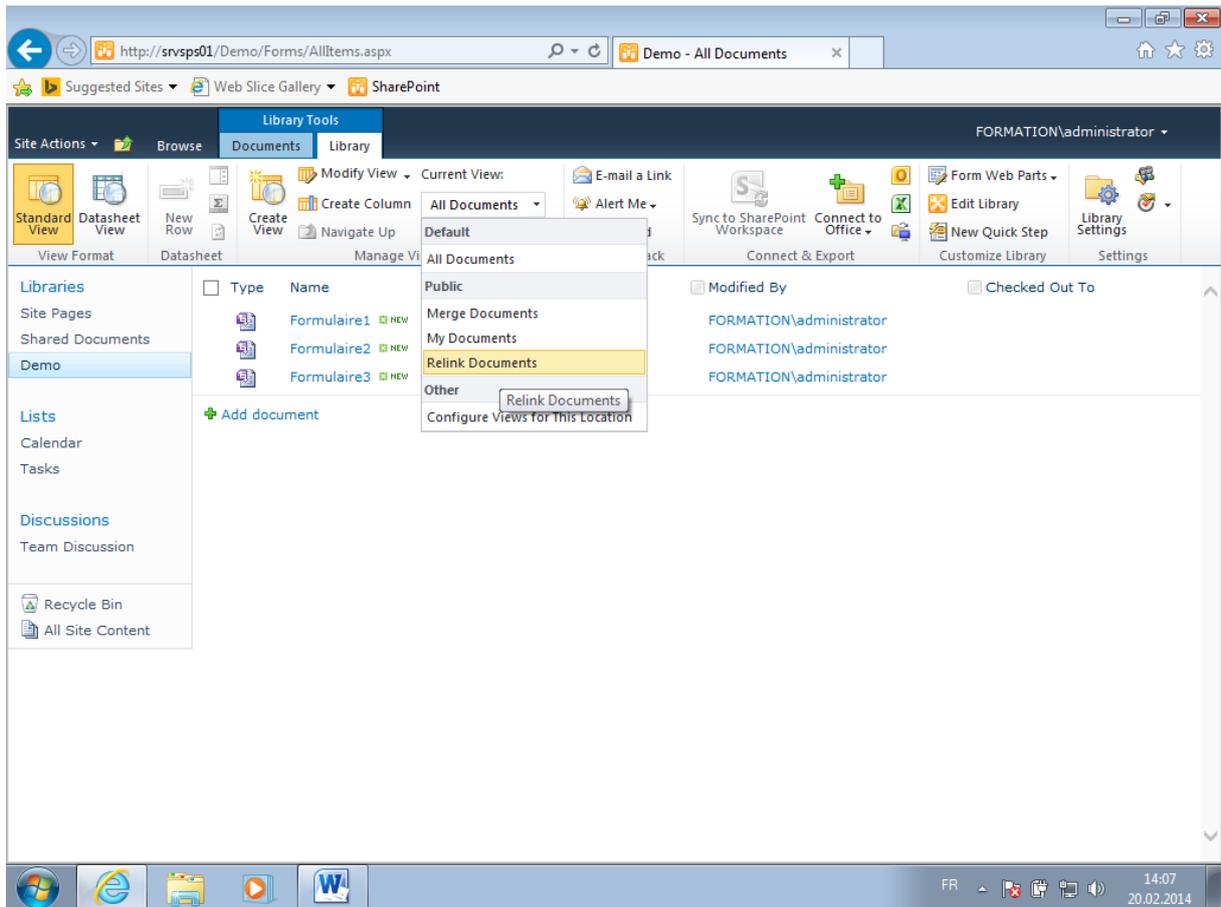
C'est simple dit comme cela mais par contre quand il y a en des milliers c'est un cauchemar. Il faut alors passer par des applications tierces du type:

<http://www.codistri.com/relink-infopath-forms/>

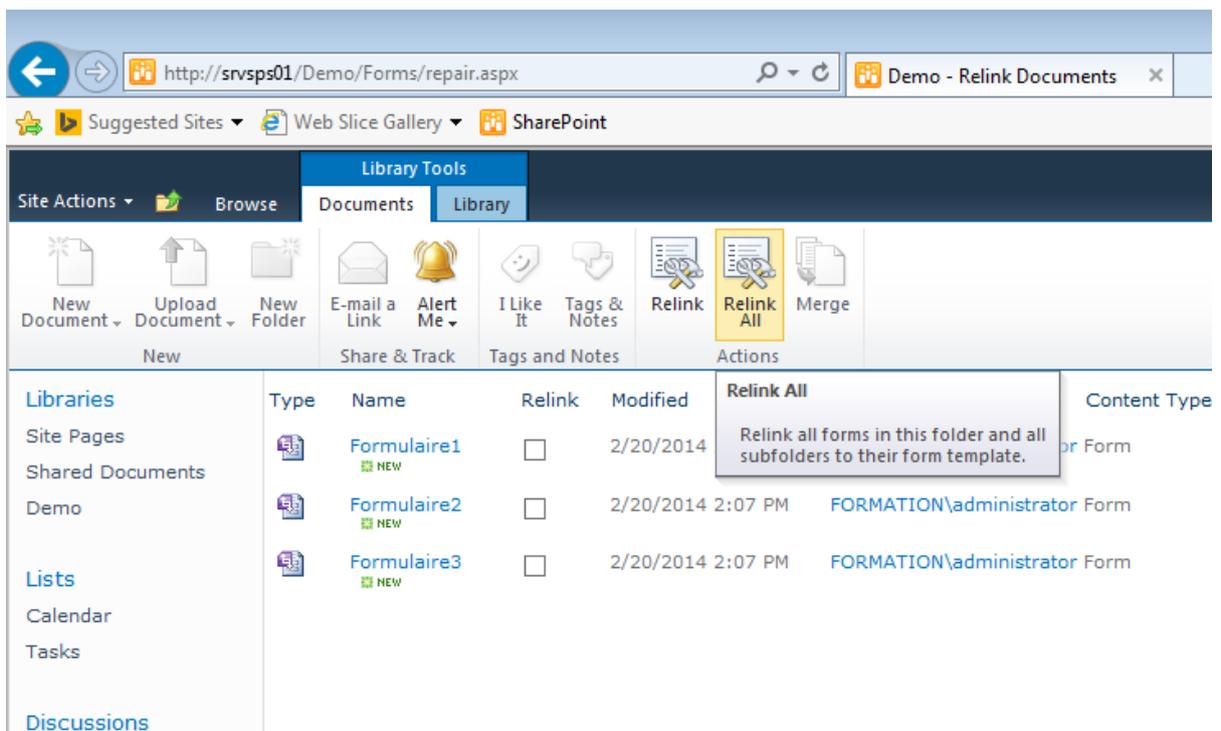
### 11.6.2 InfoPath/SharePoint 2010

Avec InfoPath/SharePoint 2010 la procédure c'est simplifiée considérablement. Voyons cela!

Le début ça ne change pas:

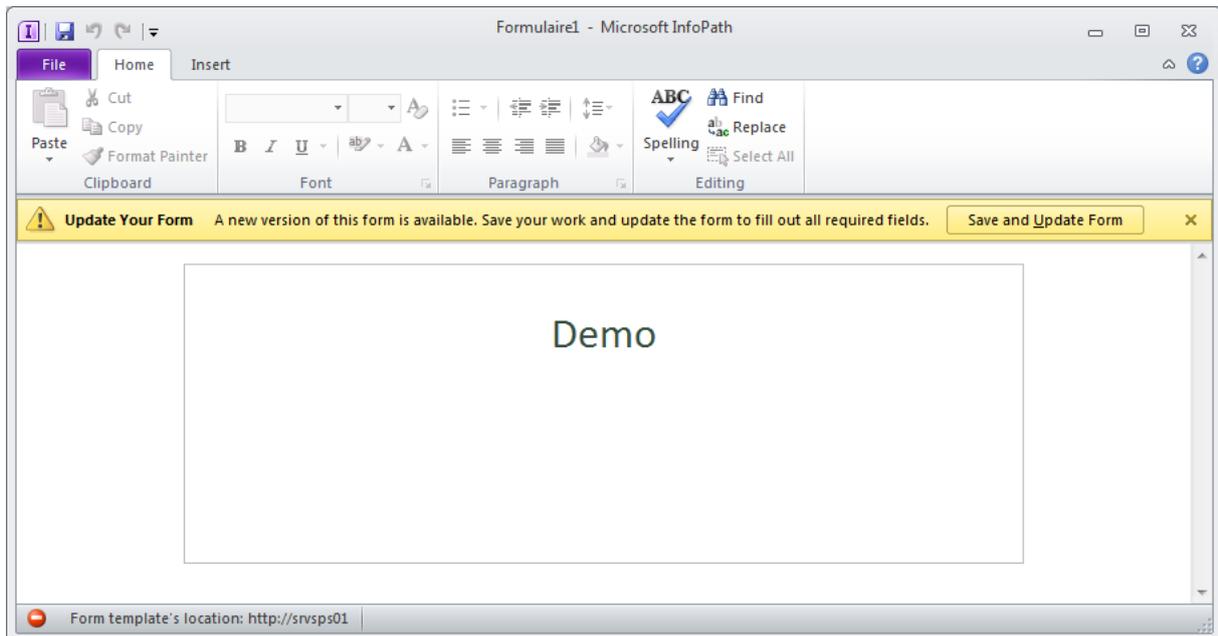


Par contre après cela change!!! Il y a un nouveau bouton qui se nomme *Relink all* qui permet de toute mettre à jour d'un coup sans avoir besoin de faire une sélection formulaire par formulaire:



Vincent ISOZ

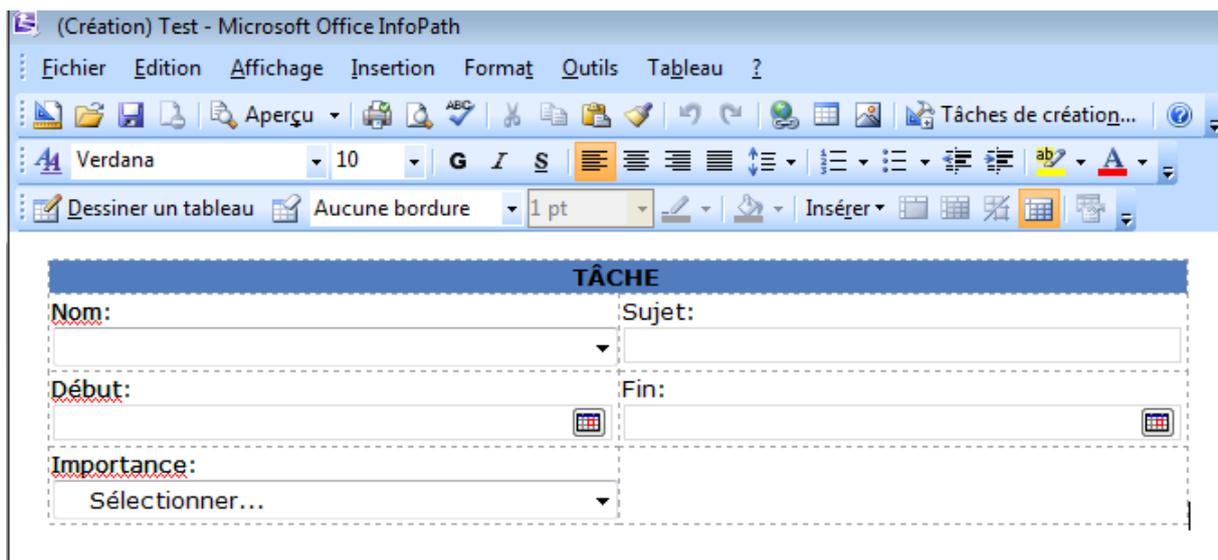
Ou encore lorsque le formulaire est mis à jour par le designer et publié, ceux qui vont ouvrir un des anciens formulaires saisis verront le message d'écran suivant pour en mettre automatiquement à jour le structure:



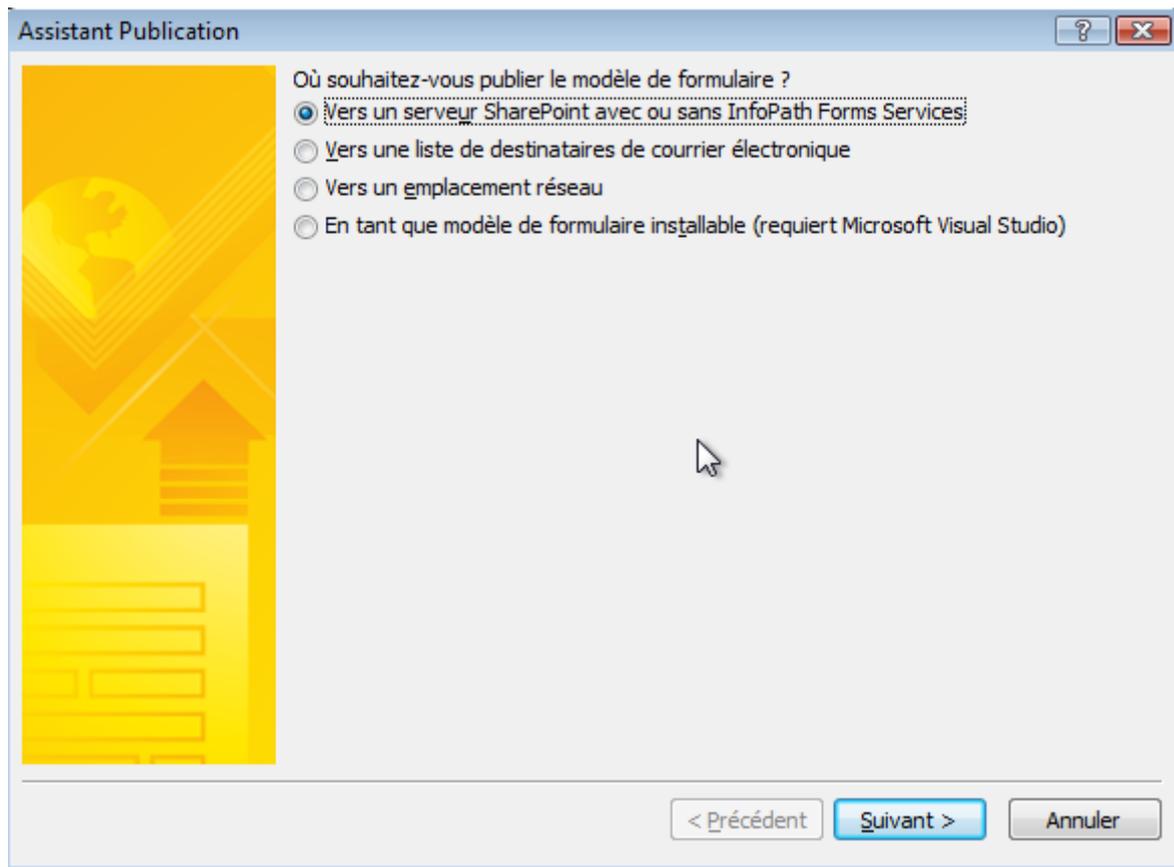
Bref... une nette amélioration quoi!

### ***11.7 Publier un formulaire en tant que Content Type (v2007)***

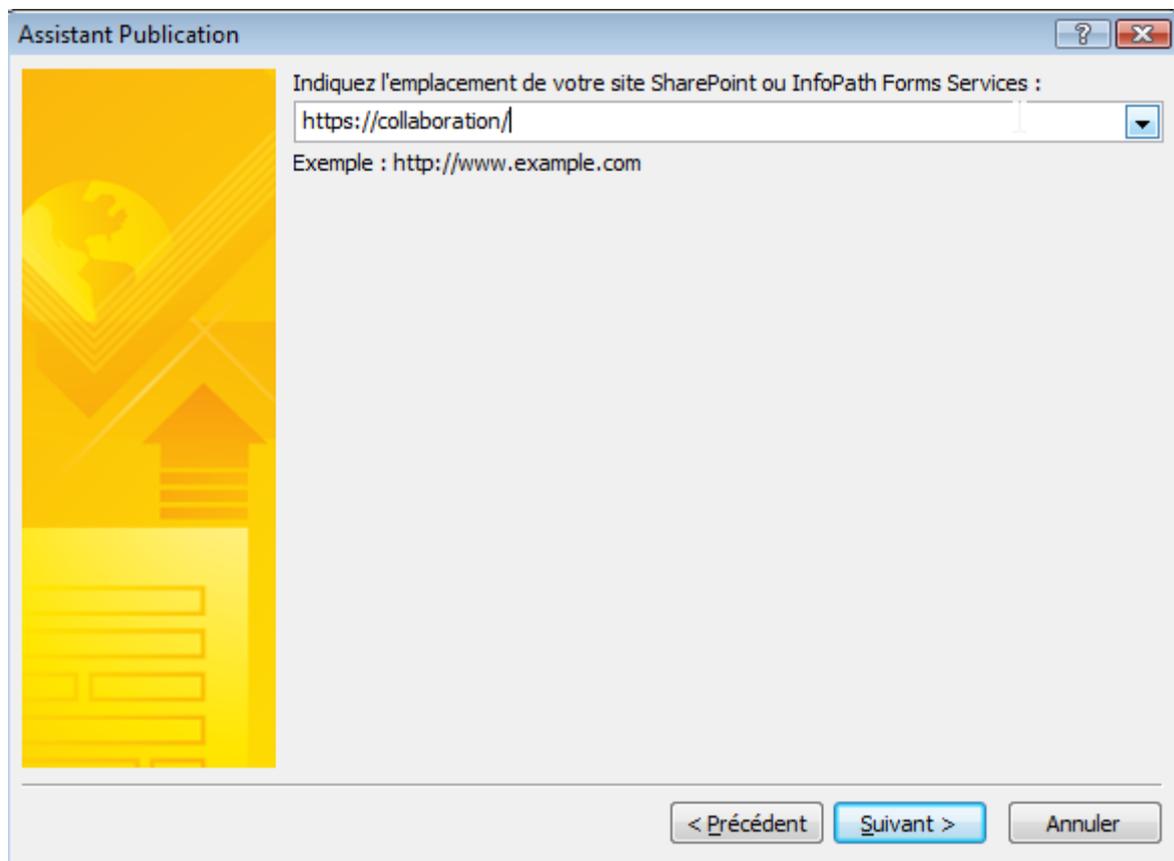
Il peut arriver lorsque l'architecture et la stratégie de gouvernance SharePoint de l'entreprise soit mal conçue qu'on ait besoin d'un même formulaire dans plusieurs bibliothèques d'une même collection de site. Pour cela il va falloir évidemment passer par les types de contenu (cf. cours SharePoint). Voyons dès lors comment procéder à partir du petit formulaire suivant:



Nous allons dans *Fichier/Publier...* comme à l'habitude:

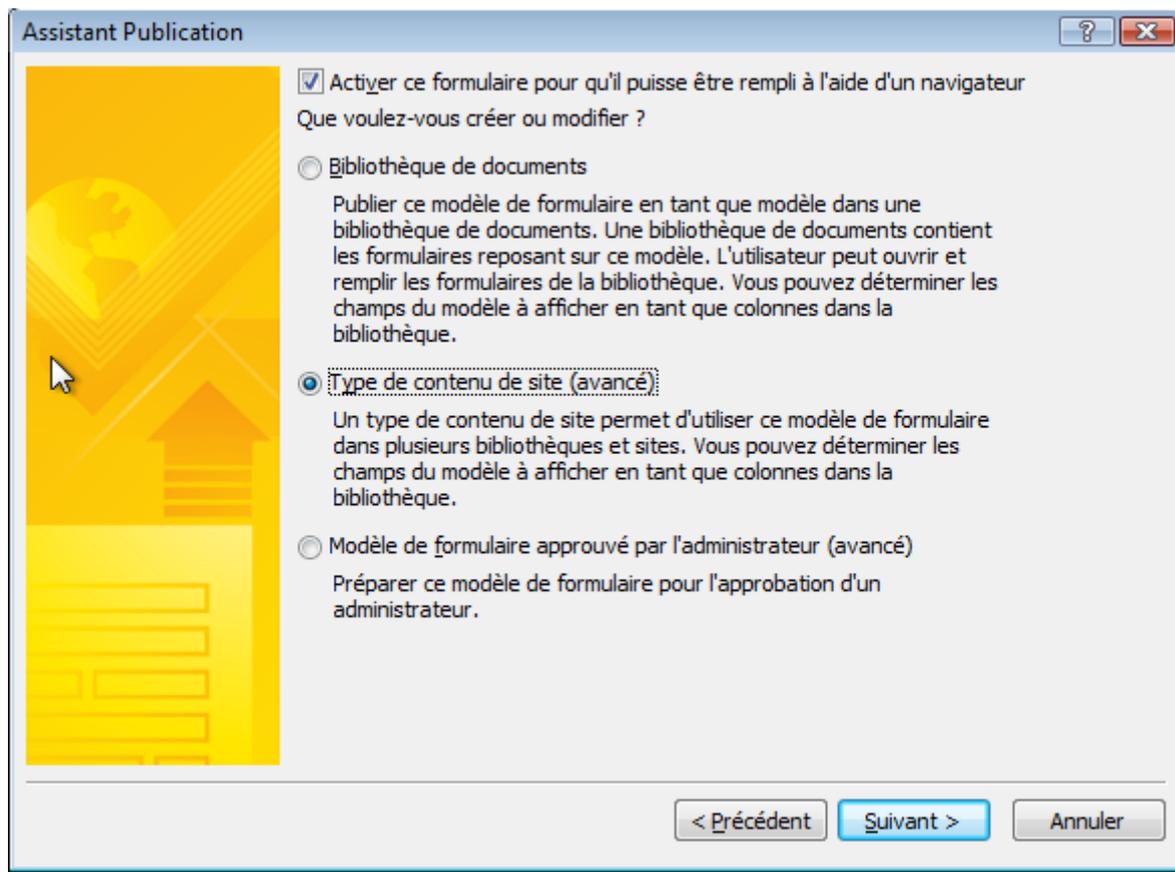


Nous cliquons sur *Suivant* et nous saisissons le top level site:

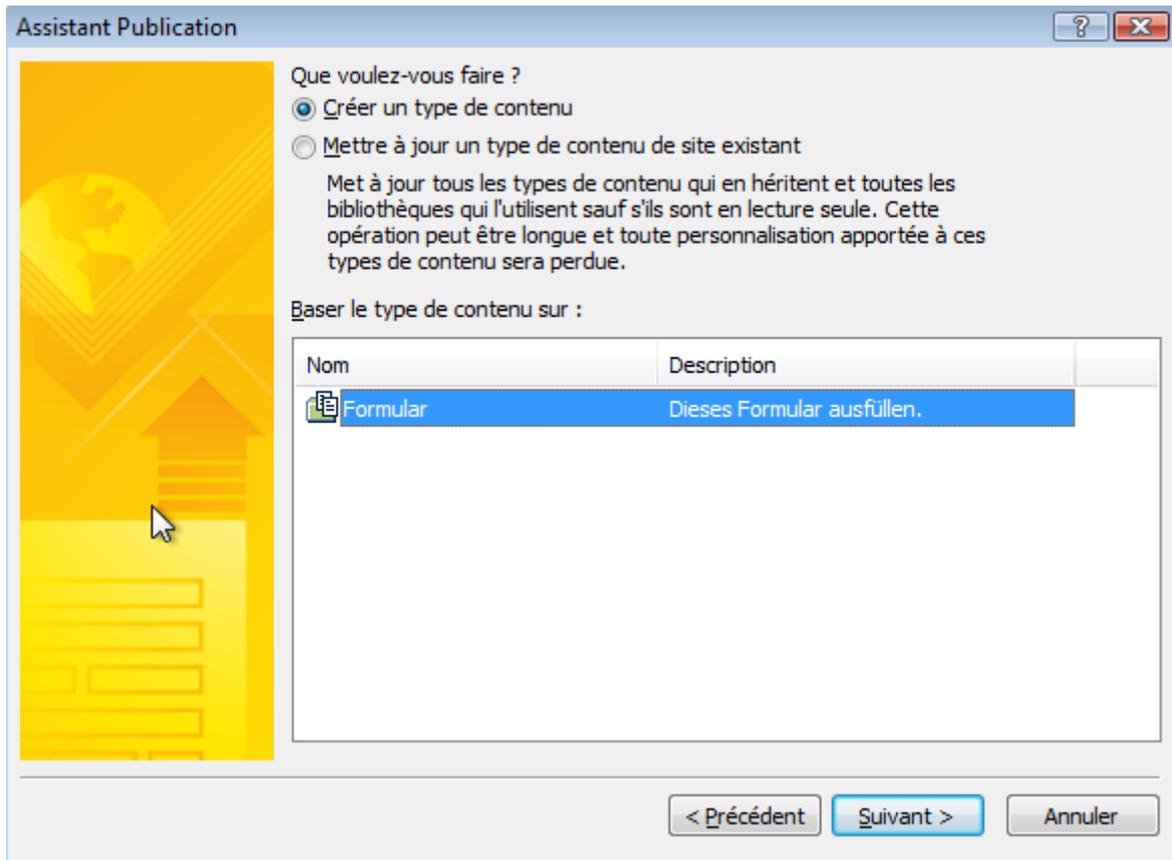


Vincent ISOZ

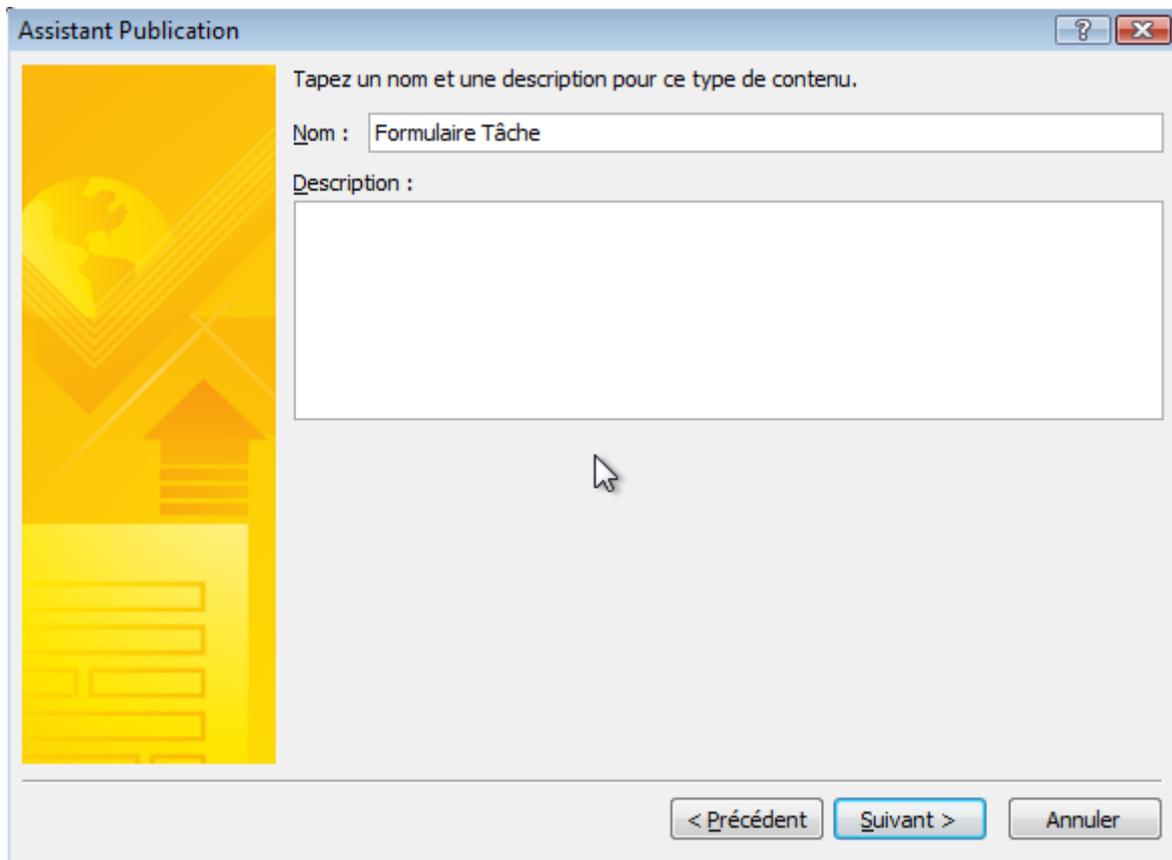
et nous cliquons sur *Suivant* et prenons *Type de contenu de site (avancé)*:



Nous cliquons sur *Suivant* en choisissant de *Créer un type de contenu*:

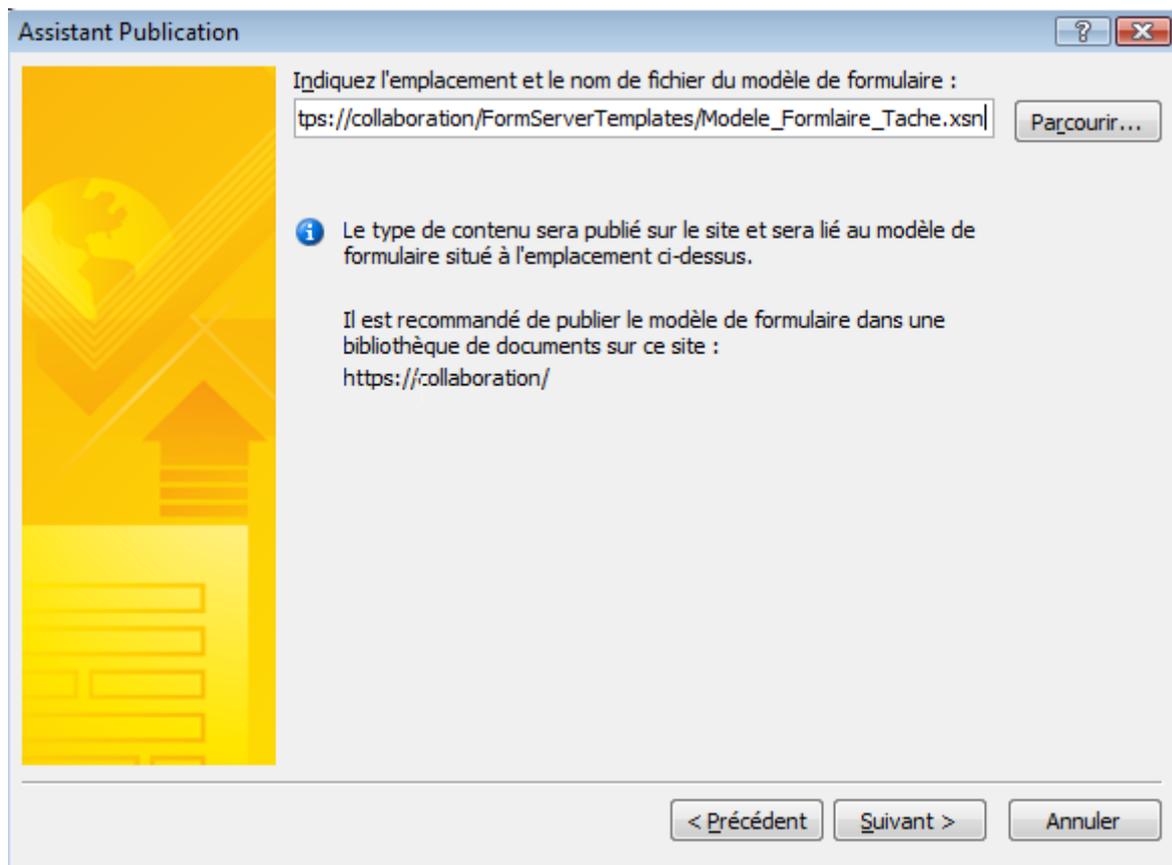


et nous cliquons sur *Suivant* pour y saisir le nom du type de contenu:

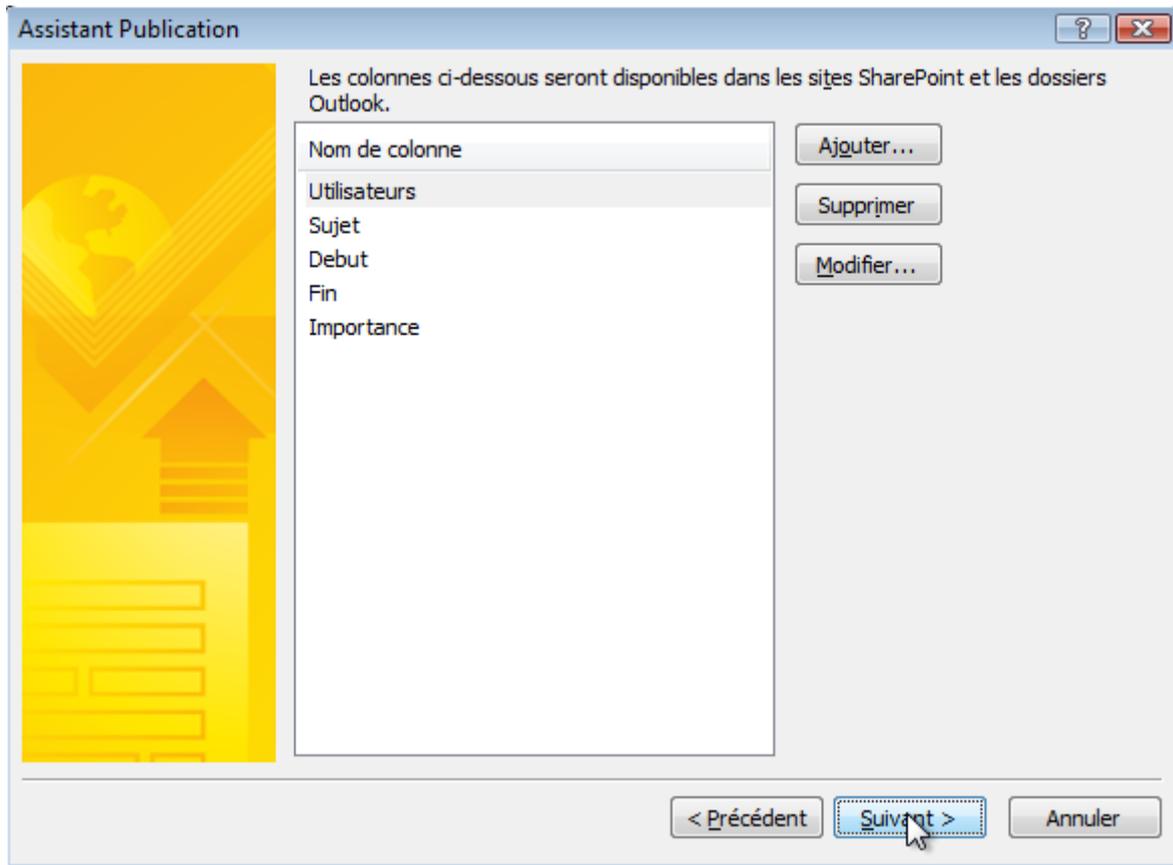


Vincent ISOZ

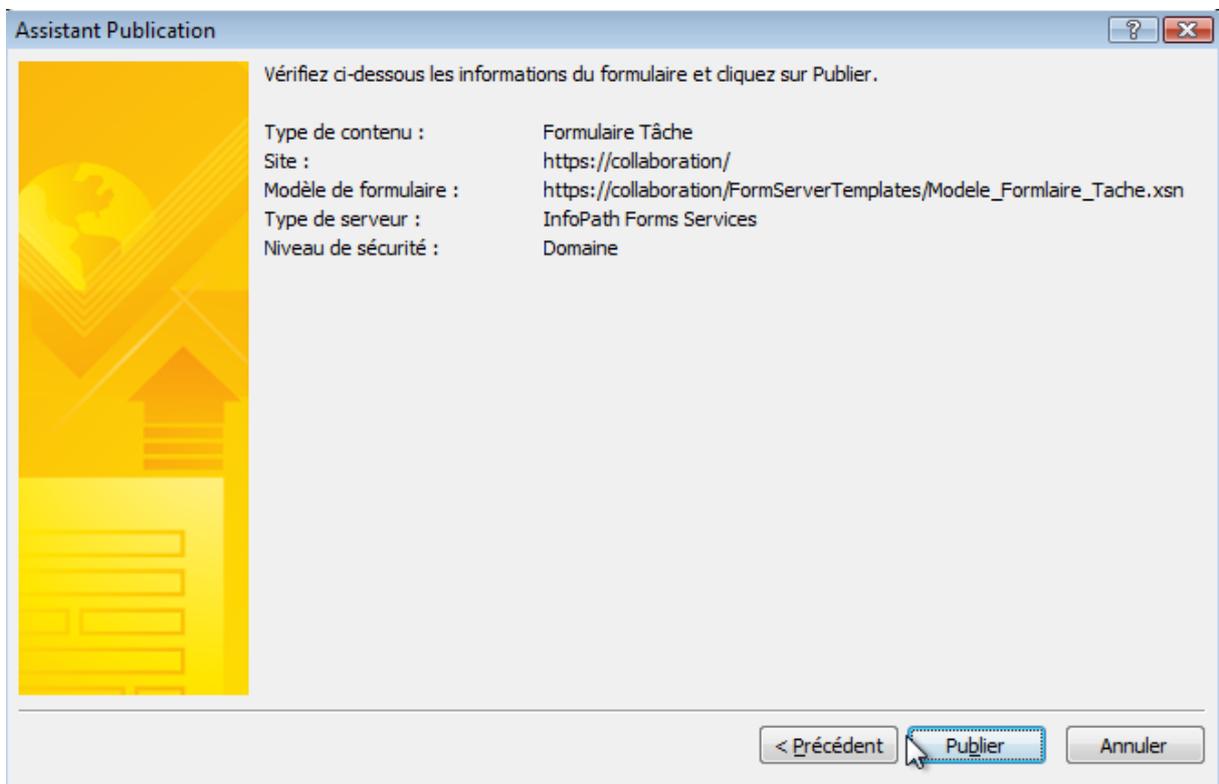
Nous cliquons sur *Suivant*:



et stockons le modèle de formulaire dans un librairie réservée à cet effet (soit elle doit être créée à l'avance, soit elle existe déjà). Nous cliquons sur *Suivant*:



et sur *Suivant*:



et enfin sur *Publier*.

Vincent ISOZ

Vous pourrez alors réutiliser ce modèle de type de contenu dans tous les sous-sites dont l'héritage de la sécurité est activé. Si vous avez cassé l'héritage c'est trop tard... Soit vous republiez le document dans le sous-site (et bonjour l'horreur...) soit vous réactivez l'héritage dans le sous-site en allant dans *Autorisation de site/Actions/Hériter des autorisation*:



et là aussi vous pleurerez car vous perdrez toutes vos autorisations.

### 11.8 Création d'une liste déroulante en cascade (v2007)

Dans l'exemple qui va suivre, nous souhaiterions avoir une liste de Pays et de Département (Cantons) se trouvant dans SharePoint 2007 et de pouvoir en faire usage sous forme de liste déroulante en cascade [dans un formulaire internet InfoPath Services 2007](#) (car le faire dans le client est très simple comme nous l'avons déjà vu!).

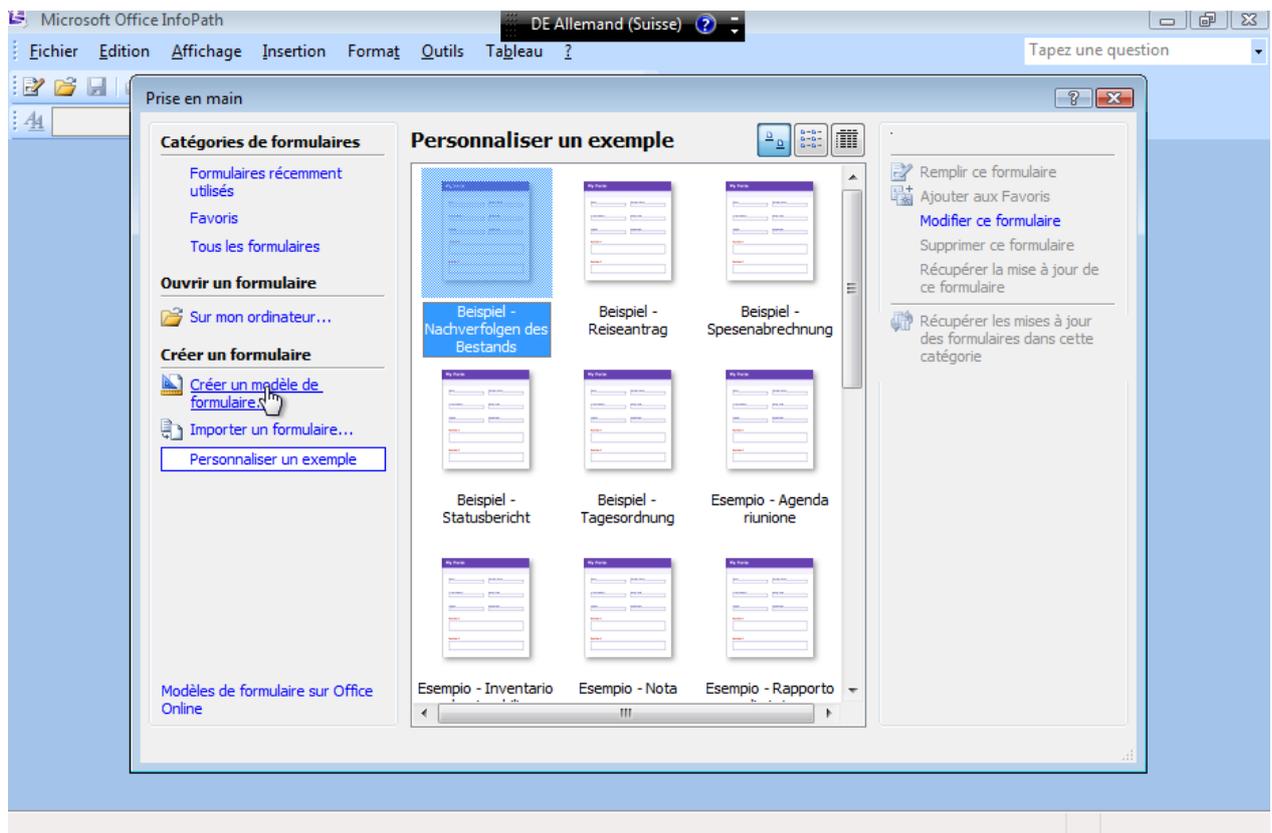
Nous créons sur le même site SharePoint une petite liste de pays:



et une liste de Département (Cantons) avec une colonne de type Lookup associée au pays:

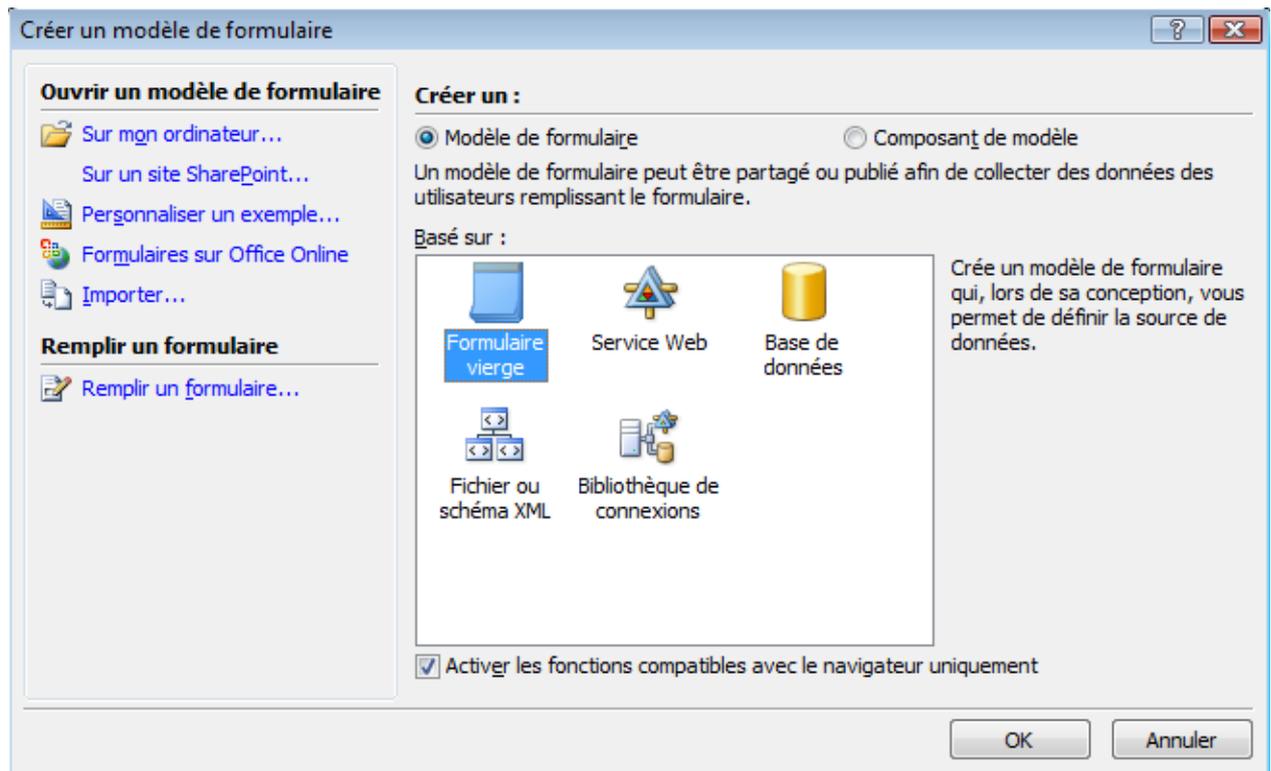


Ensuite, nous ouvrons InfoPath 2007:

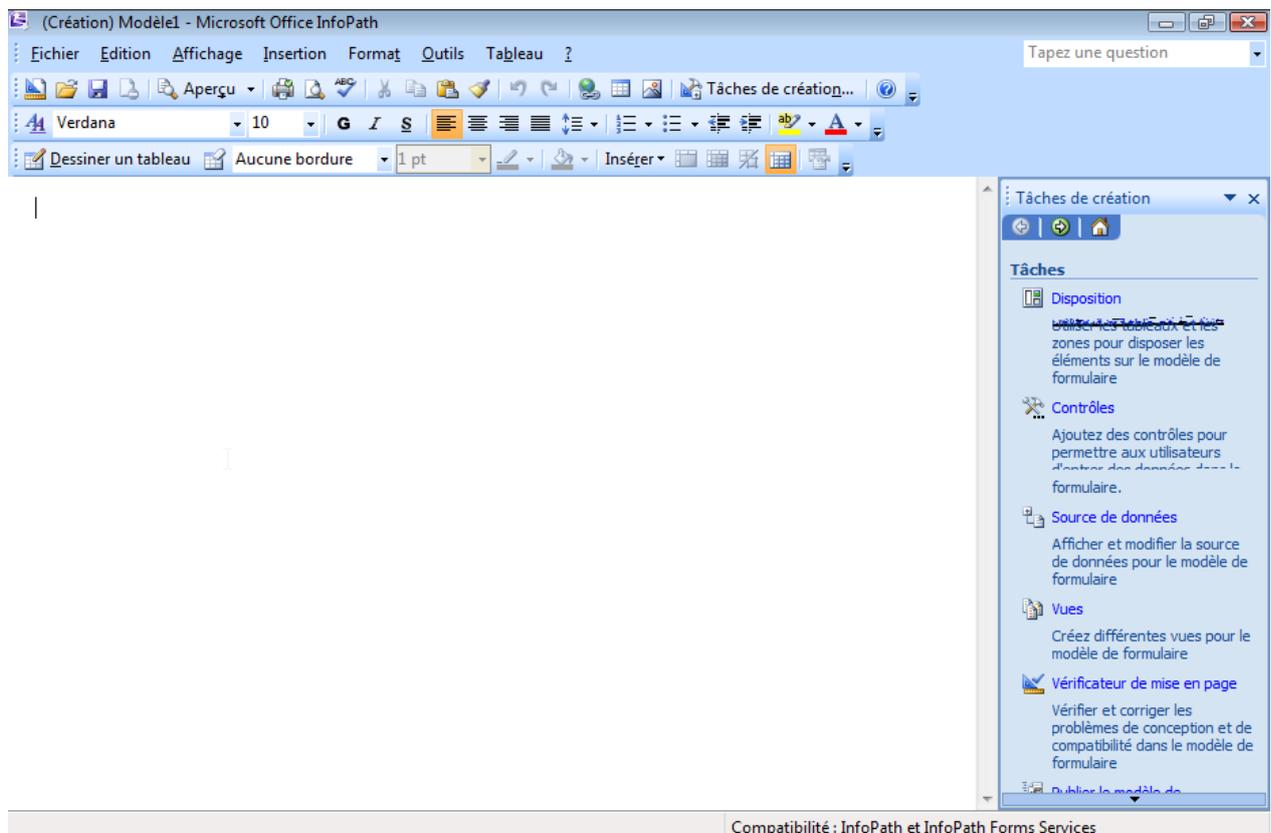


et nous cliquons sur **Créer un modèle de formulaire...**;

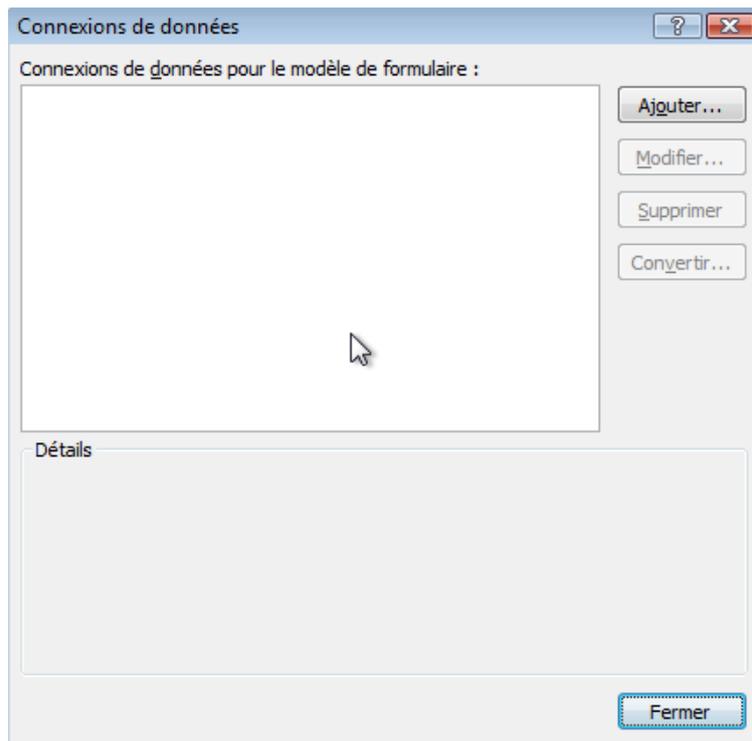
Nous sélectionnons ensuite **Formulaire vierge** en prenant bien garde à cocher la case **Activer les fonctions compatibles avec le navigateur uniquement**:



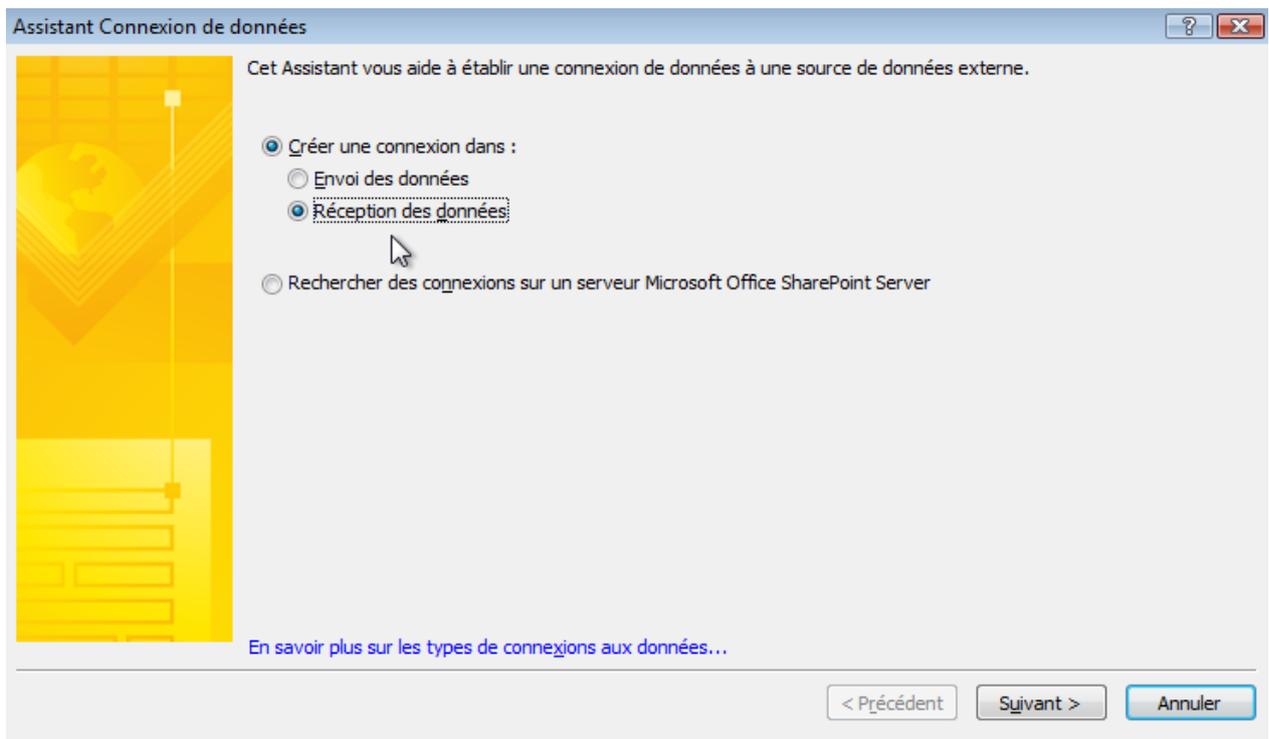
et nous validons par **OK**:



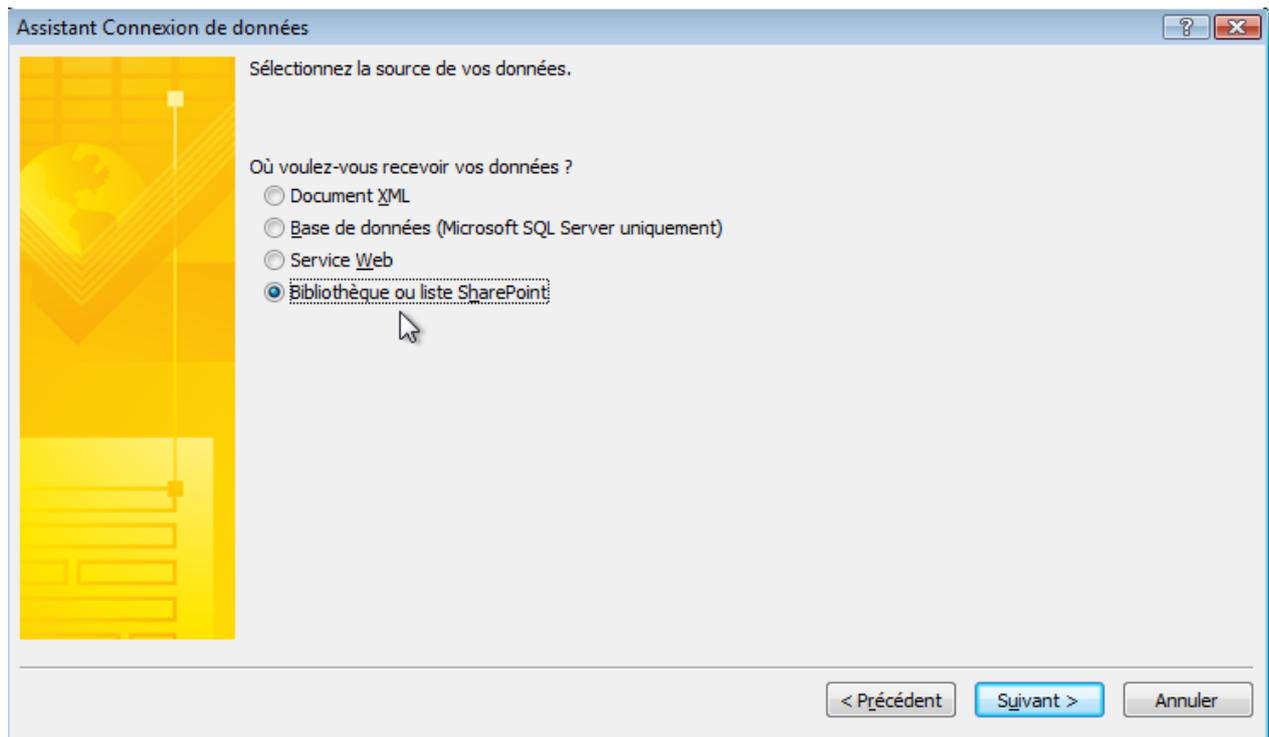
et nous créons de suite les connexions aux données. Nous allons donc dans le menu **Outils/Connexions de données**:



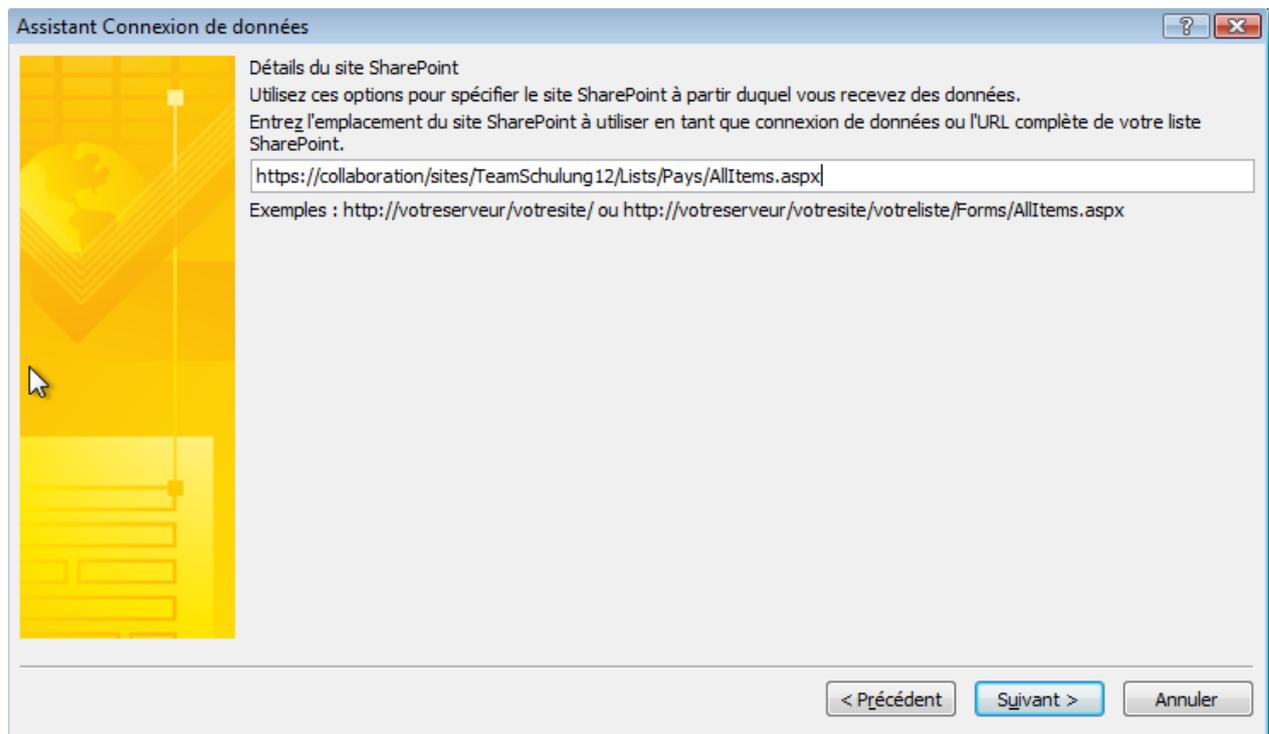
Nous cliquons sur **Ajouter**:



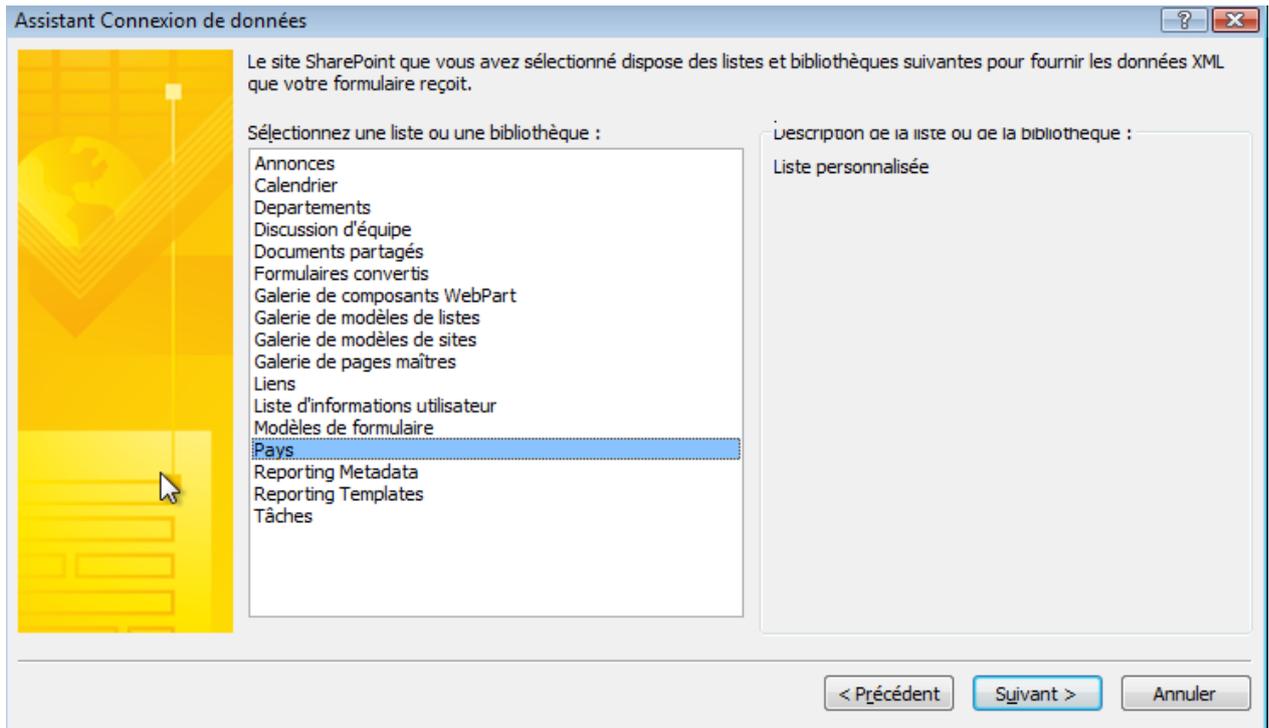
Nous cliquons sur **Suivant**:



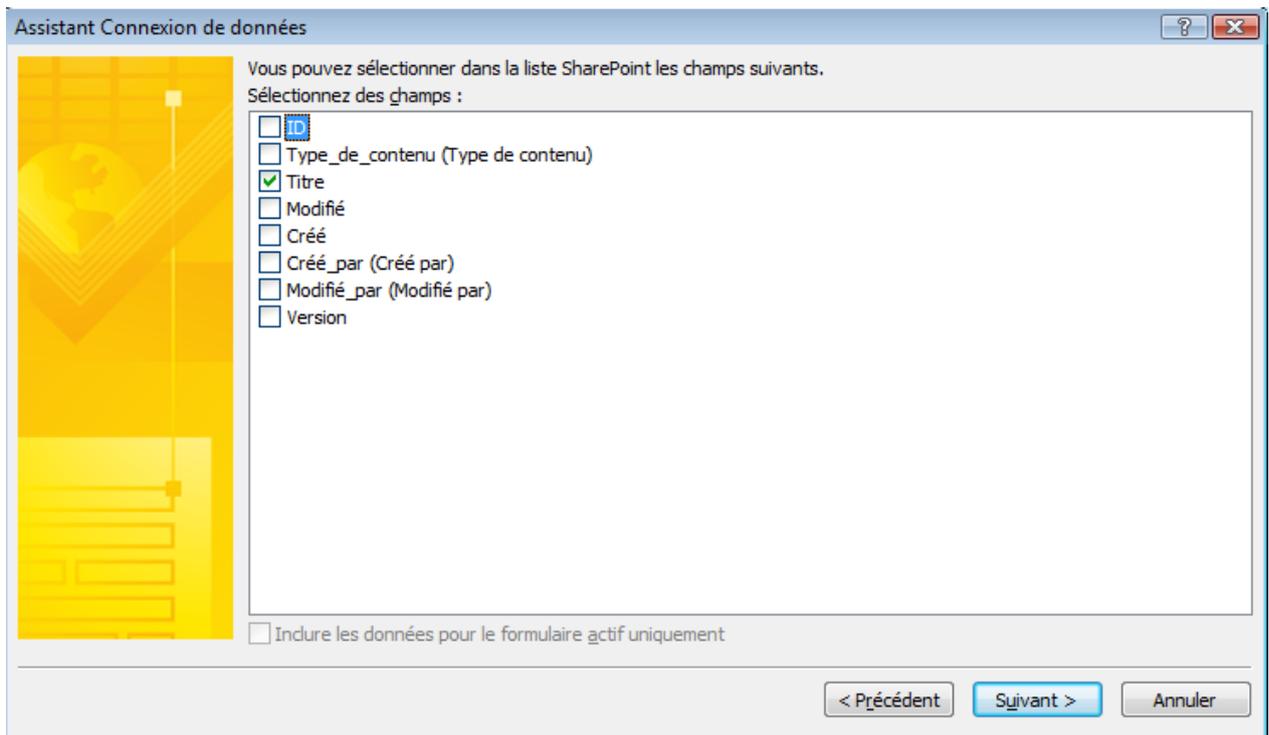
Nous cliquons sur **Suivant** et collons le lien vers la liste des Pays:



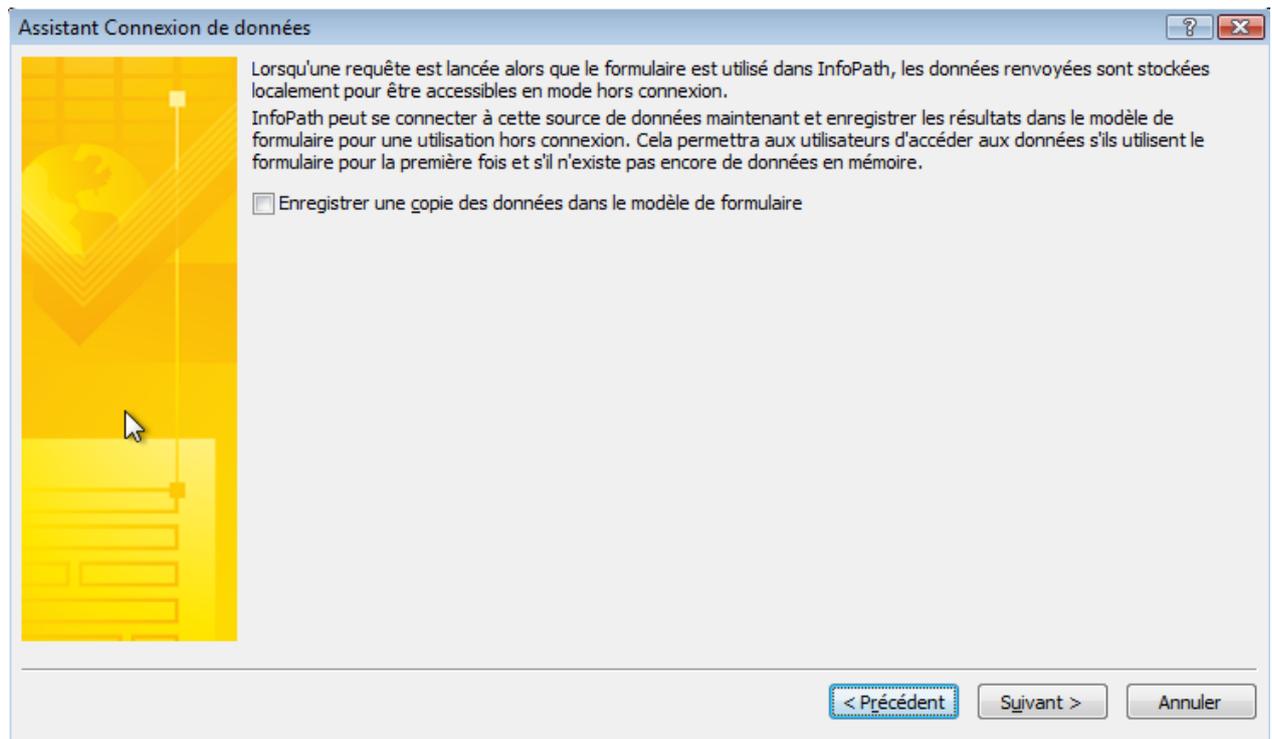
Nous cliquons sur **Suivant** deux fois:



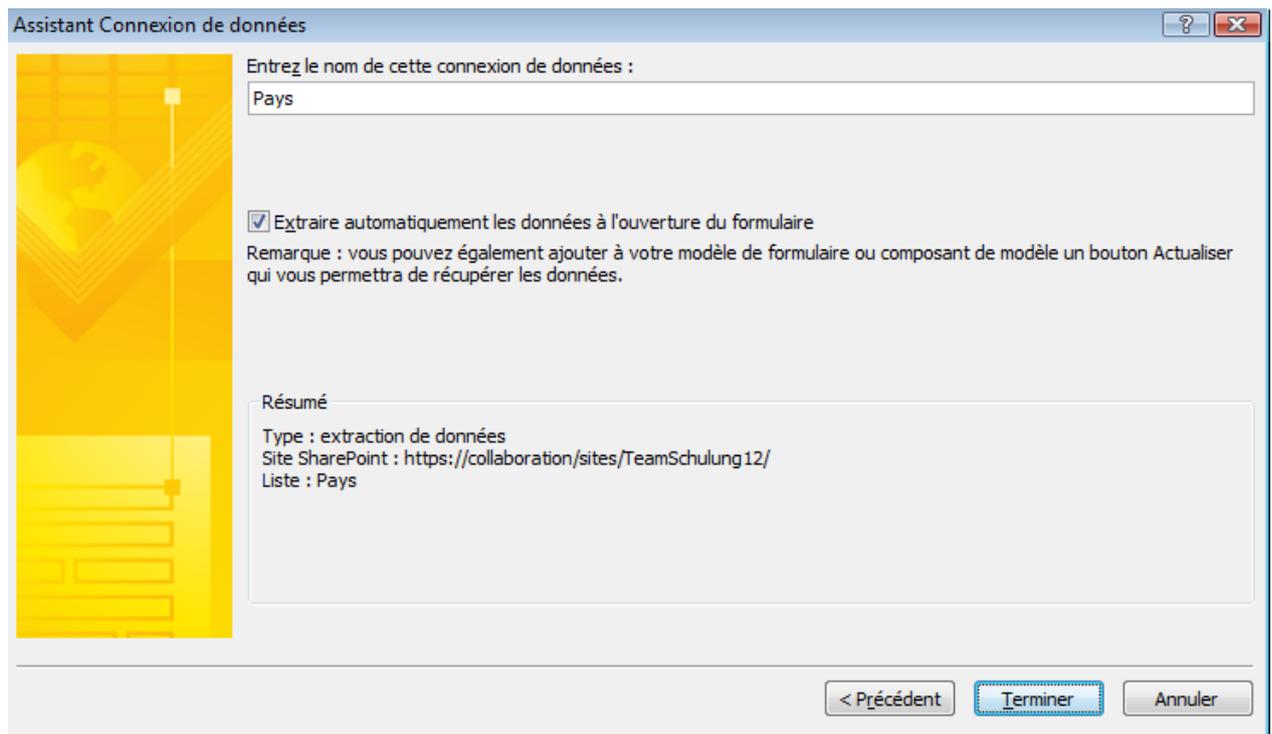
et nous sélectionnons le champ **Titre**:



Nous cliquons sur **Suivant** (cela devient lassant...):



encore une fois sur **Suivant**:



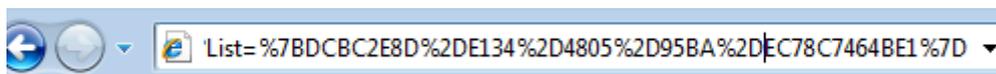
et enfin sur **Terminer**.

Pour les Départements (Cantons) vient l'astuce concernant InfoPath Services 2007....! Nous n'allons pas nous connecter à la liste SharePoint mais directement à son flux XML.

D'abord, nous devons trouver le GUID de la liste des départements. Pour cela, de retour sur la page sharepoint e la liste *Départements*, il vous faut aller dans **Paramètres Liste**:



et se concentrer sur le code de l'URL se trouvant après l'argument List:



Nous y avons donc:

**%7BDCBC2E8D%2DE134%2D4805%2D95BA%2DEC78C7464BE1%7D**

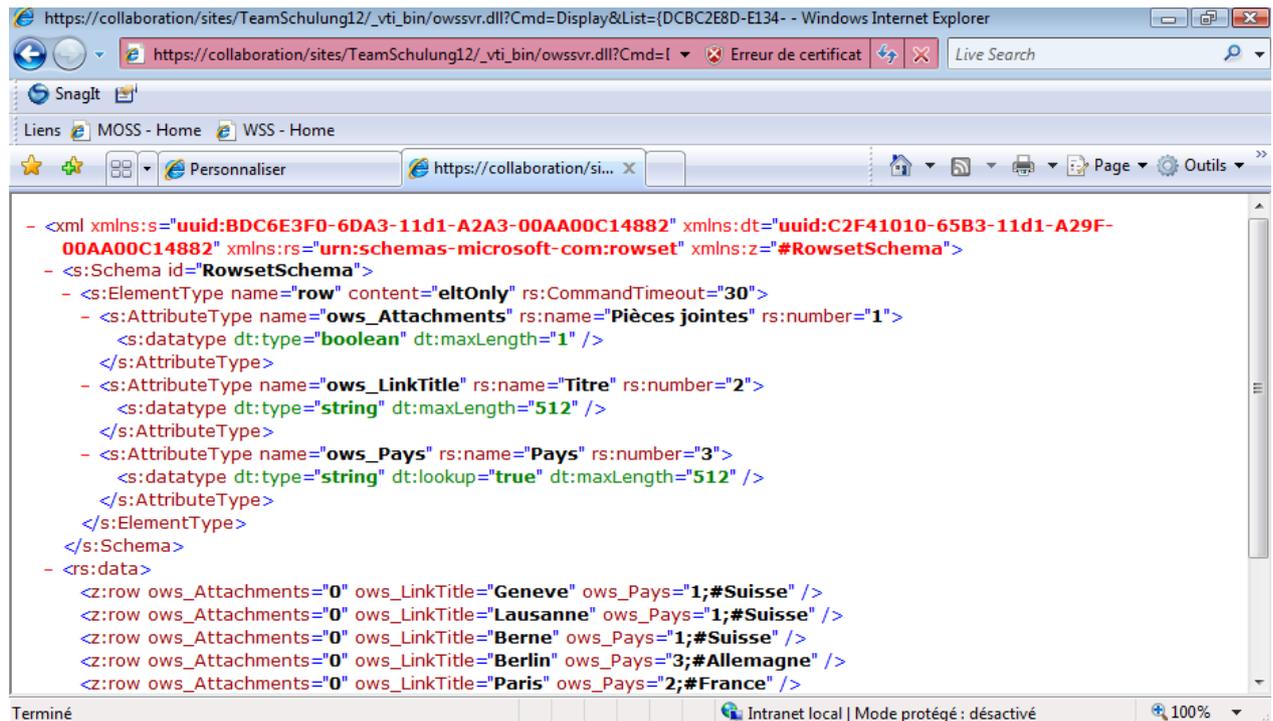
ce qui une fois nettoyé devient:

**{DCBC2E8D-E134-4805-95BA-EC78C7464BE1}**

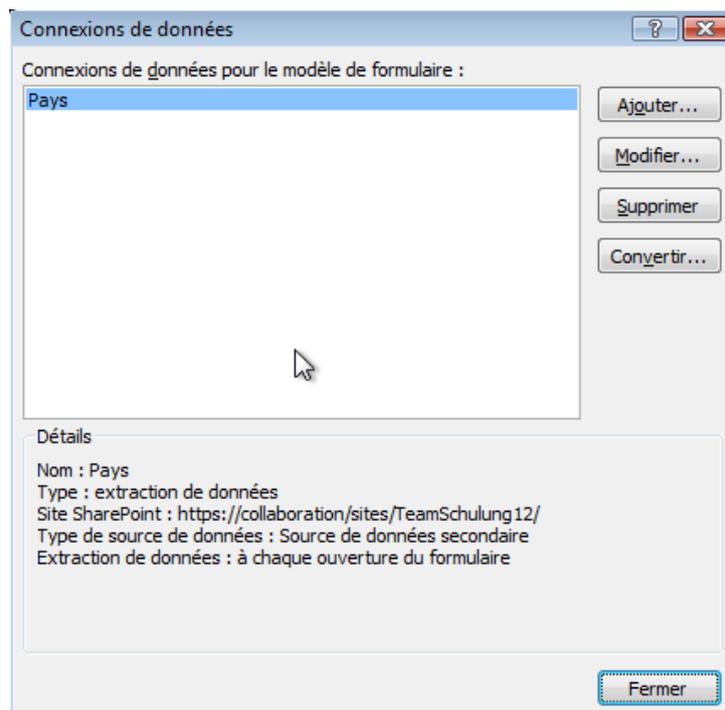
Maintenant que nous avons le GUID, nous allons reprendre l'URL de notre liste mais en la modifiant de la manière suivante:

**http://adresse\_site/\_vti\_bin/owssvr.dll?Cmd=Display&List={mettez l'ID ici}&XMLDATA=TRUE**

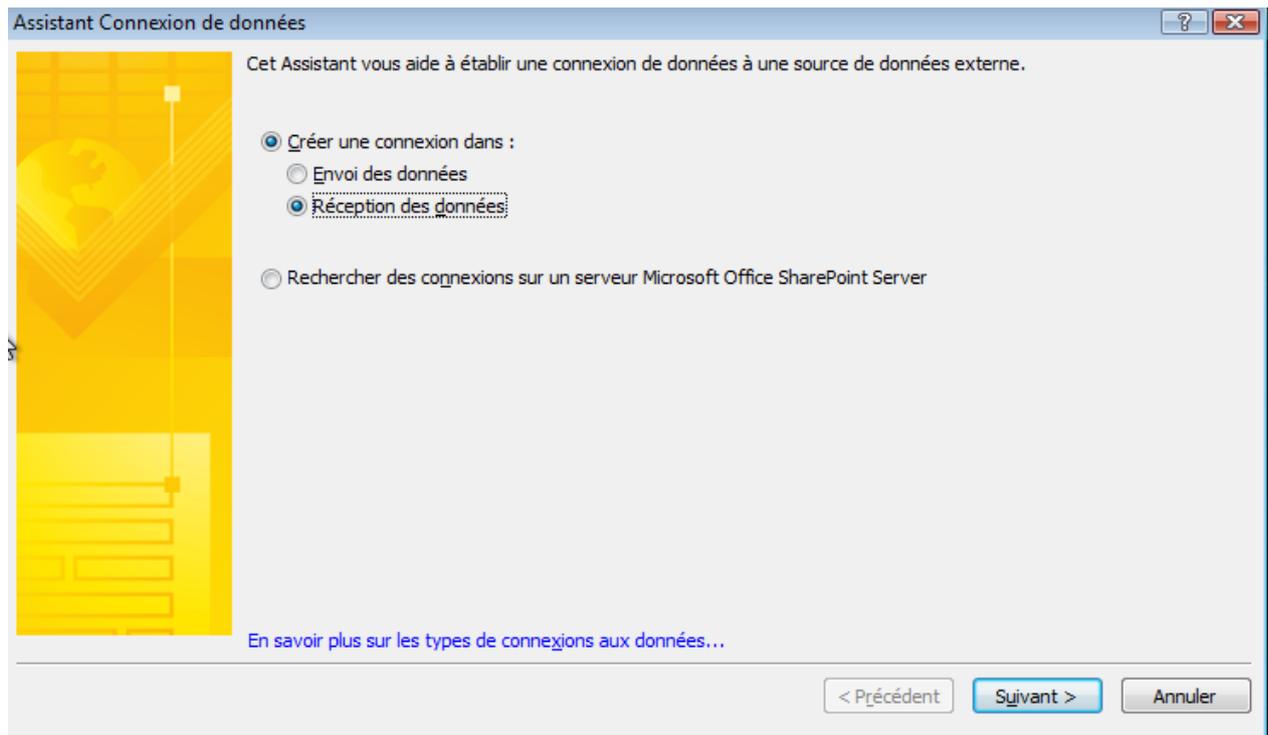
Vous aurez alors:



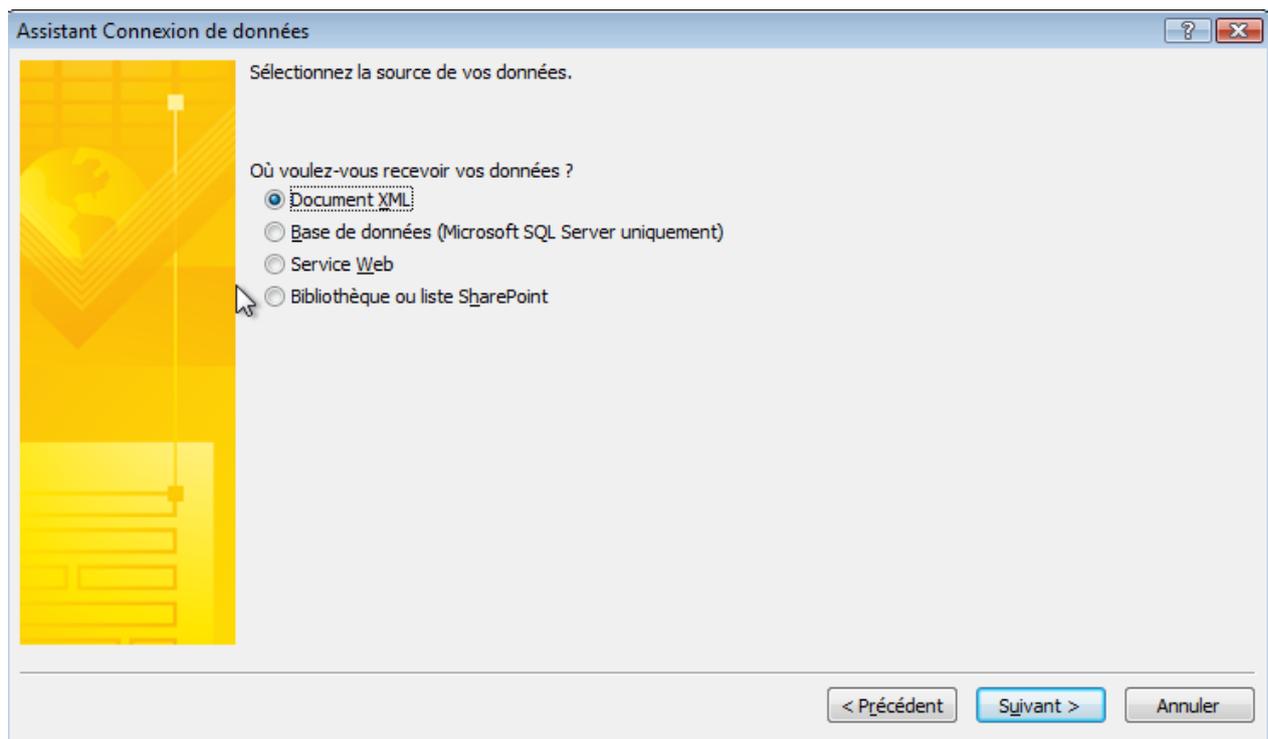
et nous créons alors une nouvelle connexion vers cette source de données en allant à nouveau dans la boîte de dialogue **Connexion de données**:



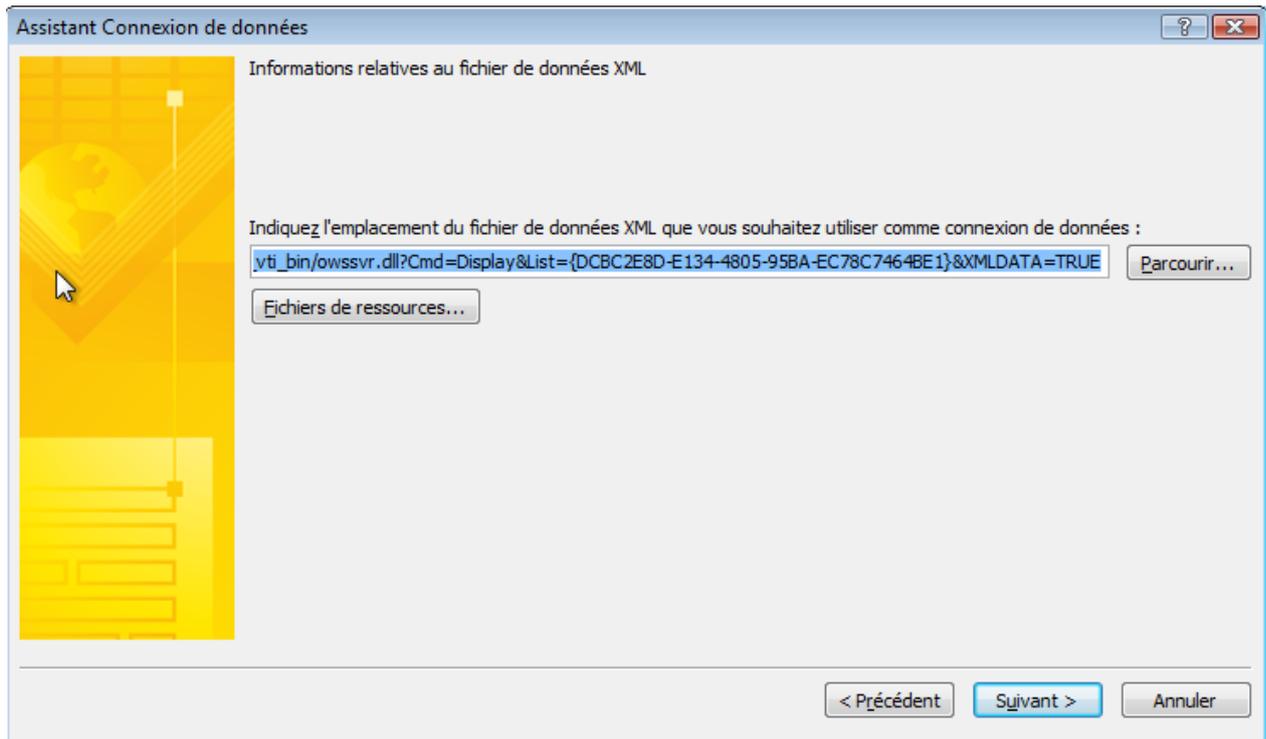
Nous cliquons sur **Ajouter**:



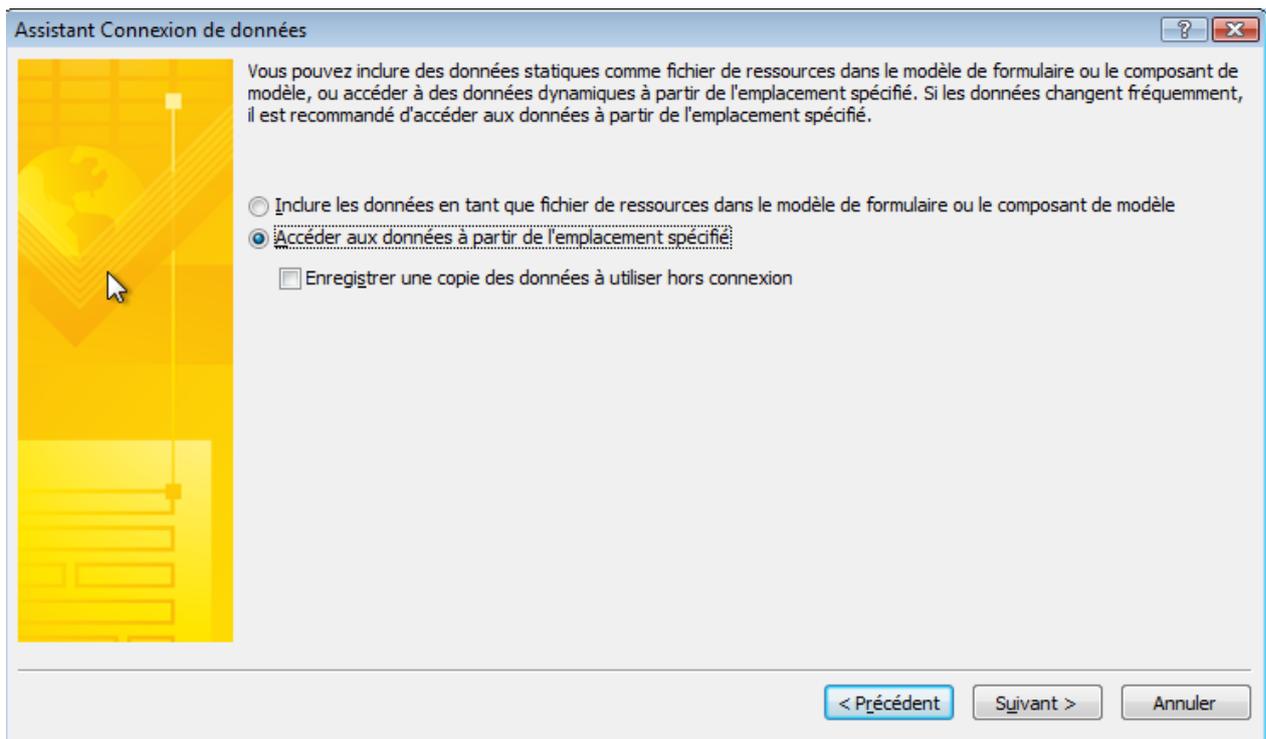
Et après avoir choisi les paramètres visibles ci-dessus, nous cliquons sur **Suivant**:



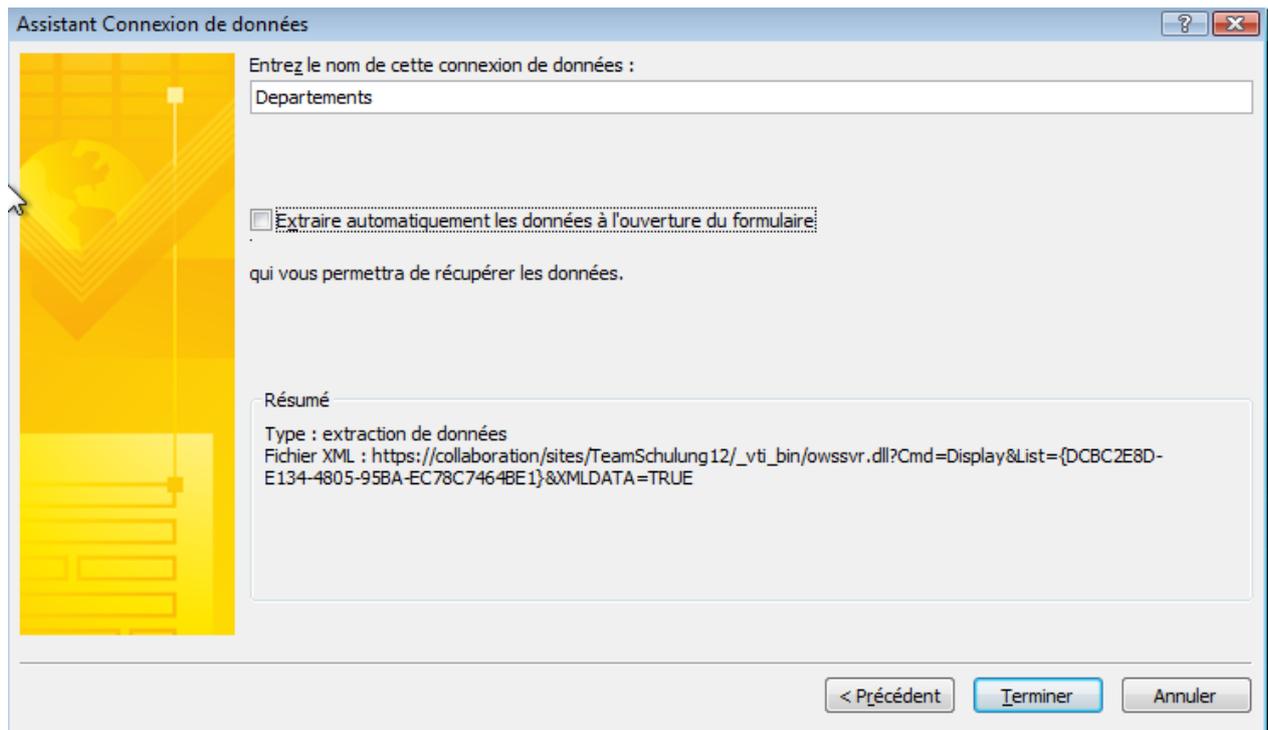
Nous laissons **Document XML** et cliquons sur **Suivant**:



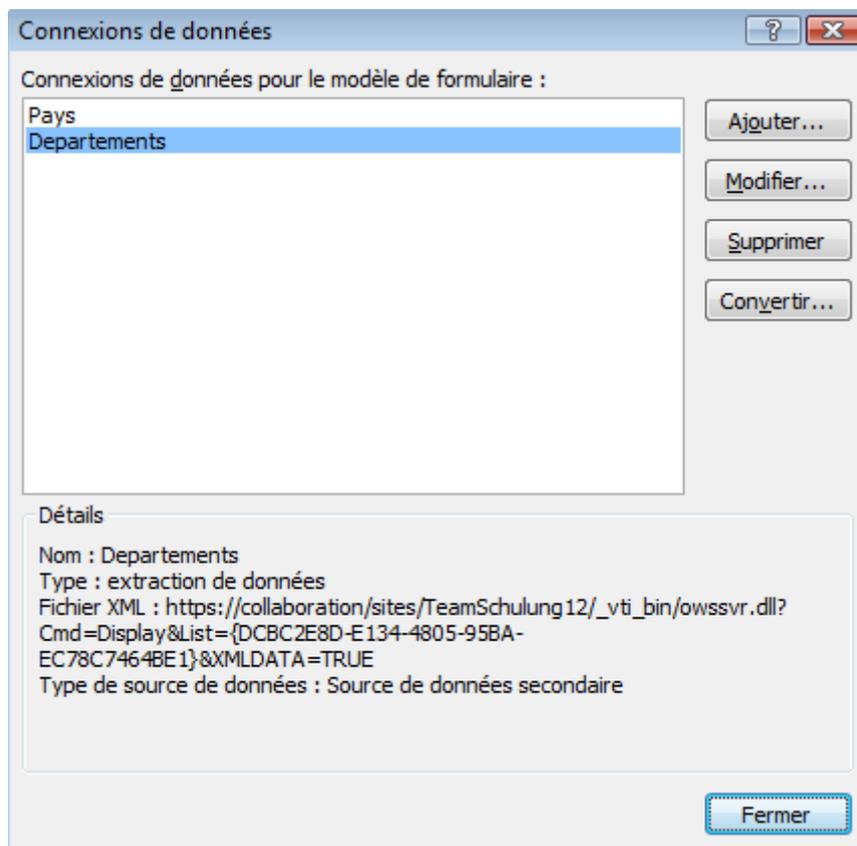
et nous y collons le lien test précédemment et cliquons sur **Suivant**:



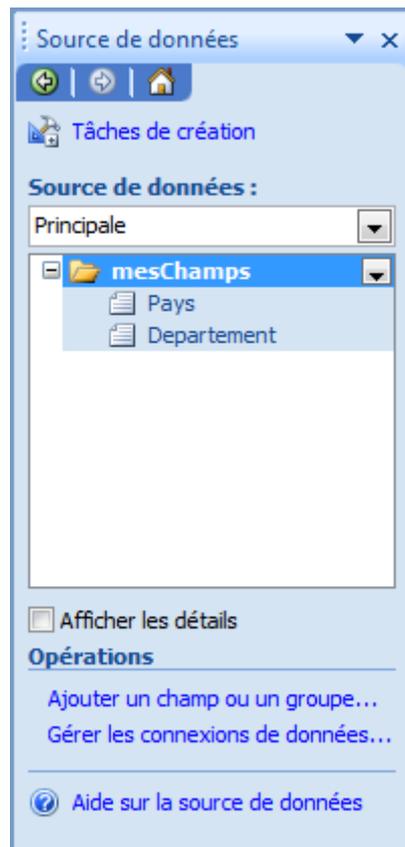
en choisissant les paramètres mis en évidence ci-dessus. Nous cliquons encore et toujours sur **Suivant**:



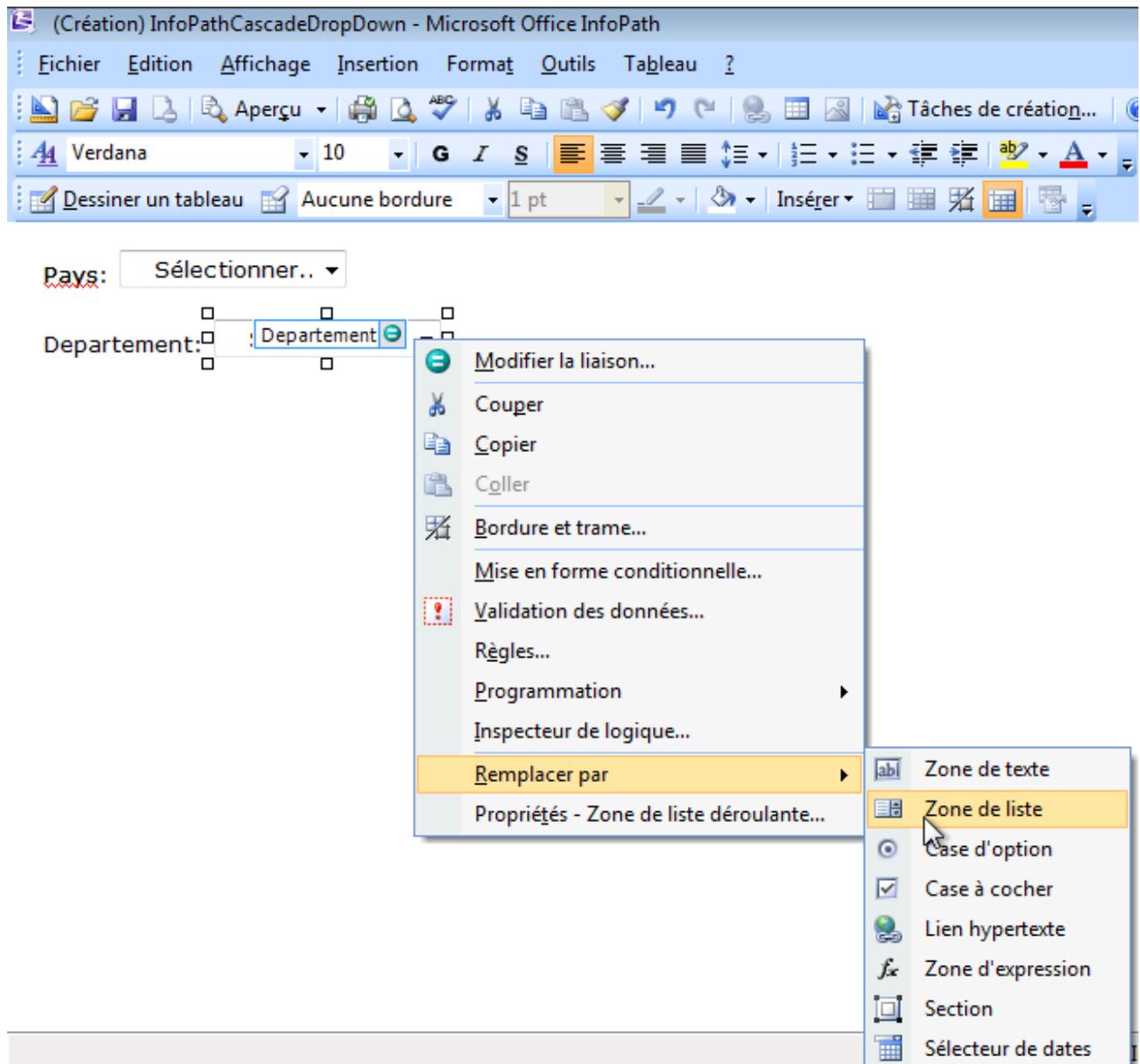
en prenant bien soin de **décocher** l'option **Extraire automatiquement les données à l'ouverture du Formulaire**. Et nous cliquons sur **Terminer**:



Maintenant, ajoutons les deux champs suivants de type **String** à la source de données de notre formulaire:

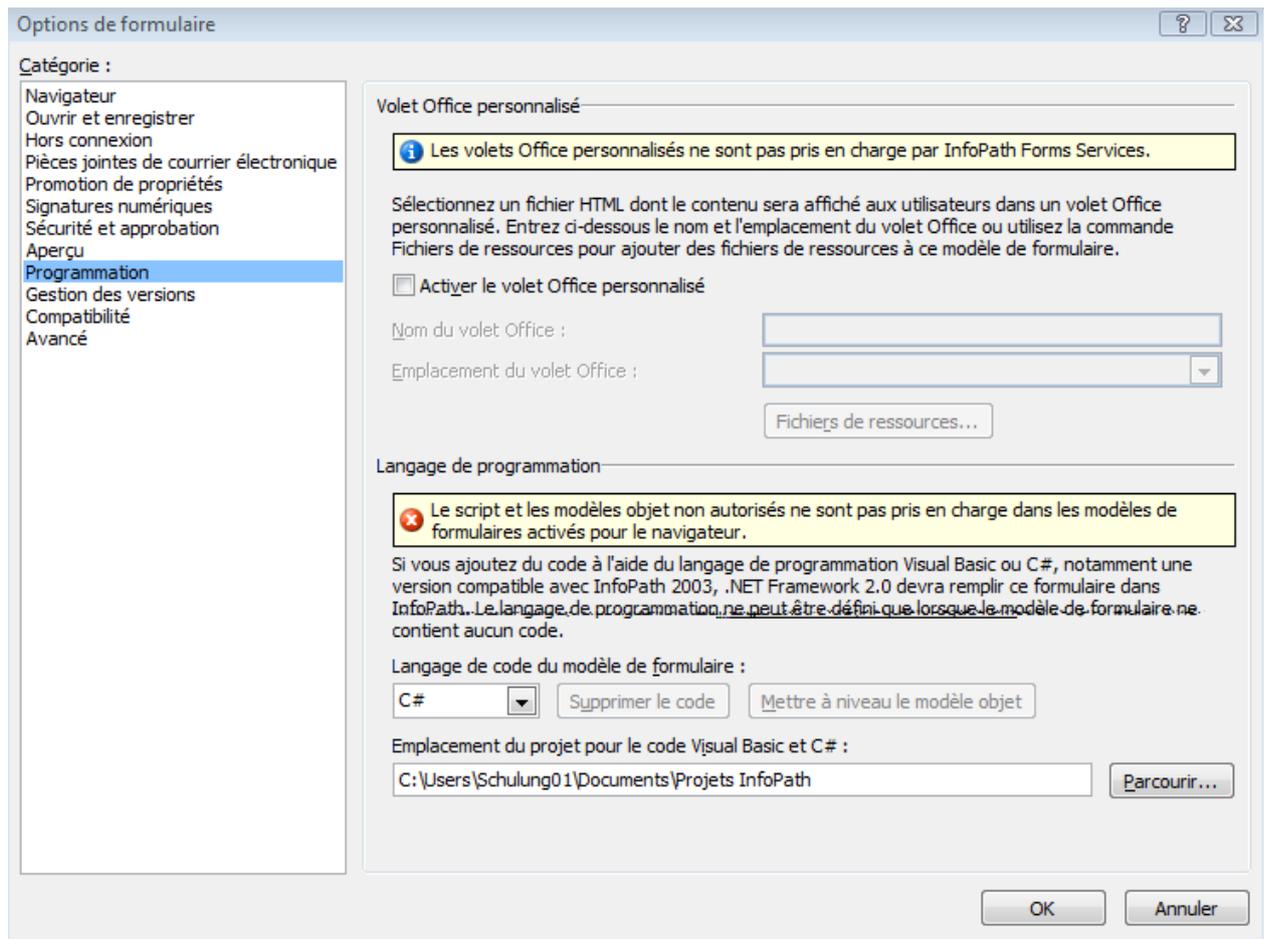


Ensuite, glissez les deux champs dans le formulaire en prenant soin de faire un clic droit dessus pour les convertir en liste déroulante:

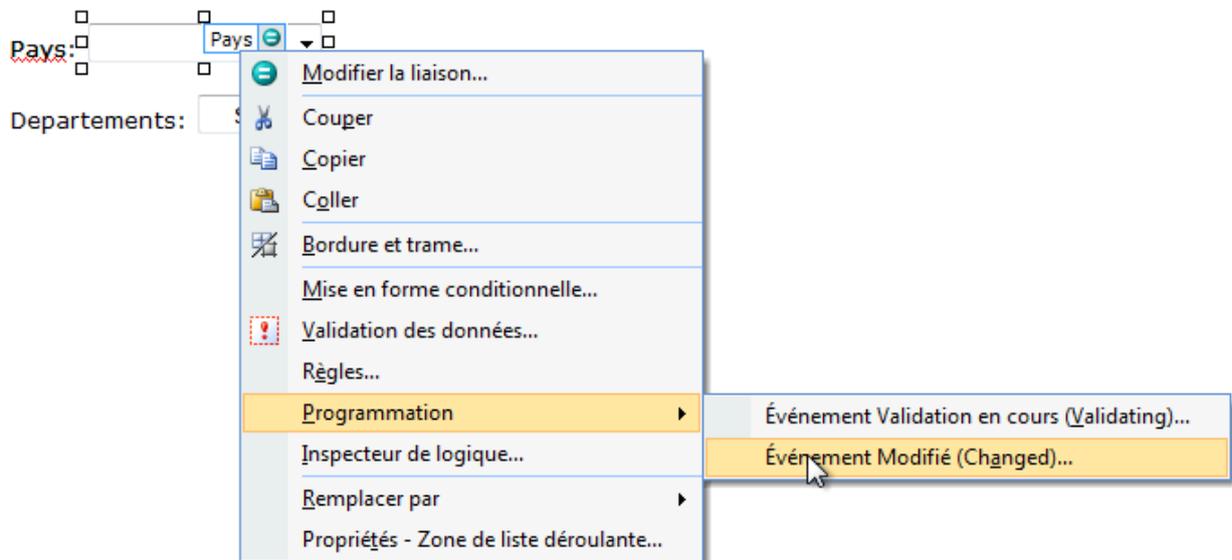


et allons dans les propriétés de la première liste déroulante pour y mettre:

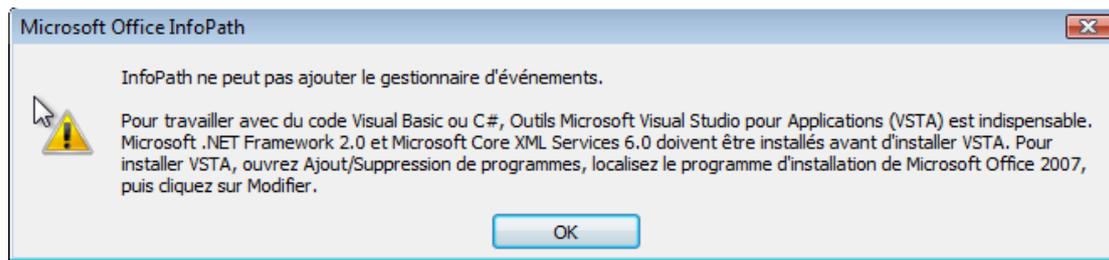
Évidemment, pour l'instant, si vous testez le formulaire, uniquement la liste déroulante des pays fonctionne. Mais pour avoir une liste en cascade qui marche avec InfoPath Services il va falloir faire un peu de code. Pour cela, nous allons déjà paramétrer InfoPath pour fonctionner avec du C#. Nous allons donc dans le menu **Outils/Options de formulaire/Programmation** nous changeons le type à C#:



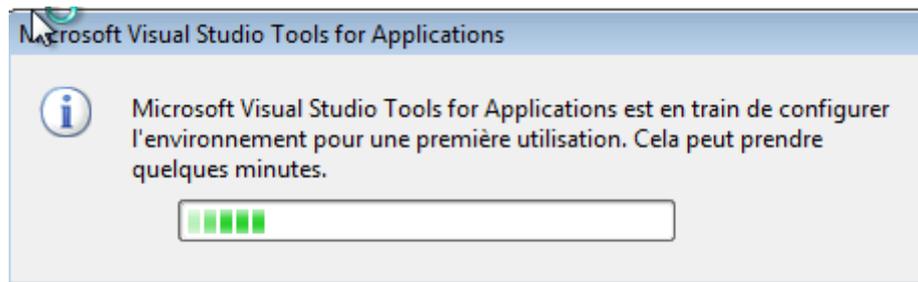
Ensuite, nous faisons un clic droit sur la liste déroulante *Pays*:



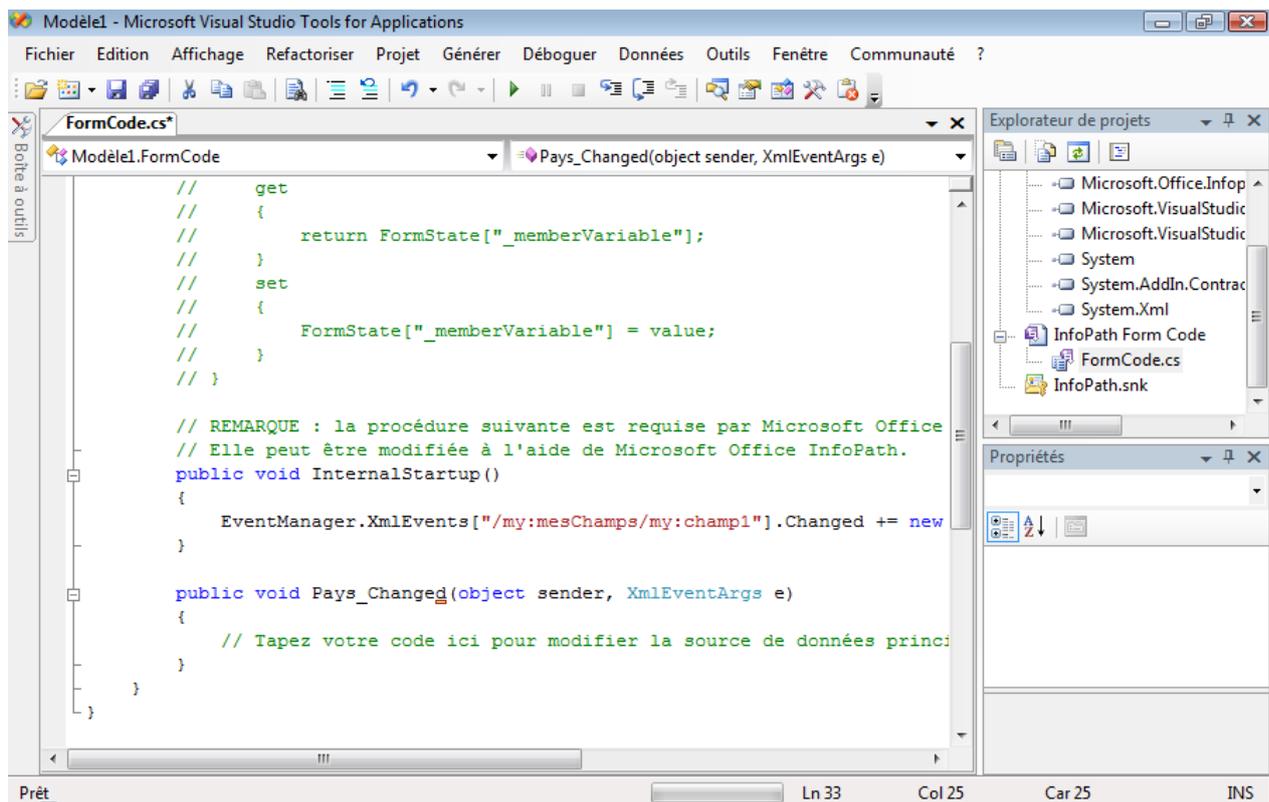
Si vous n'avez pas VSTA d'install sur votre ordinateur, vous aurez alors le message suivant:



Il faudra suivre ce qui est indiqué. Une fois ceci fait, l'action donnera:



et souvre alors:



mettez le code suivant en lieu et place de la partie public void Pays\_Change(...):

```
public void Pays_Changed(object sender, XmlEventArgs e)
{
    SetDepartementsOptions();
}
```

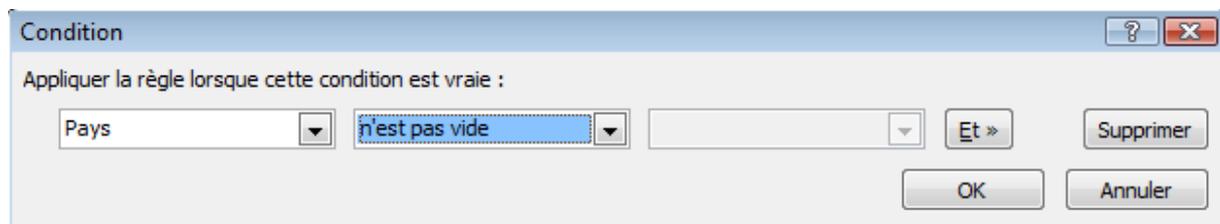
```
private void SetDepartementsOptions()
```

Vincent ISOZ

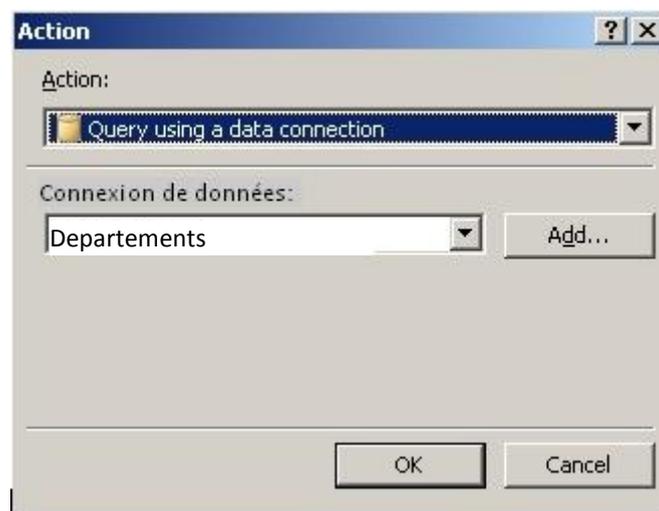
```
{  
FileQueryConnection q = (FileQueryConnection)this.DataConnections["Departements"];  
q.FileLocation = q.FileLocation + "&FilterField1=State&FilterValue1=" + GetStateValue();  
q.Execute();  
}
```

```
private string GetPaysValue()  
{  
XPathNavigator nav = this.CreateNavigator();  
string filterValue = (string)nav.SelectSingleNode("/my:mesChamps/my:Pays",  
this.NamespaceManager).ValueAs(typeof(string));  
return filterValue;  
}
```

Ensuite, faites un clic droit sur le champ *Pays* et définissez la règle suivante:



et quand vous cliquez sur **OK**, choisissez l'action suivante:



Vous validez par **OK** et il ne vous reste plus qu'à publier le formulaire... (sous réserve de problèmes ou de réglages de sécurité avec le serveur SharePoint).

Remarque: Signalons qu'il est devenu inutile de faire ce type de manipulation avec InfoPath Services 2010.

## 11.9 Ajout d'un élément à une liste SharePoint (v2003)

Dans cet exemple, nous allons étudier une technique qui permet d'ajouter un élément à une liste SharePoint de type *MesContacts* déjà existante (mais cette technique peut être utilisée pour toute autre liste bien sûr). Même si selon moi c'est inutile voyons comment procéder car il s'agit d'une technique indispensable pour ceux qui ont conçu leur SharePoint n'importe comment (sans analyse fonctionnelle, sans architecte SharePoint et sans spécialiste de la gouvernance SharePoint)

Pour ce faire, nous allons utiliser trois sources de données pour le formulaire:

1. Une connexion à la liste *MyContacts* SharePoint pour visualiser la liste de contacts
2. Une connexion au Web Service *lists.asmx* pour soumettre un nouveau contact
3. Une source XML qui utilisera le langage CAML

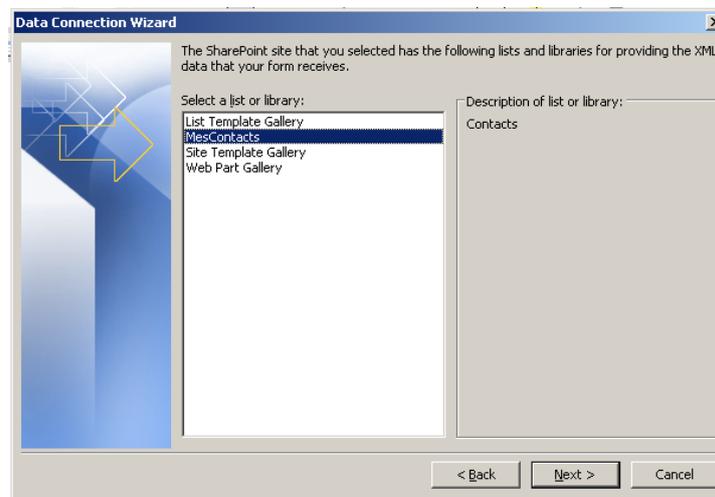
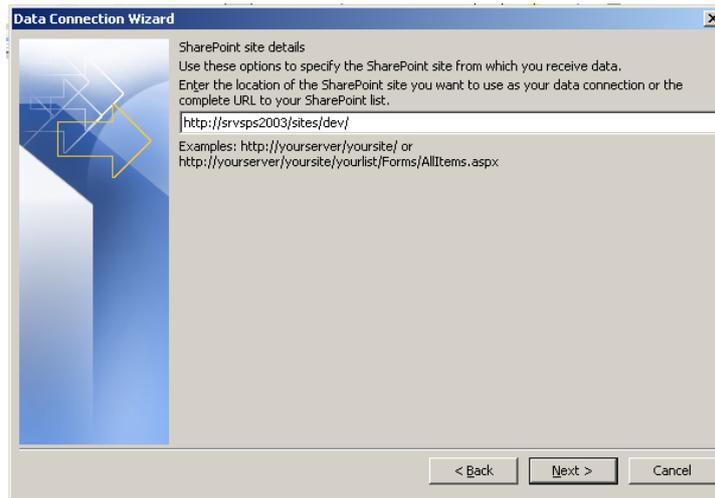
Copiez le code suivant dans un éditeur de texte et sauvegardez le fichier sous le nom *Add List Item Template.xml*:

```
<?xml version="1.0" encoding="UTF-8"?>
<Batch OnError="Continue">
  <Method ID="1" Cmd="New">
    <Field Name="Title"></Field>
    <Field Name="FirstName"></Field>
    <Field Name="Email"></Field>
    <Field Name="WorkPhone"></Field>
  </Method>
</Batch>
```

**Remarque:** Cela paraît simple mais pour d'autres listes que celle des Contacts, c'est cauchemardesque à gérer, surtout s'il y a des listes déroulantes ou des champs liés au carnet d'adresse de l'entreprise. Il vaut mieux en tous les cas, attendre InfoPath 2010 pour faire cela.

Maintenant que le modèle CAML a été sauvegardé, attaquons nous à la création d'un nouveau formulaire vierge avec une nouvelle source de données de type XML (nous ne décrivons pas les étapes car à ce stade du document cette manipulation est supposée triviale).

Nous allons maintenant ajouter une connexion à une liste SharePoint de type contacts donnée. Nous supposons à nouveau cette manipulation connue (cf. exemples précédents).



Sélectionnez lors de l'assistant de connexion les champs *Last\_Name*, *First\_Name*, *E-mail\_Address* et *Business\_Phone* et désélectionnez tous les autres.



SharePoint stocke les listes en utilisant un identificateur unique nommé GUID pour chaque liste. Pour implémenter des spécifications particulières celles que nous sommes entrain de développer, vous devez connaître le GUID de votre liste.

Vincent ISOZ

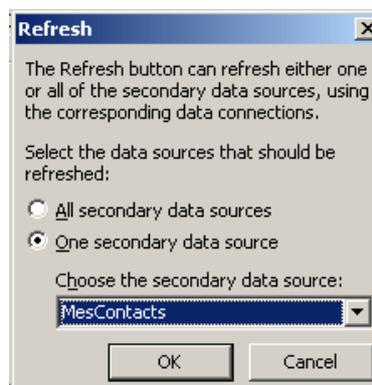
Pour obtenir le GUID de votre liste, vous devrez:

1. Ouvrir Internet Explorer et sur votre site SharePoint aller sur le site où se situe votre liste de contacts.
2. Cliquez sur le lien *Modify Settings and Columns*
3. Repérer dans l'URL la valeur entre guillemets à la fin de l'URL et copiez celle-ci dans le bloc-notes pour un usage ultérieur.



Revenez maintenant dans InfoPath et dans la vue par défaut ajoutez-y un bouton.

1. Double cliquez sur le bouton et dans l'option *Action* sélectionnez *Refresh*. Apparaît alors la boîte de dialogue suivante:

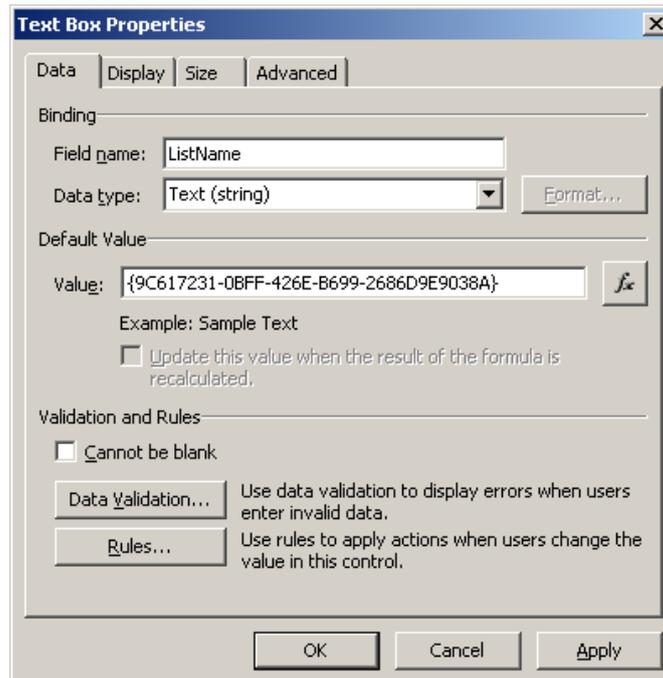


Sélectionnez les mêmes paramètres.

Validez le tout et allez quelques lignes en-dessous du bouton

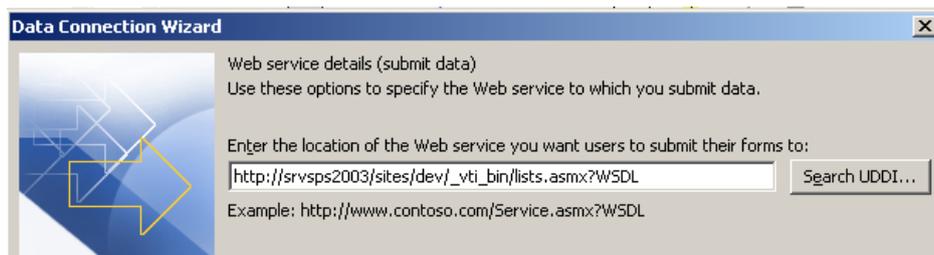
2. Ouvrez le panneau Office et dans les *Data Source* sélectionnez *MesContacts* comme *Data Source*.
3. Faites un clic droit sur le noeud *myFields/dataFields/MesContacts* node et choisissez l'option *Repeating Table*
4. Faites encore une fois quelques retours à la ligne pour descendre un peu en-dessous du tableau.
5. Insérez ensuite un champ de type *Text Box*.

6. Double cliquez sur ce nouveau champ et renommez-le *ListName* et copiez-y le code GUID de la liste *MesContacts* comme valeur par défaut. Validez le tout



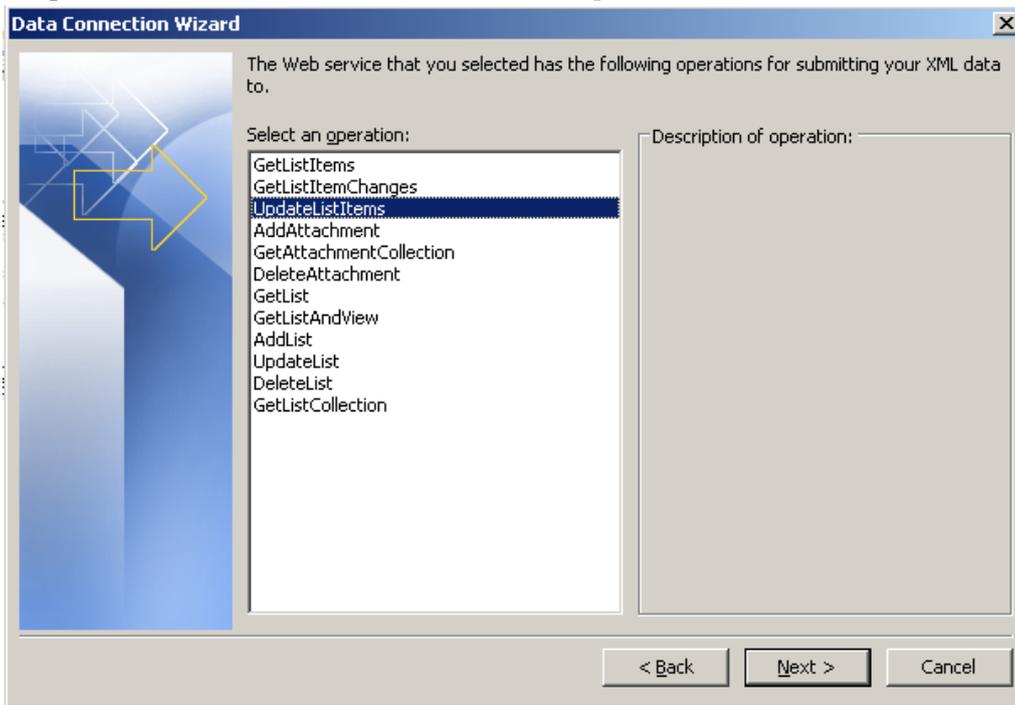
Nous avons maintenant tout ce qu'il faut pour ajouter la connexion avec le web service de SharePoint. Allons-y:

1. Allez dans *Tools/Data Connections* et cliquez sur le bouton *Add*
2. Sélectionnez l'option *Submit Data* et ensuite cliquez sur *Next*. Sélectionnez l'option *Web Service* et cliquez sur *Next*.

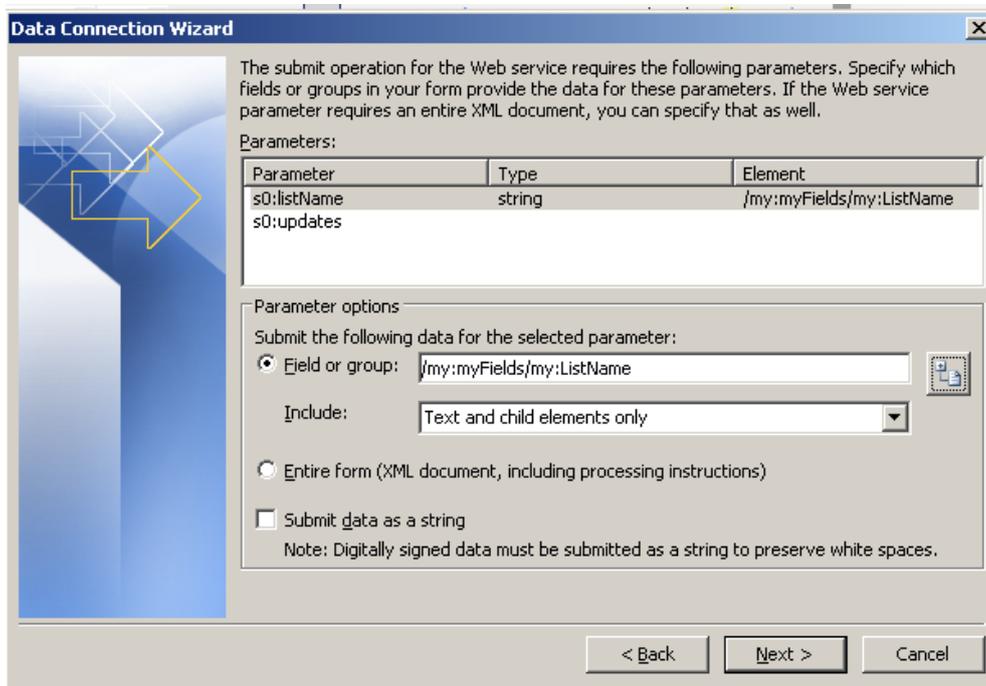


3. Saisissez l'adresse de votre *lists.amx*. Normalement cela devrait être quelque chose du genre: `http://sharepointserver/lesiteaveclistedecontacts/_vti_bin/lists.asmx`

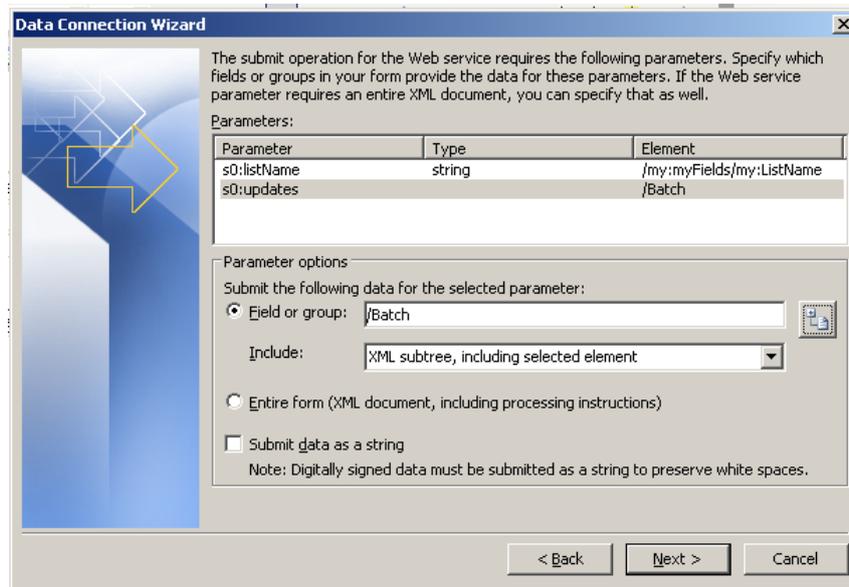
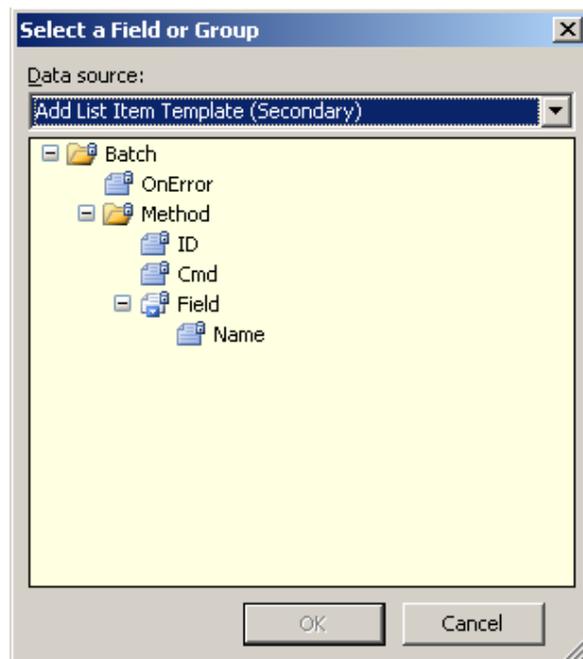
4. Cliquez sur *Next* et sélectionnez la méthode *UpdateListItems*



5. Ensuite, sélectionnez le paramètre *s0:listName* et assignez-y la valeur */my:myFields/my:ListName*



6. Sélectionnez le paramètre *s0:updates* et cliquez sur le bouton source en sélectionnant le nœud *Batch*:



7. Dans l'option *Include* choisissez *XML Subtree, including selected element*.

Maintenant que toutes les connections sont proprement configurées, nous pouvons finir notre formulaire.

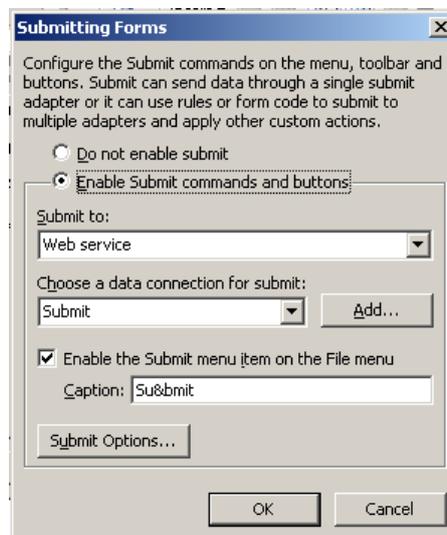
Activons à nouveau le panneau Office et sélectionnons *Add List Item Template* des sources de données. Faisons ensuite un clic droit sur le noeud */Batch/Method/Field* et choisissons l'option *Repeating Table*. Faisons ensuite un double clic sur le champ *Name* et dans l'onglet *Display* activons la case *Read-Only* et validons par *OK*.

Remarque: Le champ doit être en read-only car s'il change, notre modèle CAML sera affecté et la soumission échouera.

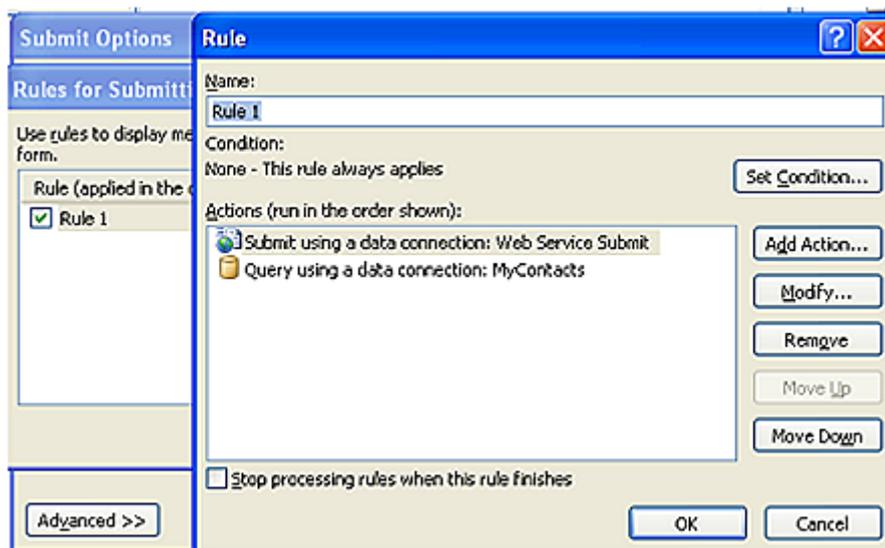
Faites quelques retour à la ligne en-dessous de la nouvelle table et insérez-y un élément de type *Button*.

Vincent ISOZ

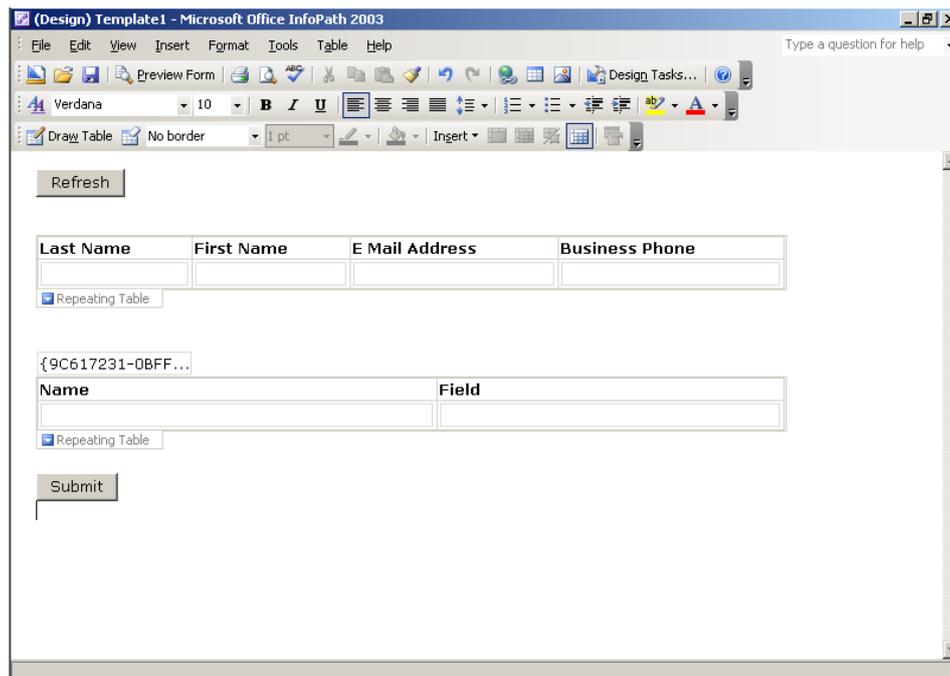
Double cliquez sur le bouton, sélectionnez comme action *Submit*. Ensuite, sur la boîte de dialogue qui apparaît, sélectionnez *Enable Submit commands and button* et dans *Submit to* prenez *Web Service* et validez le tout par *OK*.



Cliquez sur le bouton *Add...* à droite de *Submit* et choisissez de soumettre la liste *MesContacts*:



Il ne vous reste plus qu'à teste le formulaire:



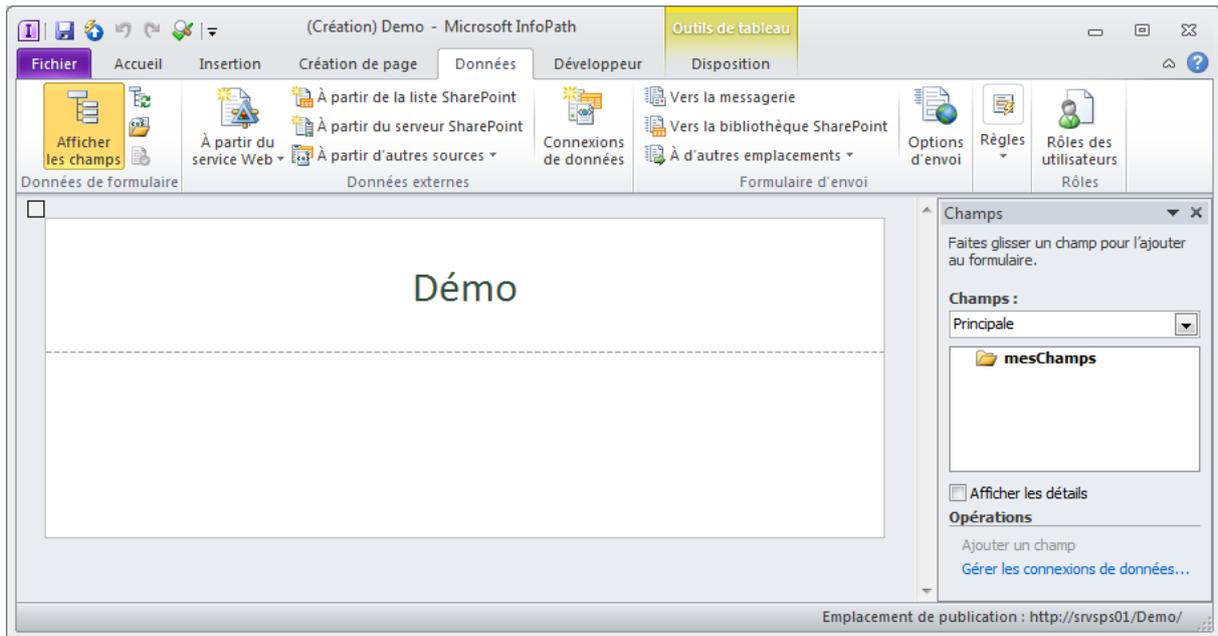
1. Activez le formulaire via le *Preview* (ou faites *Alt+P*)
2. Cliquez sur le bouton *Refresh* pour visualiser dans le tableau supérieur la liste des contacts déjà existante
3. Si dans la partie inférieure vous saisissez un nouveau contact et faites *Submit* et *Refresh* vous verrez que le nouveau contact a été ajouté à la liste SharePoint.

### ***11.10 Incrémenter automatiquement un nom de fichier formulaire InfoPath avec l'ID de la librairie SharePoint (v2010)***

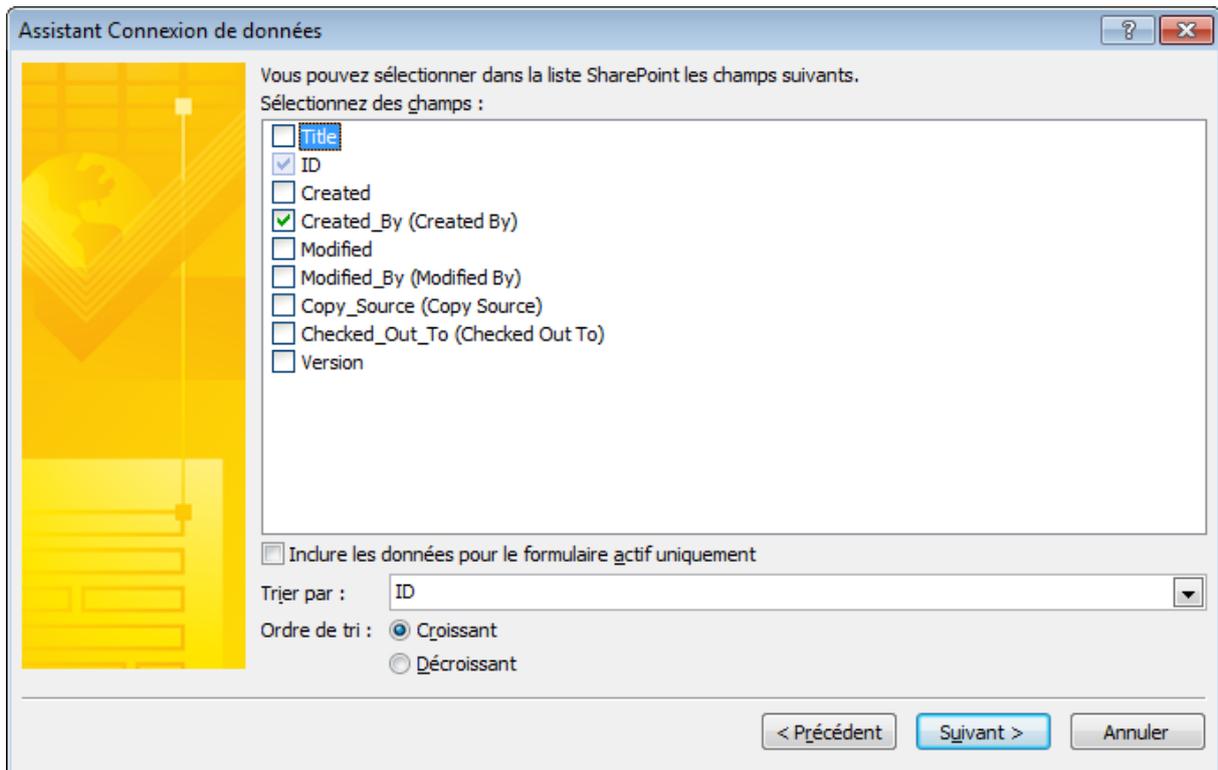
Attention cette technique qui permet d'éviter à vos utilisateurs de taper un nom de fichier (gain de temps), qui vous permet de garantir des noms homogènes de fichiers (puisque le nommage se fait automatiquement) comporte un certain nombre de problèmes subtils dans la pratique.

Nous reviendrons sur ces derniers une fois que nous aurons présenté cette approche simple de nommage automatique d'enregistrement de formulaires (il existe effectivement plusieurs approches possibles suivant le niveau d'impact qu'ont les limitations de la technique présentée ci-dessous).

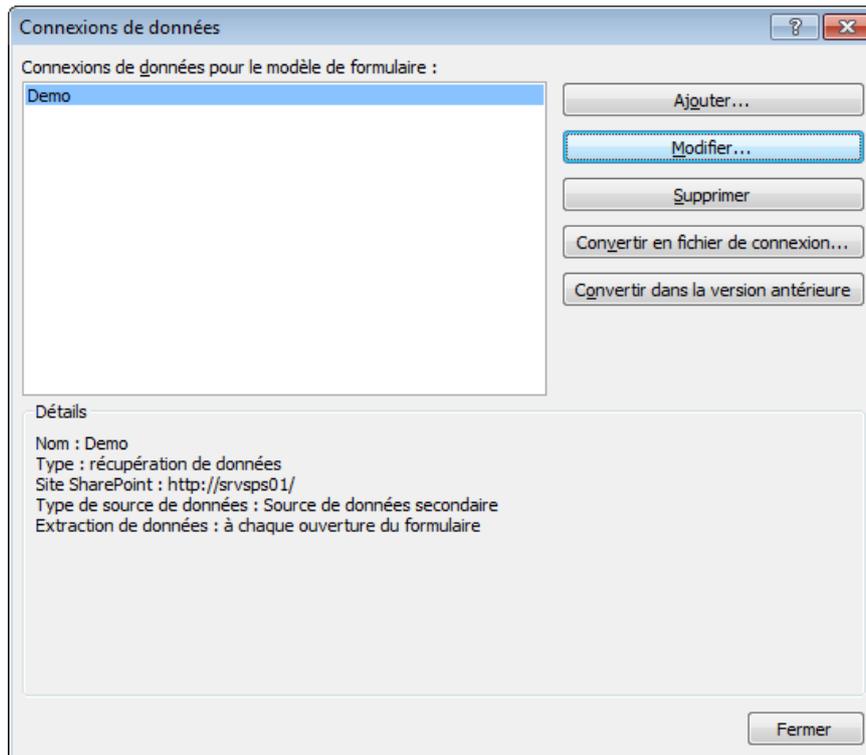
D'abord, nous ouvrons notre formulaire en mode conception:



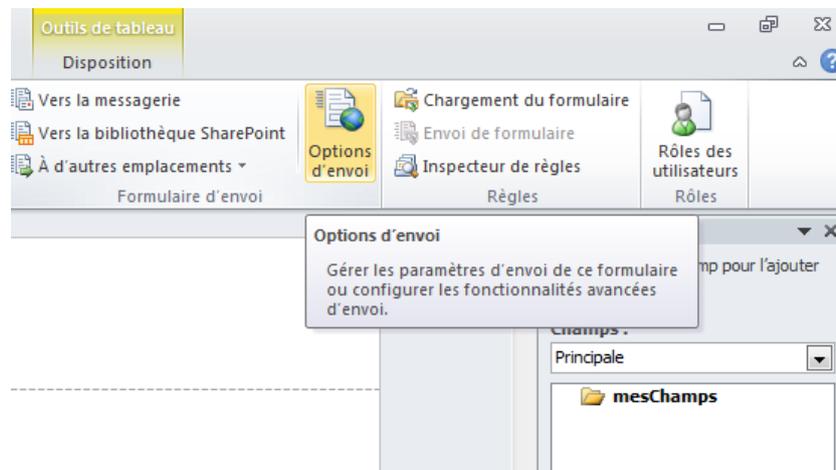
et nous créons comme à l'habitude une *Connexion de données* avec dans le cas présent les colonnes de la librairie SharePoint qui stocke notre modèle de formulaire et nous prenons bien garde à rapatrier l'*ID* (champ unique local qui est pour rappel créé automatiquement par SharePoint lors de la création d'une bibliothèque de fichiers).



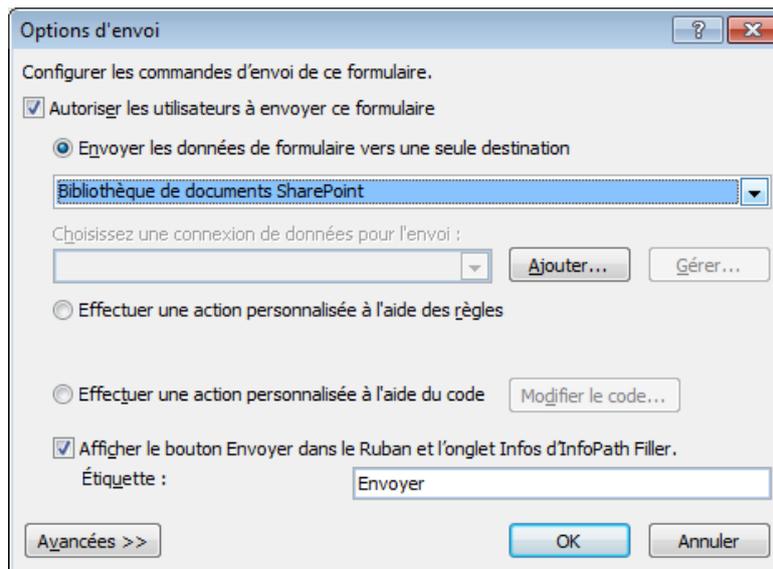
Cela nous donnera la connexion suivante que nous avons appelé *Démon*:



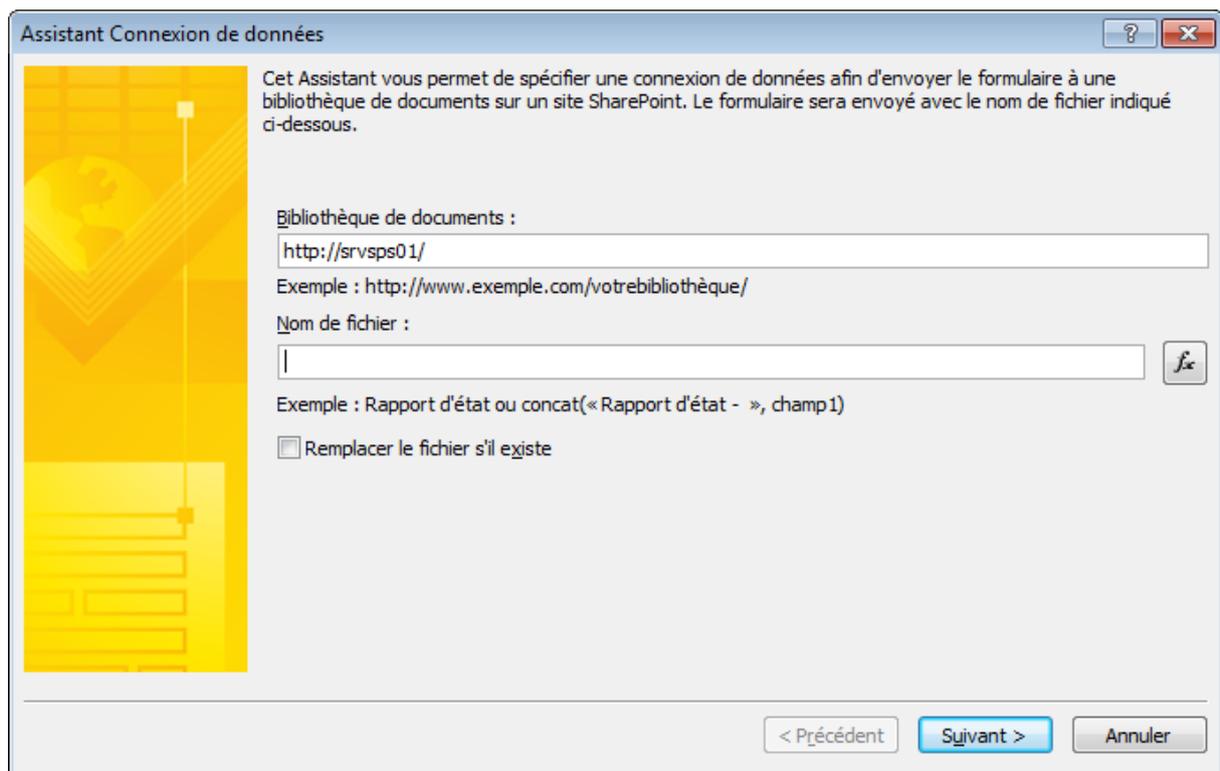
Ensuite, dans le ruban *Données*, nous cliquons sur *Options d'envoi*:



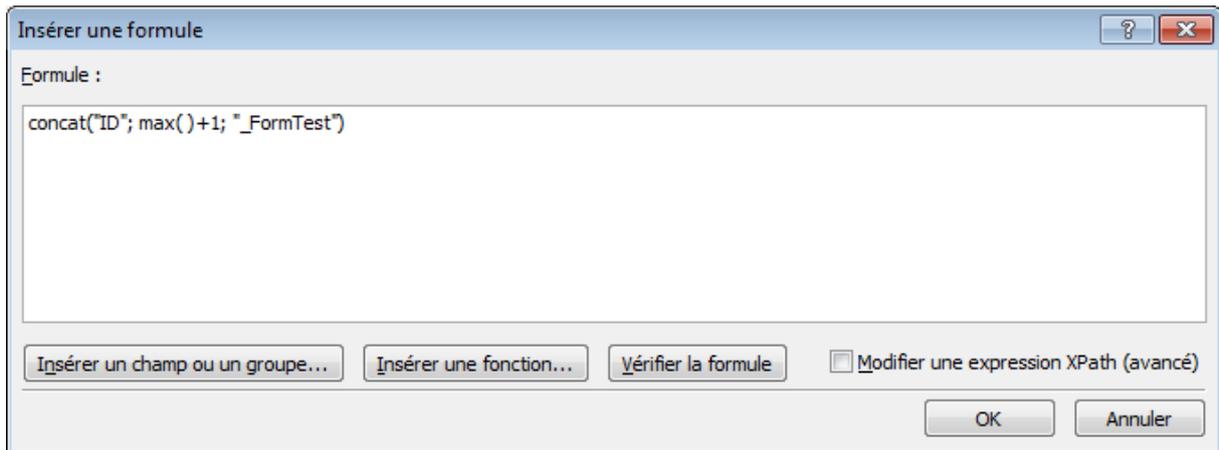
Ce qui nous amène à la boîte de dialogue suivante:



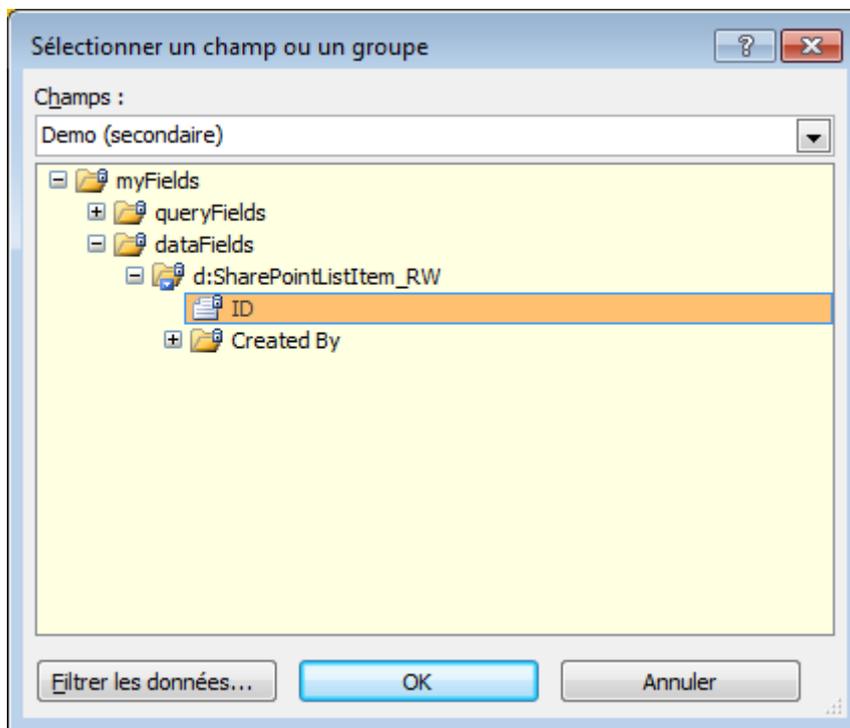
Quand nous cliquons sur *OK* apparaît la boîte de dialogue ci-dessous ou nous entrons l'URL SharePoint de l'emplacement où se trouve la bibliothèque de formulaire où est stocké notre modèle et dans le champ *Nom de fichier*, nous cliquons sur le *fx*:



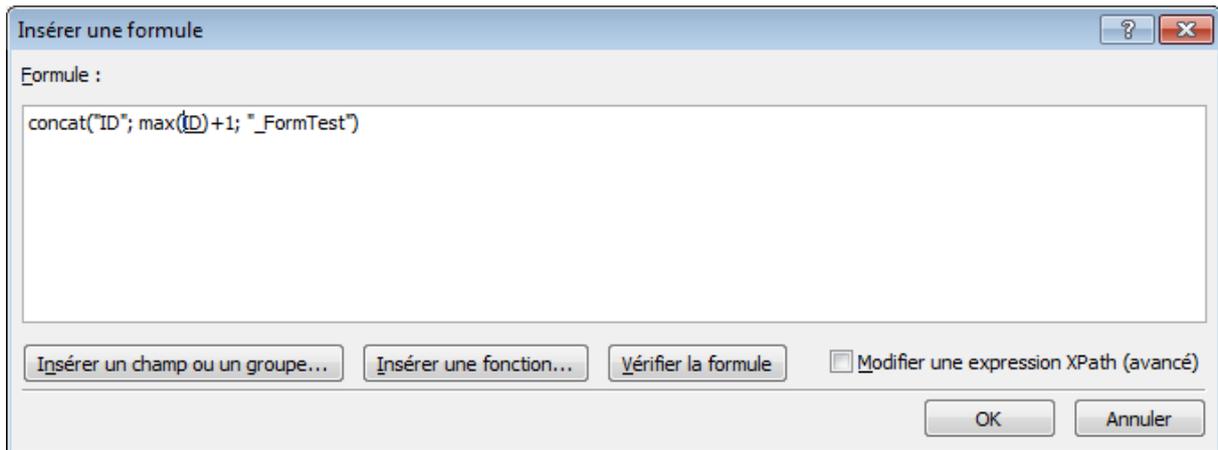
Ce qui nous ouvre la boîte de dialogue suivante de formule dans la quelle nous écrivons la formule visible (bien évidemment libre à vous de la compliquer à loisir!):



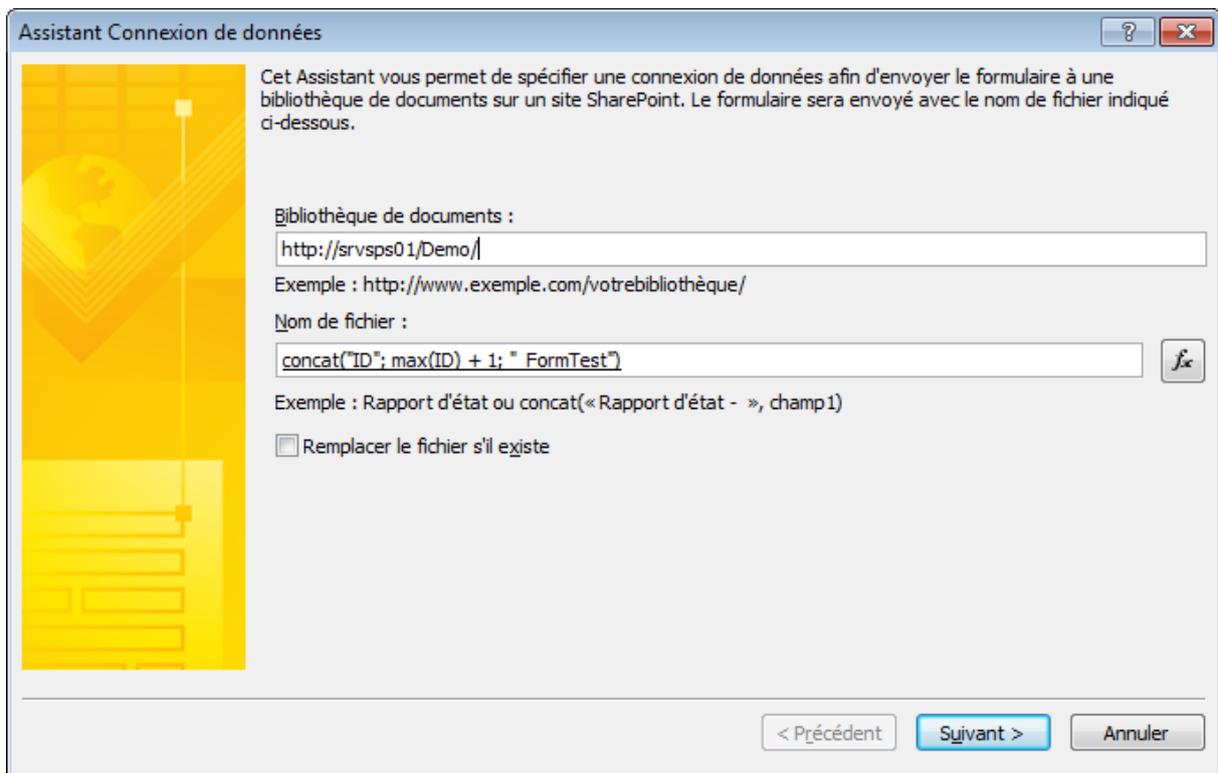
Ensuite, dans la fonction *max()* nous allons chercher en cliquant sur le bouton *Insérer un champ ou un groupe* le noeud *ID* de la source secondaire *Demo* que nous avons créée précédemment:



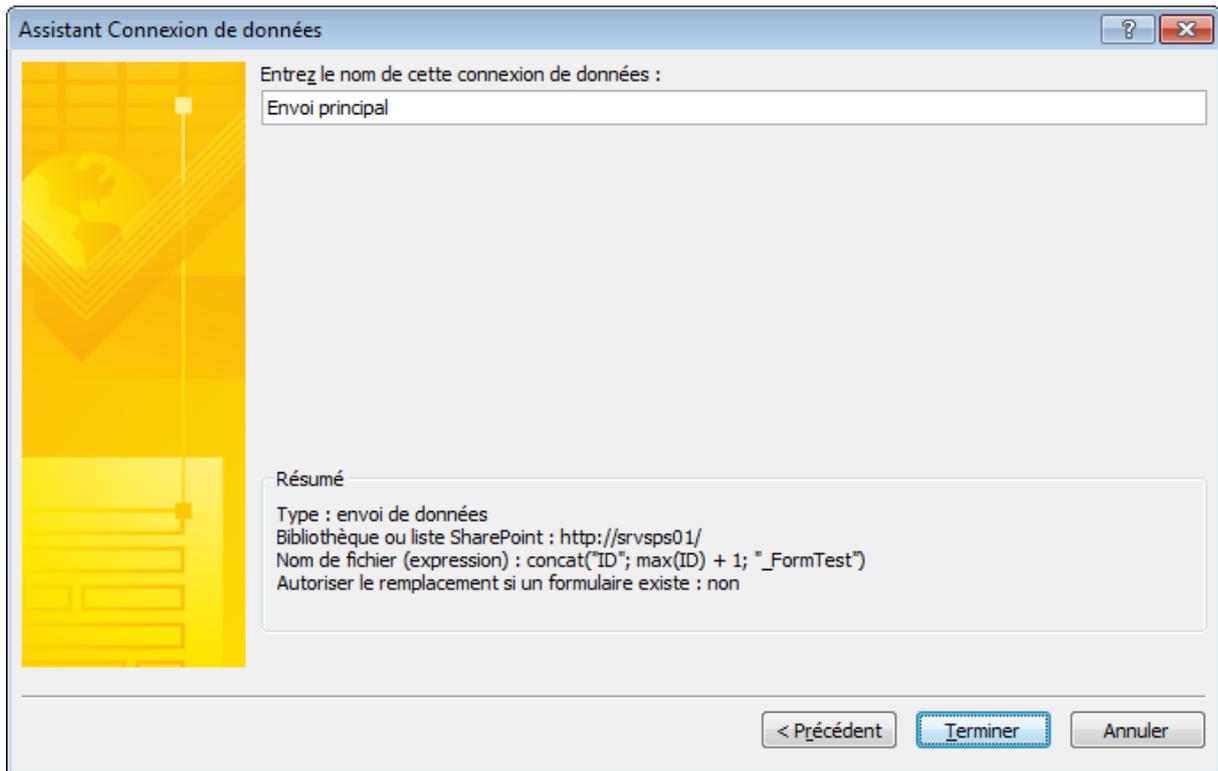
Ce qui donnera:



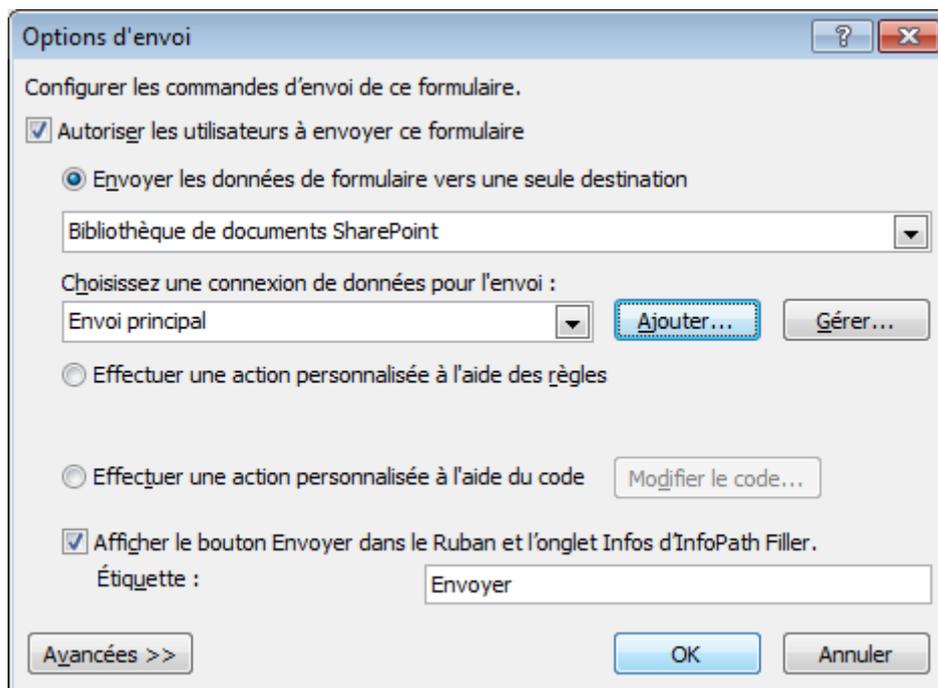
Ensuite, nous validons par *OK* ce qui nous ramène à:



Nous cliquons sur *Suivant*:



et enfin sur *Terminer* pour revenir à notre point de départ:



Il suffit ensuite de valider par *OK* et de faire quelques tests pour obtenir:

<input type="checkbox"/>	Type	Name
		ID2_FormTest NEW
		ID3_FormTest NEW
		IDNaN_FormTest NEW

 Add document

Donc l'incrémentation se fera bien automatiquement. Cependant, seul le tout premier formulaire posera problème (rien ne vous empêchera de le renommer par la suite).

Si ce problème du formulaire à renommer manuellement est inacceptable pour vous, sachez qu'il existe alors plusieurs solutions:

1. Créer un champ masqué dans InfoPath et commencer à bricoler avec des actions et des conditions
2. Créer un workflow dans SharePoint avec SharePoint Designer qui renomme simplement les fichiers automatiquement (très simple à faire)

### **Problèmes connues de cette méthode:**

1. Le formulaire a donc une connexion de données à sa propre bibliothèque et en rapatrie donc au moins le champ *ID*. Ceci devient cependant un problème quand l'on commence à avoir une centaine de milliers de formulaires saisis car la masse d'information qui remonte alors devient conséquente et ralentit de façon inacceptable l'ouverture due formulaire.

Pour éviter cela, on peut alors créer une SharePoint personnalisée par et faire en sorte qu'elle n'affiche que certains des derniers formulaires saisis (mais pas la totalité donc!). Ensuite, connecter la source de données secondaire à cette vue (mais il faudra alors créer une connexion XML vers la vue). Vous diminuerez ainsi considérablement le temps de chargement.

2. La source de données ne renverra avec max(ID) que l'ID le plus grand visible dans la vue par défaut. Donc si cette vue est filtrée par un de vos collègues pendant que certains enregistrent leur formulaire cela va poser des problèmes. Raison pour laquelle il vaut mieux encore une fois connecter la source secondaire à une vue autre que celle par défaut de la librairie (cela diminue le risque sans cependant l'annuler complètement).
3. Si vous avez vraiment beaucoup d'utilisateurs il se peut que deux formulaires soient créés en même temps et donc cela provoquera un message d'erreur pour l'enregistrement de ceux qui ont été un peu lent au niveau de la connexion Internet à stocker le fichier dans le serveur SharePoint. Pour résoudre ce problème il faudra former les utilisateurs à bien réagir.

