

# Comment déployer vos applications professionnelles développées avec Microsoft Access 2013 32/64 bits en incluant le Runtime

Par Jean-Philippe AMBROSINO 

Date de publication : 18 novembre 2013

Dernière mise à jour : 5 novembre 2014

Ce document a pour but de présenter un outil de déploiement de vos applications développées avec Microsoft Access 2013.

Il concerne Microsoft Access 2013 bien que cette fonctionnalité n'ait pas été reconduite par Microsoft et comme pour les autres versions, ce tutoriel abordera l'utilisation du Runtime.

**Commentez**

1 - Avant propos.....	4
1-1 - Mais où est donc l'Assistant Package 2013 ?.....	4
1-2 - Et donc, comment fait-on alors ?.....	4
1-3 - En bref.....	4
1-4 - Remerciements.....	5
1-5 - Contact.....	5
2 - Le développement de votre application (Rappel).....	6
2-1 - Les tests à effectuer avant de générer un fichier ACCDE ou ACCDR.....	7
2-2 - Test de l'application comme si elle était installée sur un poste client.....	8
2-2-1 - .comme s'il était doté de la version complète Microsoft Access 2013.....	8
2-2-2 - .comme s'il était doté du Runtime Access 2013.....	8
2-2-3 - .doté du Runtime Access 2013 conjointement avec une version antérieure de Microsoft Access.....	9
2-3 - Comment et pourquoi convertir votre projet en fichier ACCDE ?.....	10
2-3-1 - Caractéristique d'un ACCDE.....	11
2-3-1-1 - Précisions supplémentaires.....	12
2-3-2 - Comment scinder ma base de données avec Access 2013 ?.....	13
2-3-3 - Création du fichier ACCDE.....	19
2-3-4 - Impossibilité de générer un fichier ACCDE.....	21
2-4 - Sécurisation de votre Projet.....	22
2-4-1 - Comment sécuriser votre application ?.....	22
2-4-2 - Les propriétés de protection d'une application.....	23
3 - Déploiement et empaquetage de votre application.....	25
3-1 - Ambiguïté concernant la génération de packages.....	25
3-2 - Préparation en vue d'exploiter l'application avec le Runtime Microsoft Access 2013.....	25
3-2-1 - Préparation en vue d'exploiter l'application avec la licence Microsoft Access.....	25
3-2-1-1 - Comment obtenir des outils nécessaires au déploiement de solutions Microsoft Access ?.....	25
3-2-1-2 - Installer l'outil Assistant Package sur votre poste.....	26
4 - Présentation de l'Assistant Empaquetage et déploiement.....	27
5 - Déploiement de votre application (avec ou sans le Runtime Access 2013).....	28
5-1 - Comment installer l'Assistant Package 2013.....	28
5-2 - Lancer l'Assistant Package.....	32
5-2-1 - Démarrage de l'Assistant Package.....	33
5-2-2 - Processus de déploiement.....	34
5-2-3 - Étape 1/6 - Choix de la langue et du mode d'empaquetage.....	35
5-2-4 - Étape 1/6 - Création d'un nouveau package.....	35
5-2-5 - Étape 2/6 - Définition de l'application à empaqueter.....	36
5-2-5-1 - Sélection de l'application Access à empaqueter.....	37
5-2-5-2 - Choix du dossier d'installation.....	39
5-2-5-2-1 - Le choix du dossier Racine Autre.....	39
5-2-5-3 - Choix de l'inclusion du Runtime.....	40
5-2-5-4 - Résumé de l'application à empaqueter.....	41
5-2-5-5 - Sélection de l'emplacement du package d'installation.....	41
5-2-6 - Étape 3/6 - Détermination des options de raccourcis de l'application.....	41
5-2-7 - Étape 4/6 - Sélection des fichiers et définition des clés de Registre à ajouter à votre Package.....	43
5-2-7-1 - Le package n'inclut et n'inscrit pas les bibliothèques référencées.....	46
5-2-8 - Étape 5/6 - Propriétés du package et fichiers de l'installateur.....	47
5-2-9 - Étape 6/6 - Fin de la préparation du package.....	50
5-2-9-1 - Sauvegarde du modèle et génération du Package.....	53
5-2-9-2 - Récupération du package (ou non selon l'option cochée).....	54
5-2-9-2-1 - Annulation du processus.....	54
5-2-10 - Vérification du Package généré.....	55
5-2-11 - Le fichier AutoRun pour votre CD-Rom.....	55
5-2-10-1 - Pour concevoir un fichier Autorun manuellement.....	55
6 - Évolution de votre Package.....	57
6-1 - Gestion des packages sauvegardés de l'Assistant.....	57
6-1-1 - Après avoir sélectionné un package, vous pouvez effectuer diverses opérations :.....	57
6-2 - Modification ou évolution de votre application.....	57
6-2-1 - Modification de votre projet.....	58

6-2-2 - Rempaquetage de votre projet modifié.....	59
6-2-3 - Régénération du package après modification ou évolution.....	59
6-3 - Enregistrement de votre Package sur une source d'installation.....	61
7 - Tester son programme d'installation.....	62
7-1 - Pour tester votre programme d'installation depuis un CD-Rom ou un volume.....	62
8 - Exemple d'installation sur un poste client.....	63
8-1 - Installation standard sans Runtime (Access 2013 présent sur le poste).....	63
8-2 - Installation avec Runtime (Access 2013 absent du poste cible).....	70
9 - Les tests à effectuer sur un poste de test.....	82
9-1 - Test de l'application installée sur un poste.....	82
9-1-1 - Poste doté de la version complète de Microsoft Access.....	82
9-1-2 - Poste doté du Runtime Access seul sans aucune version de Microsoft Access.....	82
9-1-3 - Poste doté du Runtime Access conjointement avec une version différente de Microsoft Access.....	82
9-1-4 - .doté d'une version 2010 de Microsoft Office.....	83
9-2 - Erreur d'exécution au cours des tests.....	83
9-2-1 - Quelques exemples :.....	83
9-3 - Création d'un formulaire de démarrage.....	84
9-3-1 - Spécification d'un formulaire de démarrage et des options de démarrage.....	85
9-3-2 - Exemple de formulaire de démarrage.....	85
9-3-2-1 - Génération d'un formulaire de démarrage temporisé.....	86
10 - Installation indépendante du Runtime Microsoft Access sur le poste client.....	87
10-1 - Installation manuelle.....	87
10-2 - Installation automatique du Runtime.....	87
10-3 - Une fois le Runtime installé.....	87
11 - Différences entre Microsoft Access et l'environnement Runtime.....	88
11-1 - Comment se présente le programme Runtime ?.....	89
11-1-1 - Tentative de lancement de Microsoft Access Runtime.....	89
12 - Les erreurs liées à l'installation.....	91
12-1 - Erreur d'exécution 429 lors de l'exécution d'un programme Access.....	91
12-2 - Erreur liée à l'utilisation de MouseWheel.dll.....	91
13 - Conclusion.....	93
14 - Correctifs de la version MSACCESS.EXE 2013.....	94
15 - Téléchargements/Downloads.....	95

## 1 - Avant propos

Ce document a pour objectif de satisfaire aux demandes des développeurs ayant besoin de déployer des applications développées avec Microsoft Access 2013.

Quatre autres tutoriels sont disponibles pour les versions **2000 ou 2002 (XP)**, **2003**, **2007** et **2010** sur le forum.

### 1-1 - Mais où est donc l'Assistant Package 2013 ?

Pour commencer, n'y allons pas par quatre chemins, il n'existe pas d'outil de **Solution de package** pour Microsoft Access 2013.

Et pour cause, la fonction a été tout bonnement supprimée et non reconduite.

Et Microsoft d'annoncer pour compenser, qu'il vous propose d'oublier tout bonnement d'envisager de réaliser des packages comme vous pouviez le faire dans les versions précédentes afin de les distribuer à d'autres utilisateurs.

Cela peut paraître un peu brutal de prime abord, mais Microsoft préfère vous orienter vers une meilleure solution pour distribuer et partager une base de données Access en créant une «**Access App**».

Ainsi, vous pouvez enregistrer une application Access en tant que package à présenter sur **Office App Marketplace** ou dans un catalogue interne d'entreprise. À ce titre, il est tout de même proposé de fournir le Runtime Access 2013 si vous souhaitez partager votre application avec d'autres utilisateurs qui ne disposent pas d'une version complète d'Access 2013.

### 1-2 - Et donc, comment fait-on alors ?

Vous l'avez donc sans doute déjà remarqué, la mise en œuvre et le mode de déploiement des applications Access 2013 est inexistant.

Et grâce à ce tutoriel, vous allez pouvoir découvrir un outil assez similaire aux versions précédentes.

Vous verrez qu'il subsiste certaines similitudes - *on ne peut pas réinventer la roue* - et j'y ai mêlé un peu de 2003 et un peu de 2010 avec quelques fonctionnalités supplémentaires afin que vous puissiez bénéficier d'un assistant d'emballage au moins aussi pratique que les précédents.

Eh oui, sans doute pris d'un élan de compassion à l'égard des développeurs habitués à déployer leurs applications comme à l'ancienne mode d'une part, puis considérant que seule la version 2013 est disponible à l'achat de Microsoft Office d'autre part et qu'enfin et par ailleurs, les entreprises sont de très loin toutes prêtes à souscrire un «**Abonnement Office App Marketplace** », je me suis dit que de développer cet outil pourrait être à la fois un sacré défi et sans doute une bonne idée.

Ce sera à vous de me le faire savoir.

Et donc, je me suis lancé.

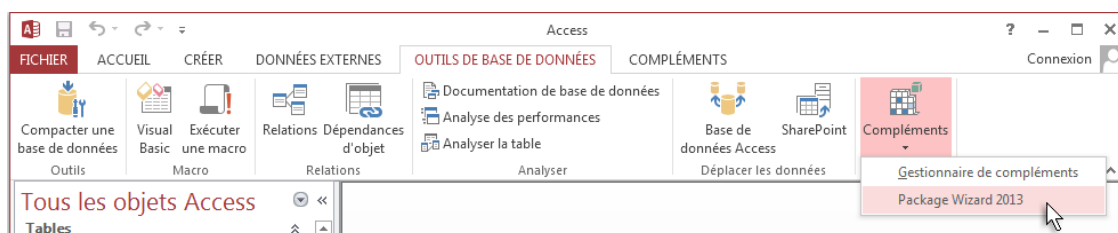
### 1-3 - En bref

À l'instar de la version 2010, j'avais mis en œuvre la possibilité de faire en sorte que l'emballage de vos applications puisse être généré depuis la **Vue Backstage de Microsoft Access** mais malheureusement, bien que fonctionnelle, cette initiative resta vaine puisque les compléments Access, par je ne sais quelle raison - si ce n'est le bug qu'on m'a remonté - ne savent pas afficher une vue *Backstage*.



*Menu Backstage*

Du coup, je me suis rabattu sur la seule alternative qui m'est offerte, à savoir, depuis le Ruban via l'onglet **Outils de base de données.**, puis le groupe **Complément** où vous choisirez l'option «**Package Wizard 2013**».



*Onglet Outils de base de données*

## 1-4 - Remerciements

Je tiens à remercier tout particulièrement toutes celles et ceux qui ont participé à la relecture de ce document en y incluant leurs remarques.

Un remerciement tout particulier à **Fabien (f-leb)** pour la relecture.

## 1-5 - Contact

Pour tous renseignements complémentaires (*non techniques*), veuillez me contacter directement (**Argyronet**) par MP. Sinon, merci de passer par le forum.

## 2 - Le développement de votre application (Rappel)

Lorsqu'il vous est demandé de développer une application Access, la difficulté à laquelle vous risquez d'être confronté est la mise à disposition de celle-ci sur un ou plusieurs postes cibles. Au sein de l'entreprise dans ou pour laquelle vous travaillez, vous avez la responsabilité d'un ou plusieurs projets et de fait, vous participerez de près ou de loin à la mise en œuvre de tous les paramètres pour que vos applications soient opérationnelles.

Les informations contenues dans ce document considèrent le cas suivant d'une part et excluent le choix de Microsoft Office Marketplace d'autre part :

- 1 1. Vous développez une application à partir de Microsoft Access 2013 ;
- 2 2. Vous avez installé le *Package Wizard 2013* qui est disponible [ici](#) ;
- 3 3. Les postes clients possèdent ou ne possèdent pas une licence Access 2013 installée ;
- 4 4. Vous travaillez sur une plate-forme Windows Seven SP1 ou ultérieure.

De nombreuses phases de test sont nécessaires avant d'envisager la mise en place sur les postes clients. Ainsi, vous devez être à même de considérer trois cas de figure en ce qui concerne les applications :

### Les applications simples

C'est ce que j'appelle des petites applications de base de données où seul est prédominant le traitement des données. Les formulaires sont pratiques à utiliser, mais sans look particulier et sans grande convivialité (comme « Northwind » exemple fourni en standard), des états lisibles contenant les informations attendues et l'ensemble est géré de façon semi-automatique avec quelques macros et pas ou peu de routines VBA.

L'utilisateur sera en face d'une interface un peu plus riche qu'avec les versions précédentes par la présence du **Ruban Microsoft Office** et du **système de navigation** grâce aux onglets et au **Volet**. Par ailleurs, les opérations courantes se limiteront à saisir manuellement un grand nombre de données pour aboutir à ses fins. Le tout fonctionnera en partie avec l'utilisation du Ruban sur l'interface Access en arrière-plan.

L'utilisateur pourra au gré de ses besoins, intervenir pour modifier ou créer des requêtes par exemple ou d'autres états à la condition que vous ne distribuiez pas un fichier compilé (*ACCDE*).

### Les applications avancées

C'est ce que j'appelle des applications semi-professionnelles de bases de données où sont prédominants le côté convivial et « *user friendly* » de votre application. Le développeur aura conçu un projet propre, doté de fenêtres agréables, inclus des boutons pour automatiser les tâches et créé un Ruban personnalisé pour chaque formulaire.

L'utilisateur bénéficiera d'un confort d'utilisation très agréable avec de nombreux automatismes, des vérifications systématiques des entrées, des contrôles auto-activés en fonction des types de données saisis.

Les états auront un rendu très soigné, les ruptures des regroupements auront été vérifiées, etc.

L'Application ne contiendra pas ou peu de macros et aucune pour les traitements des données.

Toute l'application sera exempte d'erreur.

### Les applications professionnelles

Elles restent idéales et représentent le nirvana de l'application avancée. Elles sont dotées des caractéristiques totalement similaires aux précédentes mais seront développées avec une finition irréprochable.

C'est une approche délicate, car le développeur doit oublier Access et considérer que son application est totalement indépendante.

Considérant la nouvelle interface de Microsoft Access 2013, le développeur se doit « *d'oublier Access* » et pour n'en détailler que quelques-uns :

- Supprimer et remplacer les anciennes commandes de menu (**DoMenuItem**) même si elles restent quelques fois compatibles (*Cas où la base de données est issu d'un (très) ancien format MDB*) ;
- Exploiter totalement le **Ruban Microsoft Office** et non partiellement, e.g. en générant un nouveau **Ruban personnalisé (Voir Tutoriel de Tofalu à ce sujet)** ;
- Désactiver les fonctionnalités contextuelles ;
- Masquer le **Volet de navigation** ;

- Dans l'absolu, cacher l'instance Access pour ne laisser apparaître que l'application elle-même est un must auquel je me prêtais systématiquement dans les versions précédentes. Avec cette nouvelle version et son **Ruban**, je serais tenté de dire que ce n'est plus un must dans le sens où si l'on souhaite exploiter le ruban et donner un look Office 2013 à votre projet, vous ne pouvez pas envisager ce choix.

**En remplacement des DoMenuItem.** La méthode **RunCommand** de l'objet **DoCmd** a été introduite dans Microsoft Access dès la version 97, en remplacement de la méthode **DoMenuItem** des versions antérieures. Elle permettait d'exécuter une commande intégrée de menus ou de barres d'outils. Pour chacune des commandes de menus et de barres d'outils de Microsoft Access, il existe une constante correspondante. Vous pouvez utiliser celles-ci avec la méthode **RunCommand** à l'aide de Visual Basic. **Tout est expliqué ici** dans le tutoriel de **Domi2**.

Donc, quel que soit le type d'application que vous développerez, il est envisageable que pour leur distribution, vous soyez tenu de fournir un programme d'installation automatisé.

Il est vrai que pour le cas des **applications simples**, il est considéré que l'utilisateur possède une version complète de Microsoft Access 2013, auquel cas la fourniture du fichier **ACCDB** (ou **ACCDE** ou encore **ACCDR**) par le biais d'un fichier d'installation automatisé peut paraître superflue. En fait, si vous n'avez aucun composant ActiveX externe (OCX) ou aucune référence à des bibliothèques particulières (DLL, TLB) et que tous les postes cibles (*utilisateurs*) sont équipés d'une version complète d'Access 2013, il suffit de distribuer simplement par copie, les fichiers (*accd\**).

Si votre application, quelle que soit la catégorie dans laquelle elle se range, doit être distribuée dans différents services avec pourquoi pas la notion du multilingue, qu'elle utilise des composants spécifiques externes, vous devez faire en sorte que tout soit fourni avec votre projet pour qu'il puisse être déployé et que les composants soient inscrits dans la Base de Registre de Windows du poste client si cela s'avère nécessaire (*composants COM*).

**!** Notez que pour ce dernier cas et en particulier pour Windows Vista et ultérieur, **les droits d'administration sont requis**.

Donc, si vous envisagez de distribuer vos applications avec le **Runtime Microsoft Access 2013**, l'outil de *Solution de Package* que j'ai nommé dans la version 2013 **Assistant Package...** (ou **Package Wizard...** selon la langue) est là pour vous y aider.

**Rappel :** Si vous souhaitez emballer votre application à l'aide de ce nouvel outil **Assistant Package**, vous devez d'abord l'installer sur votre poste de travail où Microsoft Windows 7 et Microsoft Office 2013 sont installés (sous licence). Cette procédure est vue plus loin à la **section 5**.

**i** => Notez au passage que cette application est totalement gratuite et nécessite que vous la téléchargez avant son installation.

=> Il n'est plus question non plus d'acquiescer une suite de composants particuliers pour en bénéficier.

Par ailleurs, l'utilisation ou la redistribution de Microsoft Access Runtime 2013 est gratuite et le nombre d'utilisateurs auquel vous pouvez distribuer ce programme n'est pas limité.

Toutes les informations à ce sujet sont abordées dans ce tutoriel.

## 2-1 - Les tests à effectuer avant de générer un fichier ACCDE ou ACCDR

Une fois le développement terminé, vous devez lancer une batterie de tests afin d'éliminer toutes les erreurs potentielles.

Le cas typique est souvent l'omission ou celle d'une macro **AutoExec** ou encore l'omission de certaines références vers des objets externes.

- Notez que dans bien des cas il est préférable de faire du **Late Binding** plutôt que du **Early Binding** (*liaison anticipée et liaison tardive en Français*) afin de garantir la compatibilité.

En d'autres termes, il est préférable d'utiliser de la fonction

```
CreateObject ("ApplicationName.Application")
```



plutôt que de la définition

```
Set Expression = New ApplicationName.Application
```

Cependant, si votre application est pleinement 2013 et vouée à être distribuée sur des plateformes Office 2013, conservez le **Early Binding**.

**i** Notez que la différence entre *Late Binding* et *Early Binding* s'effectue davantage sur la déclaration **As Object** que sur la méthode de création de l'objet par le mot clé **New**. Par ailleurs, il est possible d'utiliser la fonction **CreateObject** pour exploiter la technologie *Early Binding*, et Microsoft le préconise même parfois.  
(Merci à **Arkham46** pour ce complément d'information)

## 2-2 - Test de l'application comme si elle était installée sur un poste client

Le jeu de tests consiste à exécuter l'application dans une situation identique à celle dans laquelle elle se trouvera après déploiement sur un poste.

### 2-2-1 - .comme s'il était doté de la version complète Microsoft Access 2013

Il ne devrait pas y avoir de grande différence avec le poste de développement depuis lequel vous avez créé l'application.

Le jeu de tests consistera alors à naviguer entre les différents formulaires, l'impression des états (*attention à l'orientation des états et aussi à la notion d'imprimante par défaut*), à l'exécution de l'ensemble des tâches proposées par votre application.

### 2-2-2 - .comme s'il était doté du Runtime Access 2013

Du fait que certaines fonctions Microsoft Access standards sont masquées ou désactivées dans **l'environnement Runtime**, vous devez vous assurer que l'application fonctionne correctement dans cet environnement avant de la distribuer.

Pour simuler **l'environnement Runtime** pour votre **application 2013**, il suffit de renommer votre fichier avec une extension **ACCDR** :

- dans le cas du projet abordé ici, l'application de démonstration a été créée pour l'exemple.
- pour simuler son usage en *environnement Runtime*, son nom deviendra alors **Mon application.ACCDR**

**i** Pour obtenir de nouveau une version non verrouillée de votre application, il suffira de la renommer de nouveau avec son extension standard (si bien entendu elle est au format 2007-2013).


Ici aussi donc, vous procéderez aux mêmes jeux de test à effectuer que celles qui sont stipulées en section **2-2-1 ci-avant**.

Dans tous les cas, n'oubliez pas de tester également tout ce qui est procédure **d'export vers Excel** ou tout ce qui est génération de **documents Word** via Automation sans oublier la notion de **droits d'accès** du fait que vous, en tant que développeur, vous êtes certainement administrateur de votre poste de travail, ce qui ouvre des autorisations que les autres n'ont pas forcément.



## 2-2-3 - .doté du Runtime Access 2013 conjointement avec une version antérieure de Microsoft Access


Considérez ici (*bien que ce soit déconseillé*) que le poste client dispose d'une version de **Microsoft Access antérieure** (par exemple 2003) à celle du **Runtime** qui, lui, correspond à la **version 2013 de Microsoft Access**. Dans ce cas, vous devrez mettre en place une syntaxe particulière pour le raccourci que vous aurez à créer et surtout de procéder à des tests sévères, car la cohabitation de deux versions si opposées dans le temps vous expose à de probables dysfonctionnements et des erreurs irrécupérables.

 *De manière générale, je déconseille la cohabitation de versions différentes sur un même poste de travail.*

Les bases de données et applications Microsoft Access 2013 possèdent la même extension que les versions 2007 et 2010 (*ACCDB, ACCDE, ACCDR*). Donc, il n'est pas nécessaire d'avoir à créer des raccourcis spécifiques.

Pour autant, le fait que vous installez une autre version de Microsoft Access représentée ici par le Runtime lors de l'installation de votre application à déployer, celle-ci prend la main sur tous les types de fichiers reconnus comme des bases données Microsoft Access, à savoir :

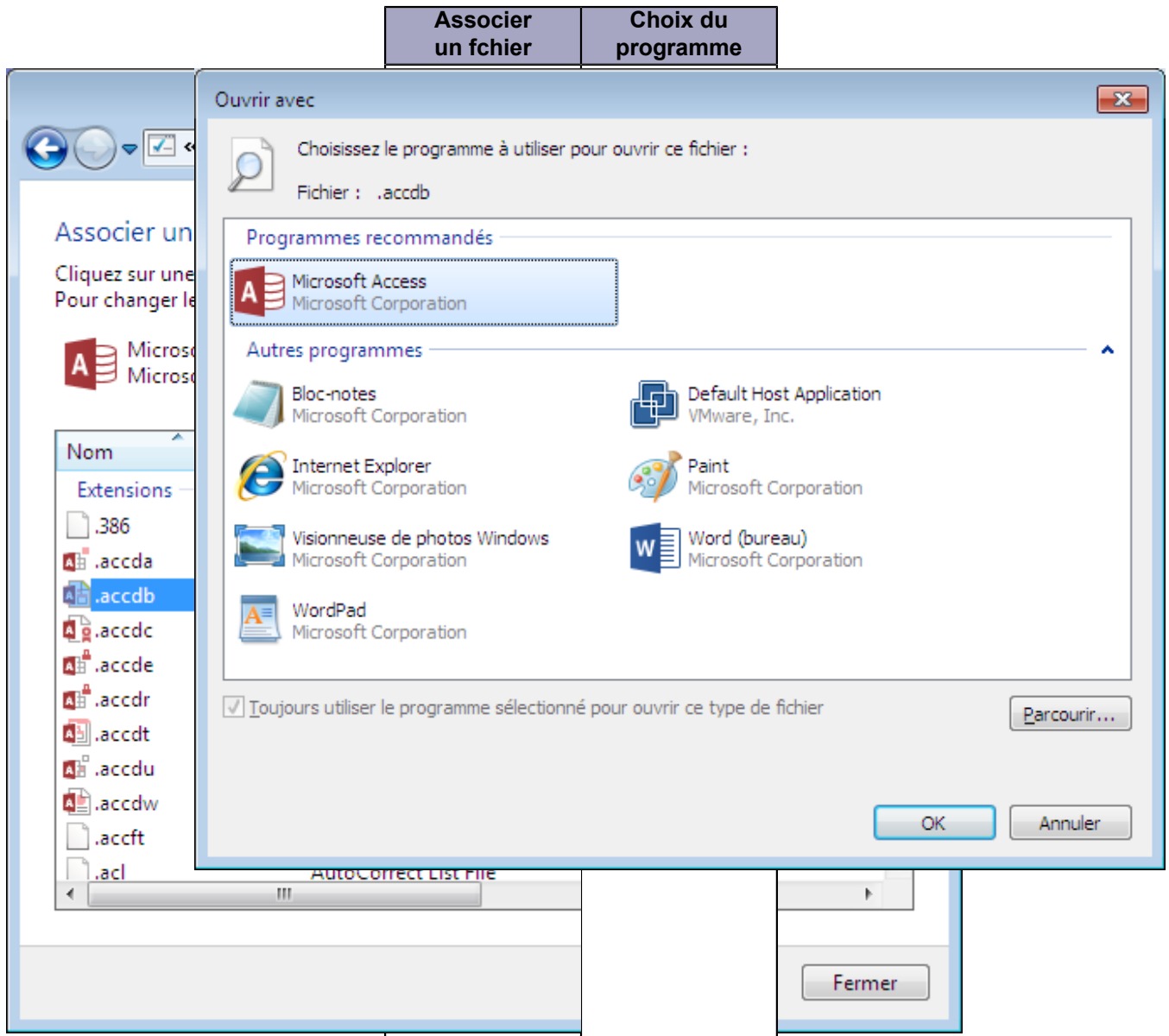
- MDB / MDE / MDA ;
- ACCDB / ACCDE / ACCDR / ACCDA / ACCDT / ACCDC ;
- pour ce qui est des ADP / ADE, ils ne sont plus supportés.

 *L'avantage de ces différences entre les extensions 2013 et les extensions des versions antérieures est que l'on peut différencier le programme à ouvrir selon le type de fichier sans avoir à créer forcément un raccourci.*

Dans la mesure où l'utilisateur du poste sur lequel vous installez votre application 2013 avec le Runtime possède une version antérieure, vous pourrez lui apprendre à modifier les associations de fichiers depuis **l'Explorateur de fichiers** à partir de la fonctionnalité des **Programmes par défaut** du **Panneau de configuration**.

[MODIFIER LES ASSOCIATIONS POUR MICROSOFT ACCESS 2013](#)

Veillez vous reporter à la même section dans [le tutoriel de déploiement 2010](#)



### INSTALLER SUR UN POSTE VIRTUEL

Si vous ne disposez pas d'un poste de test, vous pouvez alors envisager d'installer sur votre propre poste une **machine virtuelle**.

Vous avez le choix entre :

- **VMWare®**

ou

- **VirtualBox VMs**

Cela vous permettra toutes les fantaisies en matière de simulation.

Vous devrez alors installer une version 2013 de **Microsoft Office** sur ce poste virtuel pour installer ensuite votre **application** par l'intermédiaire du package venant d'être généré.

Vous procéderez alors aux essais nécessaires et aux modifications à apporter en conséquence de ce qui vient d'être cité ici.

## 2-3 - Comment et pourquoi convertir votre projet en fichier ACCDE ?

Un fichier ACCDE est à Microsoft Access 2013 ce que le fichier MDE était aux versions précédentes de Microsoft Access. À partir du moment où votre application Microsoft Access comporte du code Visual Basic, il est conseillé de l'enregistrer comme fichier ACCDE pour, en quelque sorte, verrouiller votre projet.

### Attention

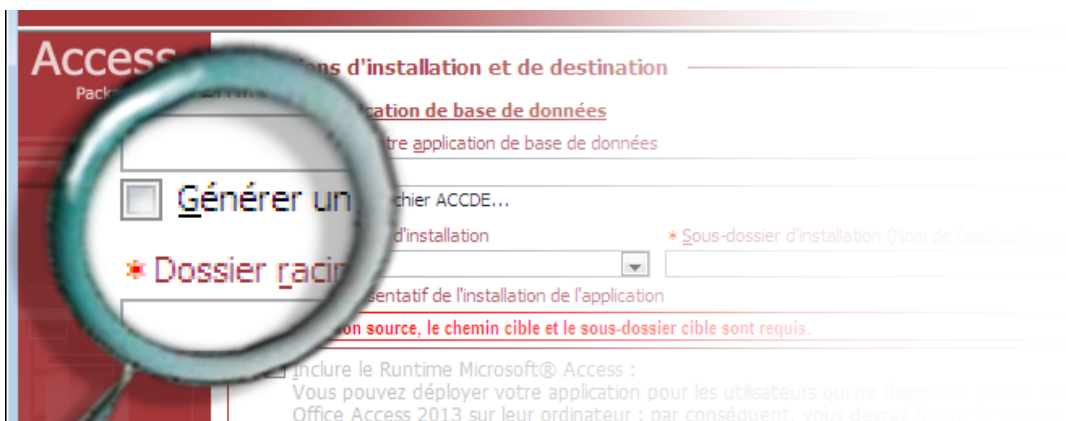
- ⚠ **Rappelez-vous qu'il existe une incompatibilité entre les versions 32bits et 64bits des logiciels Microsoft Access et donc, une base convertie en ACCDE depuis une version 32bits ne sera pas compatible avec une version 64bits, et inversement.**

Cette opération permet de compiler tous les modules et d'empêcher quiconque d'intervenir sur votre code source, celui-ci étant masqué ou de modifier la structure des formulaires et des états.

Parallèlement à cela, la base de données occupera moins de place sur le disque dur et le fait que le code soit compilé va augmenter les performances de l'ensemble.

- 💡 **L'outil *Assistant Package* présenté ici vous propose, en cochant la case appropriée, la création d'un fichier ACCDE à partir du moment où vous avez sélectionné un fichier de base de données ACCDB.**

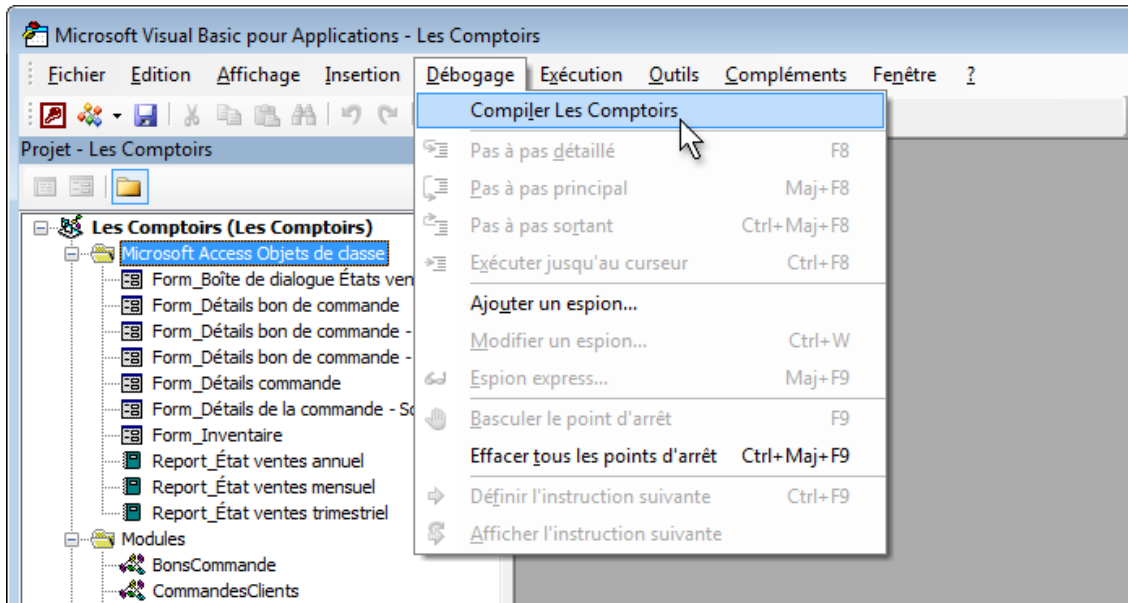
*Vous verrez cette étape plus loin dans ce tutoriel.*



## 2-3-1 - Caractéristique d'un ACCDE

- Modification, création de formulaires, d'états ou de modules désactivés ;
- Modification, ajout ou suppression des références aux bibliothèques d'objets ou aux bases de données interdites avec bien entendu, l'impossibilité d'intervenir sur le code à l'aide des propriétés ou des méthodes de Microsoft Access ;
- L'importation ou l'exportation d'objets tels que formulaires, états ou modules est impossible ;
- Seules les tables ou les requêtes peuvent être exportées ou importées vers ou depuis ce type de bases de données.

- ⚠ **Prenez en compte qu'avant de créer un ACCDE, vous devez compiler toujours votre projet à partir du menu *Débogage de VBE*.**




En parallèle à cela, conservez systématiquement une copie de sauvegarde de votre fichier **ACCDB** après avoir procédé à la conversion en **ACCDE** et ce par sécurité. Il est en effet **impossible** d'inverser la compilation d'un **ACCDB** en **ACCDE**, c'est-à-dire, de récupérer un fichier **.ACCDB** à partir d'un fichier **.ACCDE** !

### 2-3-1-1 - Précisions supplémentaires

Dans la plupart des cas, il est fortement recommandé de scinder votre application en **deux bases distinctes** à savoir une application frontale qui est le programme utilisateur et l'autre qui représente les données. Que ce soit en utilisation locale ou en réseau, vous établirez une liaison entre l'application frontale et le fichier de la base dorsale **ACCDB** contenant les tables.


#### **Vous pouvez d'interdire l'accès à la base de données contenant les tables.**

 En générant une version **ACCDR** de votre base de données, vous imposerez à quiconque d'ouvrir celle-ci en **mode d'exécution**.

Pour ce faire, modifiez simplement l'extension du fichier de base de données **.ACCDB** par **.ACCDR**. Vous obtiendrez alors une version verrouillée de votre base de données Access 2013.

#### **ACCDR n'est pas une protection !**

Notez que la modification de l'extension n'est pas une protection !


 N'importe qui connaissant «le truc» peut changer l'extension afin de pouvoir accéder aux tables (dans ce cas précis).

Une protection plus efficace reste bien entendu le mot de passe

Si l'utilisation est prévue pour être en réseau, vous poserez alors la base de données (*dorsale*) contenant les tables dans un dossier dédié sur le serveur et établirez la liaison avec l'application frontale depuis cette ressource réseau.

**Les liaisons sont inscrites en dur dans l'application frontale.**

Prenez donc en compte qu'il est fortement recommandé de générer le fichier **ACCDE** avec les liaisons réelles des tables comme si vous étiez sur le poste client sans quoi, aucune donnée ne sera disponible au moment de l'installation et l'erreur 3044 sera levée.

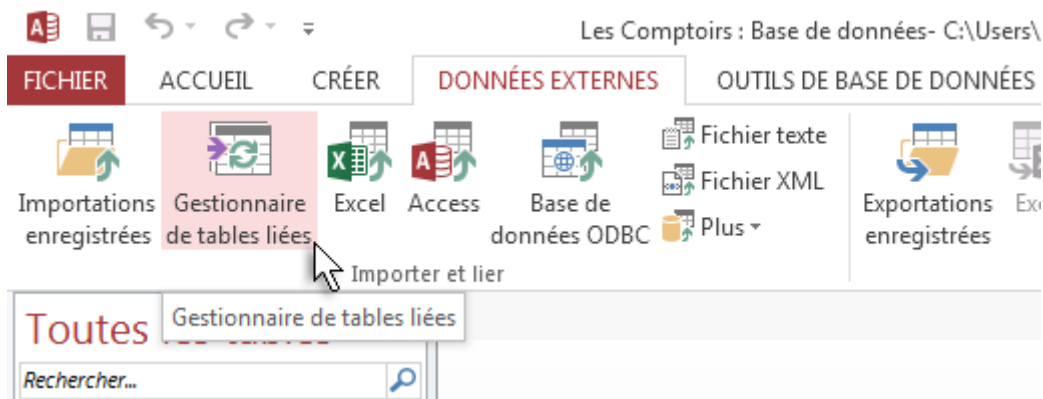
 J'entends par **liaisons réelles** les liaisons **UNC** (\\Server\Shared Path) et non pas celles définies avec un lecteur mappé comme **H :**, **P :** ou encore **U :** lettres communément employées dans les entreprises.

*Dans les situations professionnelles dans lesquelles vous pouvez être confronté, employez systématiquement la sélection des liaisons avec les conventions **UNC** même si les accès aux serveurs sont mappés, auquel cas, recopiez le nom :*

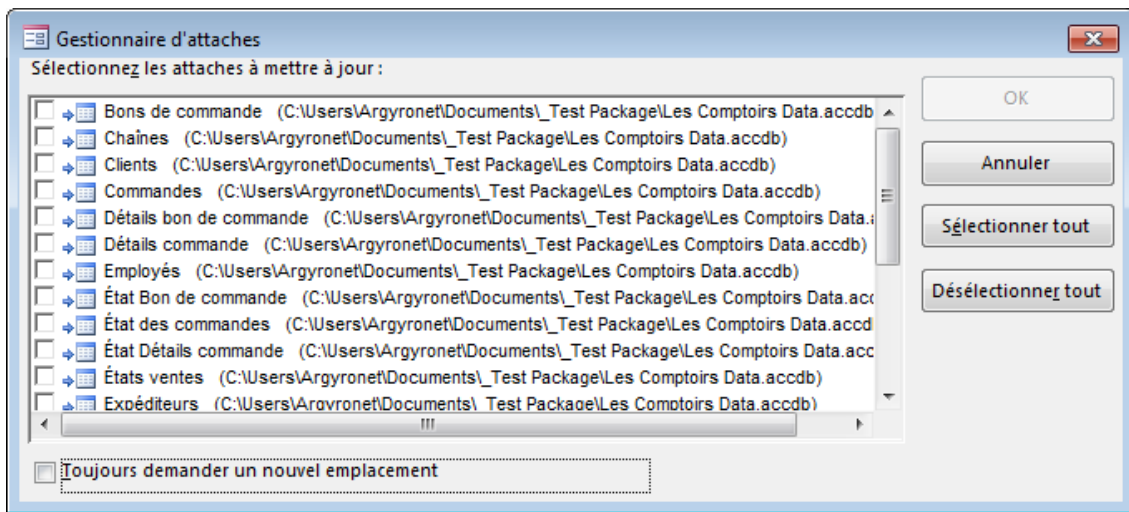
- soit depuis l'**Explorateur**,
- soit à l'aide d'une fonction maison qui renvoie le chemin UNC du lecteur passé en paramètre.

### Liaisons des tables Microsoft Access.

Pour afficher ou modifier les liaisons de votre Base de données, il vous suffit d'activer le **Gestionnaire de tables liées** depuis le bouton du même nom dans l'onglet **Outils de base de données**.



La boîte de dialogue fera aussitôt son apparition :



La distribution d'une application dont les données sont liées nécessite que vous preniez la précaution de travailler dans un contexte identique en mode développement.

Dans mon cas ici, par exemple, je choisirais d'installer mon projet dans un sous-dossier (portant le nom de mon **Application**) de **Mes documents**, et ferais en sorte que ma **Base de données dorsale** ne contenant que les tables se trouve dans un sous-dossier nommé par exemple **Data**.

J'y ferais alors référence par un rattachement dynamique ce qui fait que les liaisons ne seront jamais rompues.

### 2-3-2 - Comment scinder ma base de données avec Access 2013 ?

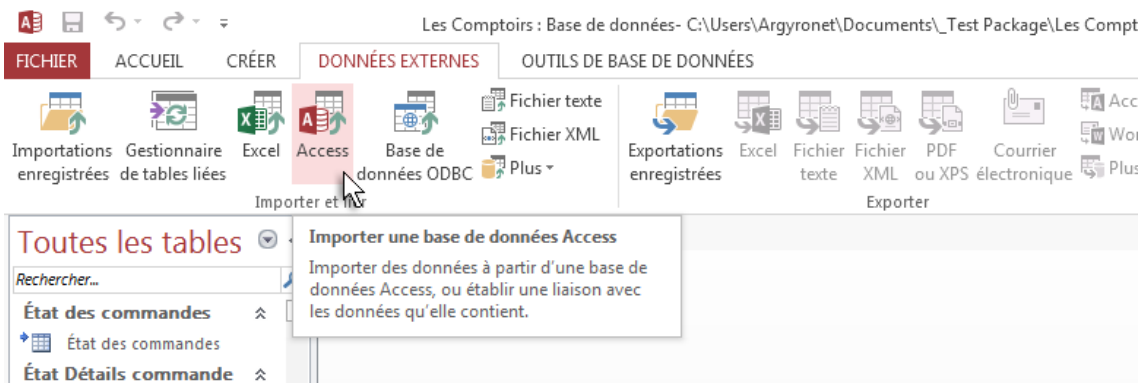
Pour pouvoir scinder votre base de données en deux bases distinctes dont l'une ne contient que les tables et l'autre tous les autres objets de l'application, vous pouvez procéder de la façon suivante :

#### **Méthode manuelle :**

- effectuez [au préalable] une copie de votre application ;
- quittez votre application ;
- dans le dossier où sera stockée votre application, créez une base de données vide (par exemple : C:\Users\Argyronet\Documents\Data\DBMonApplication.accdb) ;
- nommez cette base avec un préfixe vous permettant de différencier les deux bases de données (par exemple *DB ou Data*) ;
- depuis cette base de données vierge, importez toutes les tables de votre application actuelle exceptées celles devant rester locales ;
- refermez la base de données puis rouvrez votre application.

Vous pouvez alors supprimer toutes les tables **locales** de votre application que vous avez exportées après vous être assuré de leur intégrité.

Cliquez ensuite sur le bouton **Access** de l'onglet **Données externes** pour pouvoir établir la liaison avec l'autre base de données.

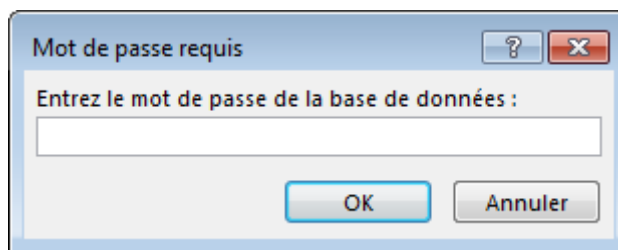


**Rappel :**

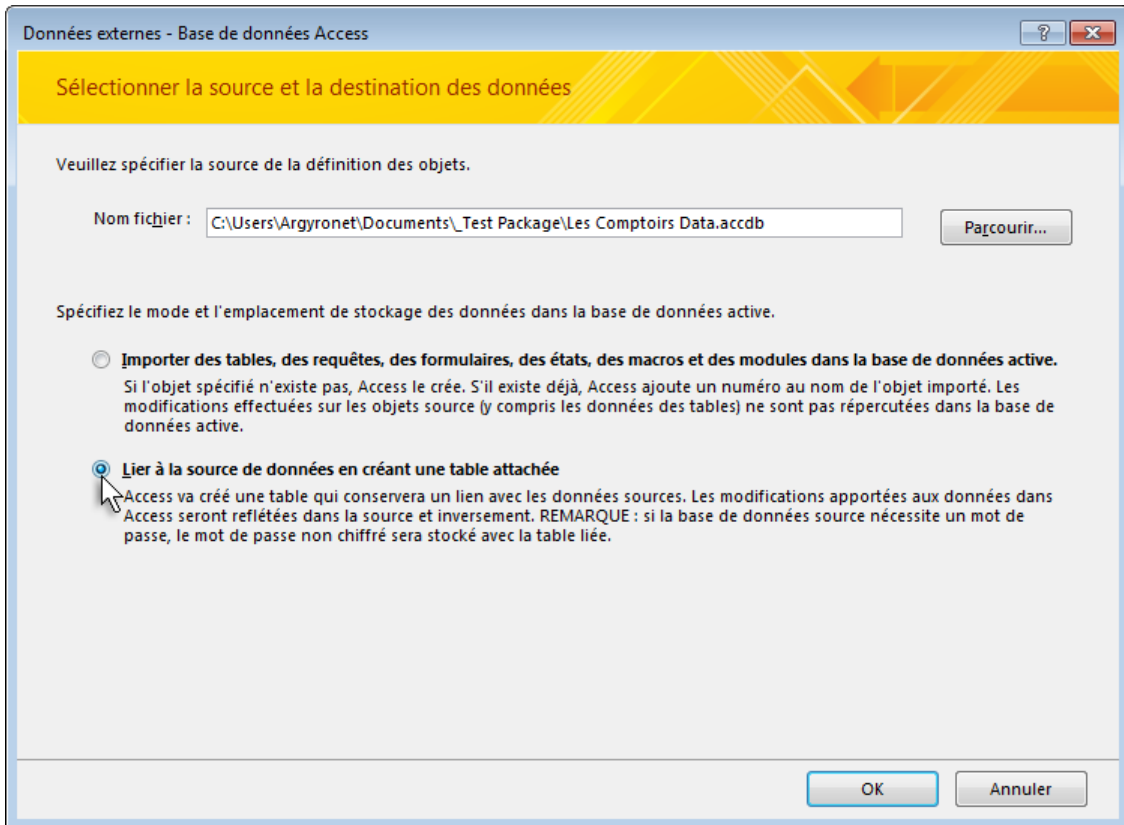
- Si votre base de données dorsale est présente sur un serveur, il vous appartient de définir les liaisons avec les **conventions UNC** comme stipulé **ci-avant**.  
Très souvent, vous êtes tenté de sélectionner le lecteur qui est mappé sur votre propre machine.

Dans la boîte de dialogue qui vous permet de sélectionner la nouvelle source de données, cliquez sur le bouton **Parcourir** et sélectionnez la base de données que vous venez de créer. Dans l'exemple de ce tutoriel, le dossier concerné est le sous-dossier **Data**.

- **!** Si la base de données possède un mot de passe, il vous sera demandé avant de pouvoir établir la liaison.



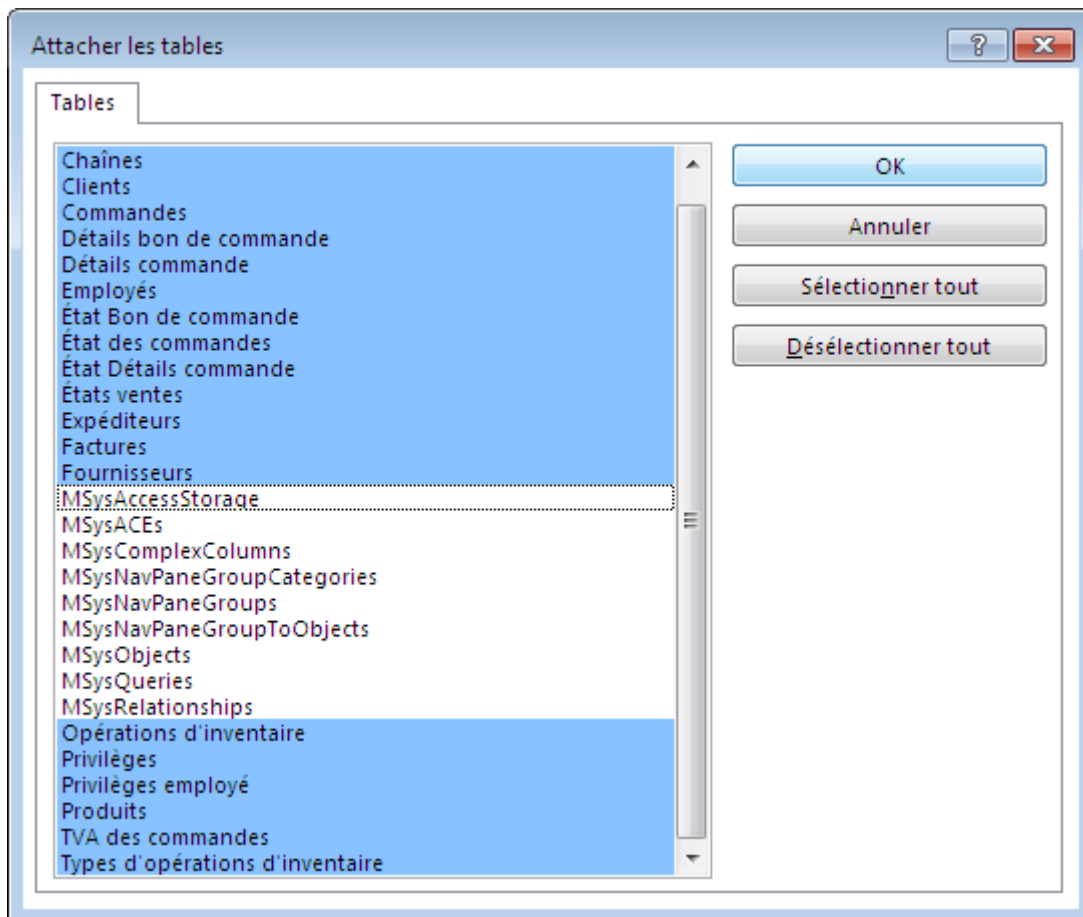
Sans ce mot de passe, la liaison ne sera pas effectuée. Le chemin complet et le nom de fichier doivent apparaître dans la zone de texte de cette même boîte de dialogue. Vous pouvez alors cocher l'option appropriée nommée **"Lier à la source de données en créant une table attachée"**, puis cliquez sur le bouton **OK**.



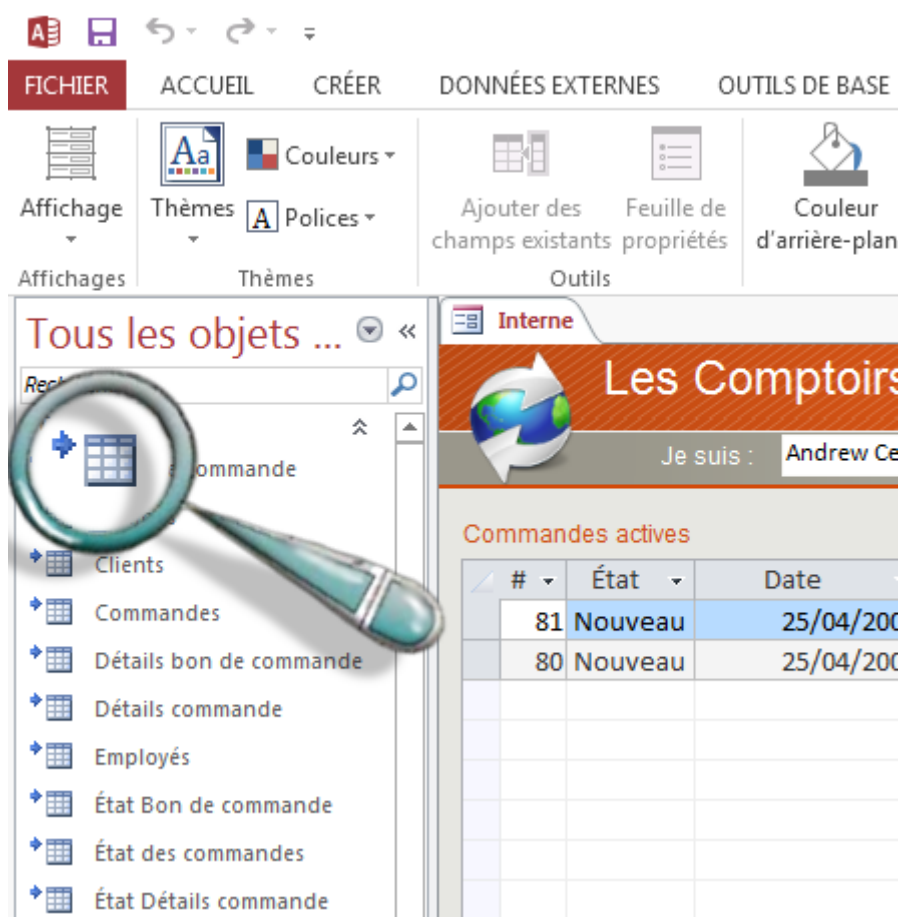
Dans la boîte de dialogue suivante, la liste des tables de la base de données fait son apparition ; vous les sélectionnez [toutes] puis cliquez sur le bouton **OK** pour valider.

**⚠ Prenez garde de ne pas sélectionner les tables systèmes si toutefois vous aviez laissé la vue sur ce type d'objet dans Microsoft Access.**





Les tables sont dès lors attachées à votre application.



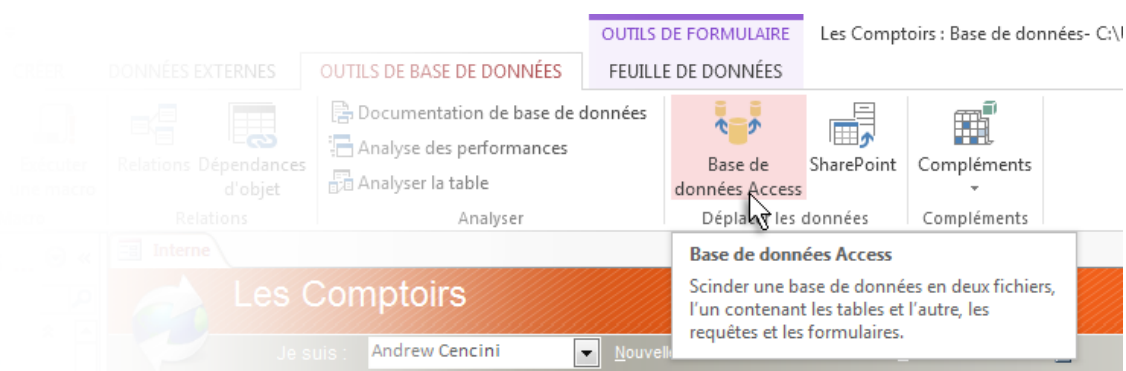
Remarquez les icônes pourvues d'une petite flèche caractéristique des tables liées Microsoft Access.

### **Méthode automatique et assistée :**

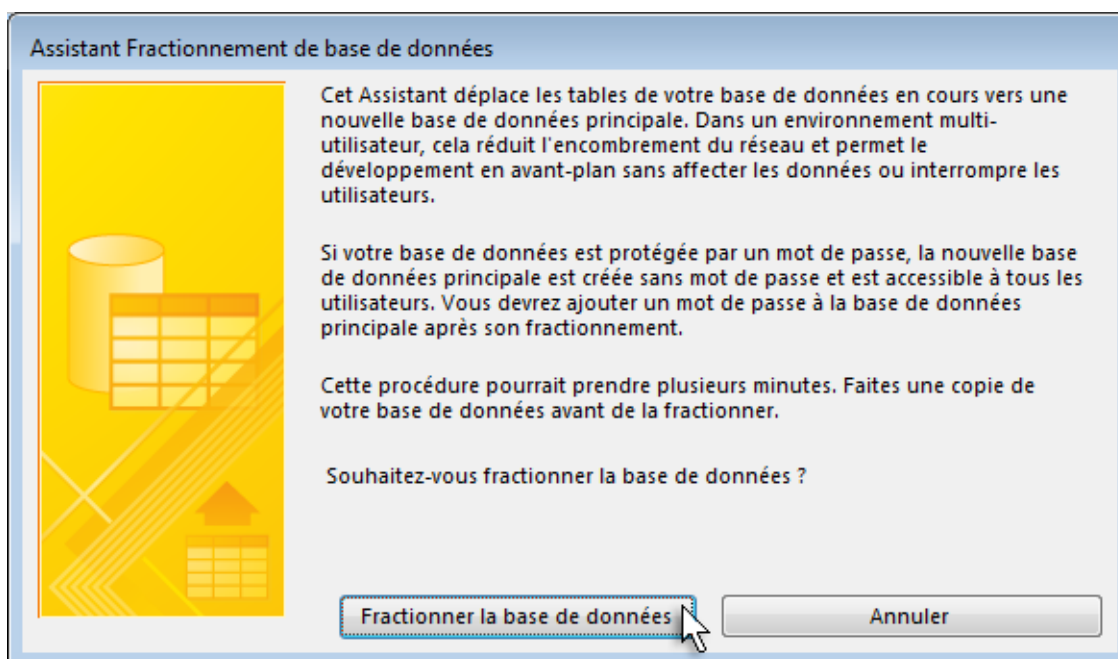
Vous avez également la possibilité de procéder à cette séparation par un **Assistant** disponible dans Access 2013.

**i** *Cet Assistant existait dans les versions précédentes de Microsoft Access. Vous le trouviez dans le Menu **Outils** ; il se nommait «**Fractionner une base de données**».*

Pour ce faire, cliquez sur le bouton **Base de données Access** depuis le ruban **Outil de base de données**,



et suivez les instructions à l'écran.



Au sein de votre projet, vous retrouvez alors toutes vos tables à l'identique avec la petite flèche bleue, symbole caractéristique d'une table Access liée.

Deux tutoriels abordent ce sujet en détail :

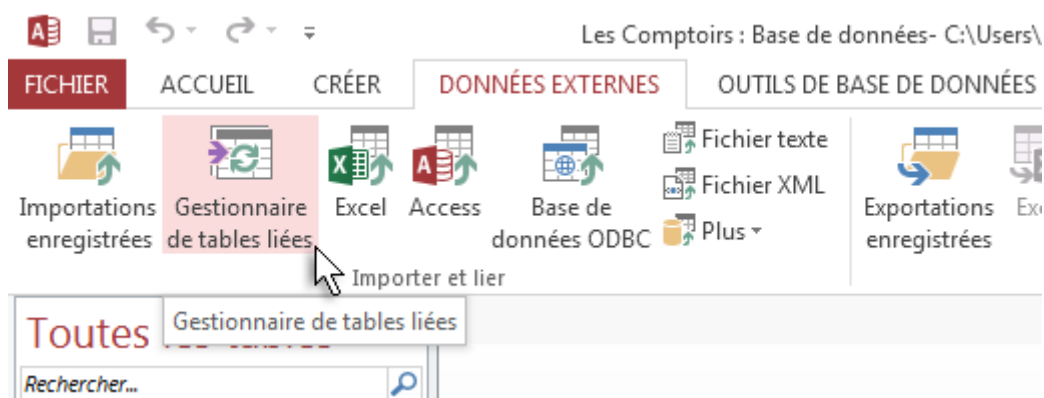
- **Liaisons dynamiques pour bases fractionnées**
- **Comment utiliser une application en mode multi-utilisateur**

**Remarque :**

**i** Dans le cas d'une liaison établie avec un autre type de base de données ou via ODBC, cette icône change d'aspect.

**Vérifier, rétablir ou modifier les liaisons des tables**

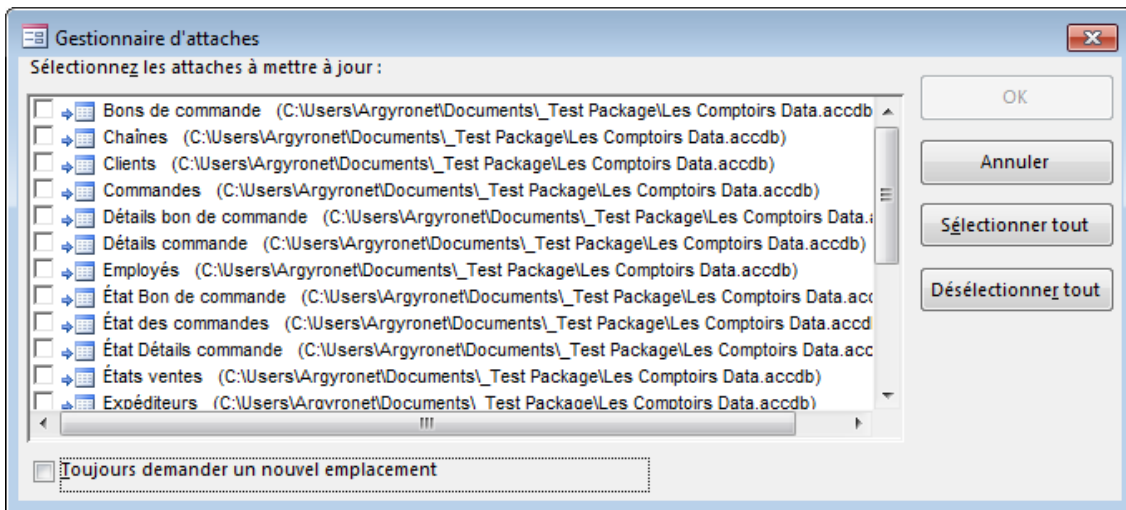
En admettant que vous ayez déjà établi une liaison avec une base de données externe comme il se doit mais que celle-ci n'est pas située au bon endroit, vous pouvez rattacher les tables. Pour ce faire, utilisez le **Gestionnaire de tables liées** et vérifiez l'emplacement.




Si vous jugez que la modification est nécessaire :

- sélectionnez toutes les tables concernées ;
- cochez alors la case «**Toujours demander un nouvel emplacement**» ;
- cliquez ensuite sur le bouton **OK** ;

- localisez la nouvelle base de données ;
- validez.



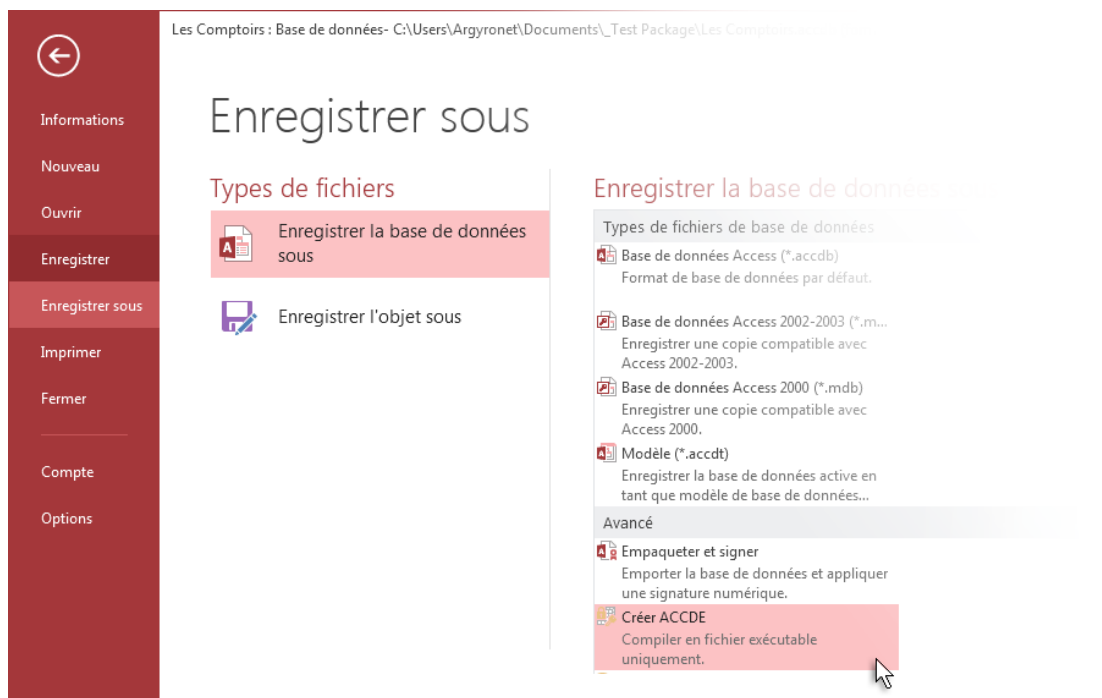
La liaison s'établira d'elle-même avec les nouvelles tables si peu que celles que vous avez sélectionnées soient présentes dans la base en question.

 *Si vous développez votre application en dehors de l'entreprise, il est probable que vous rencontriez une erreur au niveau de la liaison, car l'accès au réseau ne sera pas possible.*

### 2-3-3 - Création du fichier ACCDE

Dans cet Assistant Package, j'ai inclus une case à cocher qui permet de **générer un fichier ACCDE**. Après confirmation, et si votre base est exempt d'erreurs de compilation, le fichier est généré, il remplacera celui que vous avez sélectionné. Vous verrez ce point plus loin à la section **5-2-2. (Processus de déploiement)**

Sinon, vous pouvez aussi générer manuellement un fichier ACCDE, ne serait-ce pour vérifier que votre application tourne aussi bien dans ce format que dans l'autre. Pour ce faire, vous sélectionnez depuis l'onglet **Fichier**, l'élément **Enregistrer sous** après quoi vous repérez la catégorie *Enregistrer la base de données sous* et cliquez sur l'élément «**Créer ACCDE**».



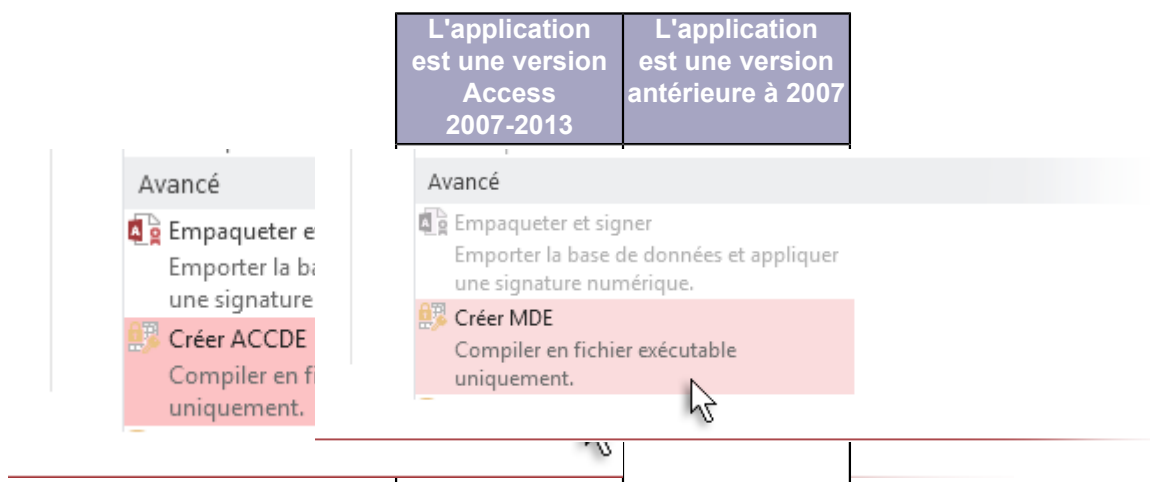
La boîte de dialogue s'ouvre et Microsoft Access vous demandera de confirmer le nom du fichier pour le générer. Par défaut, le nom proposé est celui du fichier **.ACCDB** mais muni d'une extension **.ACCDE**.

*Une fois que vous avez cliqué sur le bouton **Enregistrer** pour créer votre fichier, l'application originale, c'est-à-dire celle avec l'extension **ACCDB** se recharge aussitôt après. Donc, prenez les précautions qui s'imposent pour empêcher l'exécution de certains processus automatisés tels que par exemple la macro **AutoExec**.*

Vous pouvez ici changer le dossier dans lequel vous souhaitez créer le fichier **ACCDE**. Par défaut, ce dossier est celui où a été créée l'application.

*Vous ne pouvez générer un fichier **.ACCDE** qu'à partir d'un fichier au format supérieur ou égal à Microsoft Access 2007.*

**!** *Si vous ouvrez un fichier d'une version antérieure, vous ne serez qu'en mesure de créer un fichier MDE et de toute façon, ce format n'est pas supporté par le présent outil de déploiement de ce tutoriel.*



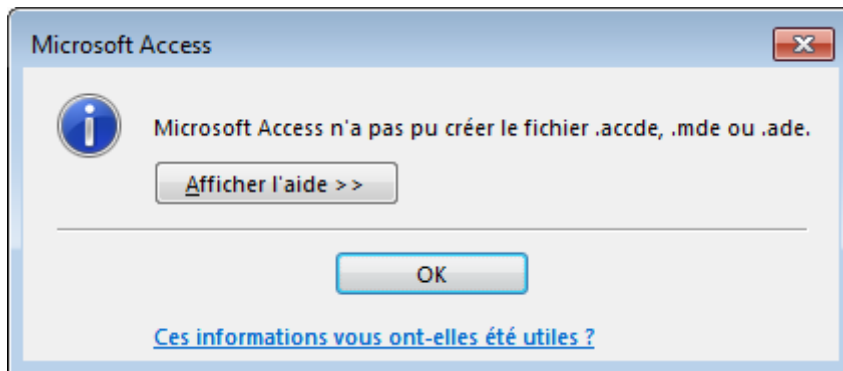
**Attention.**

**!** Il est important aussi de considérer que si votre projet est issu d'une base Access Office 2003 ou antérieure, et que vous souhaitez la convertir en vue de l'enregistrer dans le nouveau format, quelques adaptations seront sans doute à prévoir, notamment en ce qui concerne les références et les interactions avec les autres applications comme Excel et Word.

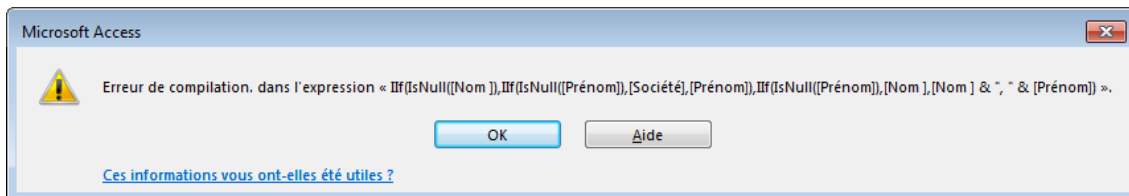
**2-3-4 - Impossibilité de générer un fichier ACCDE**

Du fait que vous ne pouvez créer un fichier **ACCDE** qu'à partir d'une version Microsoft Access 2007 au minimum de votre application, toute tentative avec un format 97-2003 renverra une erreur. Vous ne serez pas non plus en mesure de générer un fichier de ce type si la base de données possède une ou plusieurs erreurs de compilation dans le code (*référence de bibliothèque invalide, erreur de syntaxe, etc.*) Dans les deux images ci-après, vous avez un exemple caractéristique de deux messages précisant que la génération du fichier ACCDE n'est pas réalisable.

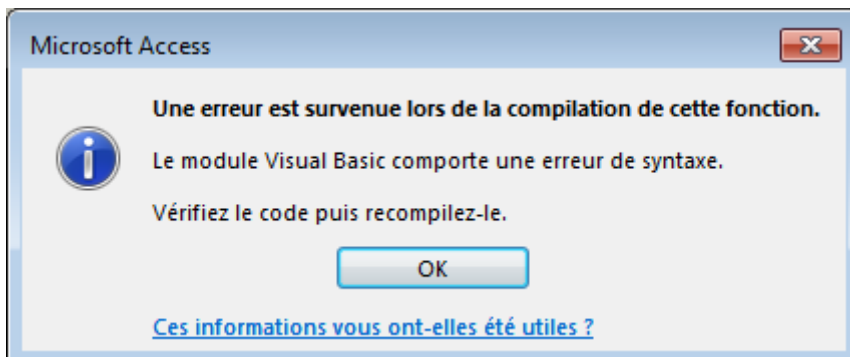
Sans en préciser la raison,



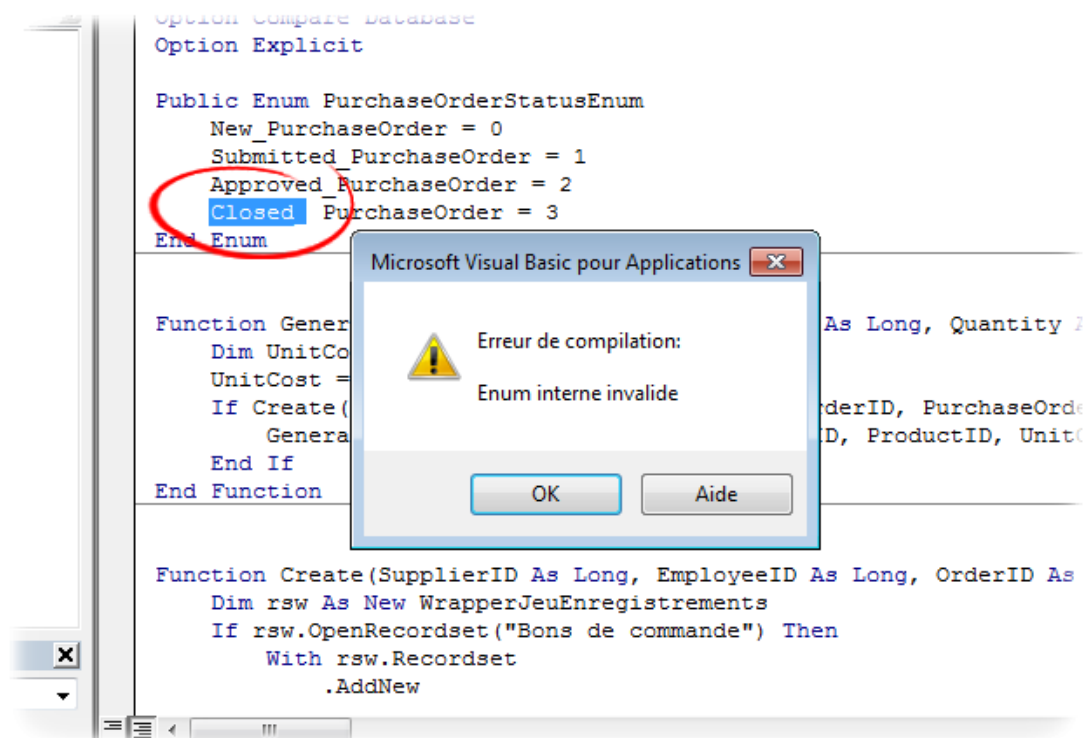
mais parfois, il vous la renvoie :




et même, il vous dit qu'il y a une boulette :




et débordant de bonne volonté, il vous dit même où (*enfin si vous compilez le code, il est gentil, mais pas dupe*) :



 Je recommande donc à tous les développeurs d'envisager de mettre en place des gestionnaires d'erreurs idoines dans **la plupart** des procédures et fonctions sauf dans certaines classes ou sous-procédures mais également de poser les instructions **Option Explicit** dans toutes les pages de codes.


## 2-4 - Sécurisation de votre Projet

*Juste un petit mot sur la sécurité où je ne serais pas plus exhaustif que mes acolytes.*  
 *Un authentique article sur la sécurité irait beaucoup plus loin.*  
*Cette section a pour seul but de vous montrer que la notion de sécurité doit être prise en compte de façon sérieuse.*

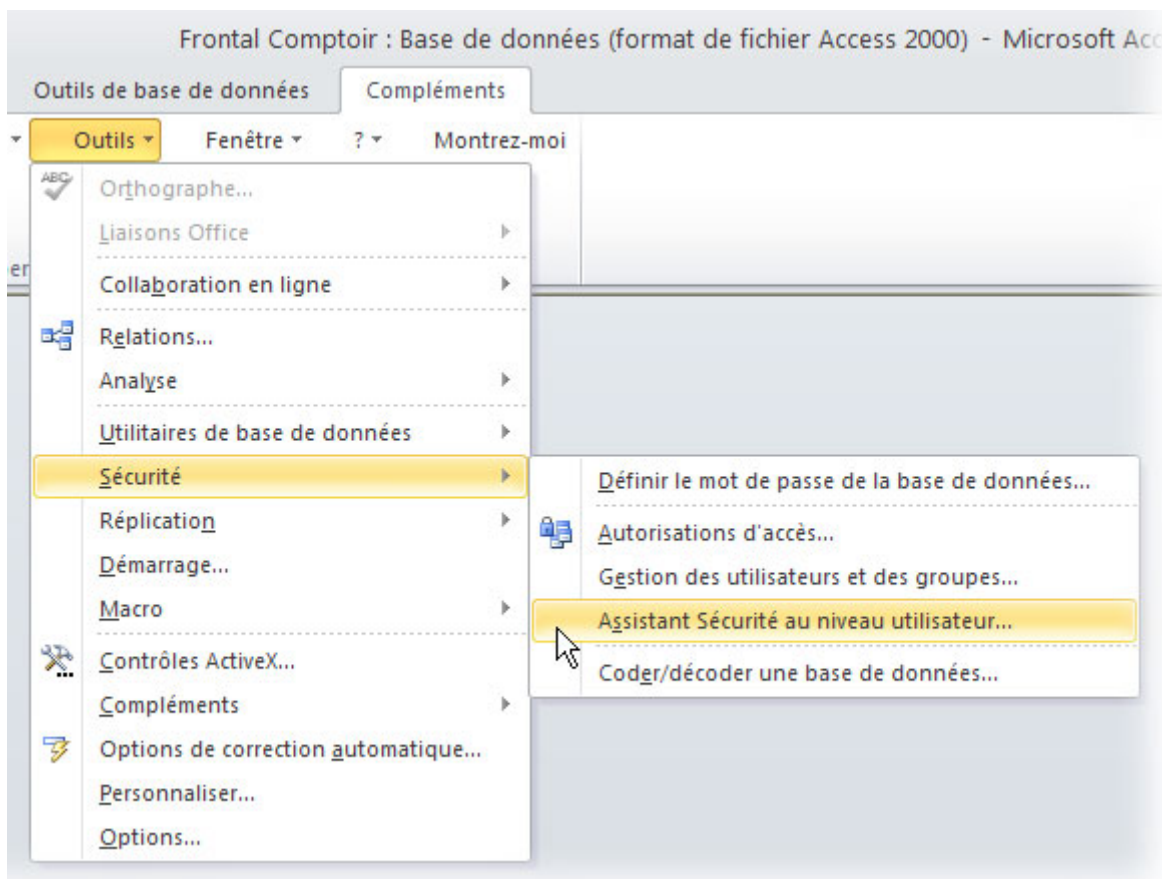
Un vaste débat initié par **Christophe Warin** y est abordé [ici](#).

### 2-4-1 - Comment sécuriser votre application ?

Si vous distribuez votre application à des utilisateurs qui disposent de la version 2013 de Microsoft Access sur leur PC, plusieurs précautions s'imposent pour protéger votre application, et ce dans le but d'empêcher les éventuelles modifications des objets et du code ou de perturber par inadvertance le fonctionnement de l'application.

 *Microsoft Access 2013 ne permet plus d'utiliser la sécurité de niveau utilisateur pour les bases de données au format 2007-2013 (.ACCDB ou .ACCDE).*





Pour obtenir davantage d'informations à ce sujet, vous pouvez inscrire la rubrique *Sécurité utilisateur* puis *Fonctionnement de la sécurité avec les bases de données de versions précédentes d'Access ouvertes dans Access 2013* dans l'aide de Microsoft Access 2013.

#### **En dehors de cela, pour sécuriser votre application, il vous faut :**

- Configurer toutes les propriétés de démarrage de la base de données susceptibles de permettre aux utilisateurs d'accéder au **Volet d'Exploration** et par la même au mode **Création**.
- Donner à la propriété **AllowBypassKey** la valeur **False** pour désactiver la touche **Maj**. Cette précaution interdit aux utilisateurs d'ignorer les propriétés de démarrage ou la macro **AutoExec**. Vous trouverez un exemple illustré pour paramétrer la propriété **AllowBypassKey** par le biais de Visual Basic **en cliquant ici**.
- Protéger le code Visual Basic par un mot de passe (environnement VBE, Menu **Outils, Propriétés** de Votre projet, onglet **Protection**).

**!** *Si vous choisissez de mettre en œuvre la protection de votre projet avec la propriété **AllowBypassKey**, prévoyez pour vous-même une possibilité de désactiver celle-ci sans quoi, vous ne pourrez plus accéder (facilement) à votre application. Par exemple, vous pouvez ajouter un bouton caché dans un formulaire qui permet de passer cette propriété à **True**.*

## 2-4-2 - Les propriétés de protection d'une application

Il existe plusieurs propriétés dont vous pouvez écrire l'affectation par programme en VBA. Parmi elles :

- - utiliser les touches spéciales d'accès (*AllowSpecialKeys*) ;
- - autoriser la vue du code sur erreur (*AllowBreakIntoCode*) ;

- - afficher la fenêtre de base de données (*ShowStartupDBWindow*) ;
- - autoriser les modifications des barres d'outils/menus (*AllowToolbarChanges*).

Pour configurer ces propriétés, cliquez sur l'onglet **Fichier** puis cliquez sur le l'élément **Options** puis choisissez **Base de données active** ou bien définissez ces options par programme en VBA.

***Rappel***

*Si votre base de données contient du code Visual Basic, distribuez-la sous forme de fichier ACCDE.*

*En effet, ce format de base de données ACCDE compile tous les modules, supprime le code source et compacte la base de données tout en protégeant l'application de toutes modifications.*

*En revanche, si vous souhaitez laisser aux utilisateurs la possibilité de créer leurs propres formulaires ou leurs états, ne distribuez pas votre application au format ACCDE et ne verrouillez pas les options de démarrage.*



***Rappel :***

*Si vous avez mis un mot de passe sur l'éditeur VBA, il n'y aura pas possibilité de créer de procédures événementielles sur les contrôles ajoutés ou modifiés dans les formulaires et/ou les états. Ceci dit, depuis Access 2010, les **macro-incorporées** sont utilisées en lieu et place du code VBA pour les commandes simples à l'aide de l'**Assistant contrôle** tout en laissant une certaine transparence pour certains utilisateurs de votre application.*

Pour plus d'information à propos de la sécurité de **l'accès à la base de données elle-même**, même si elles ne sont plus trop d'actualité, vous pouvez vous rendre sur le forum où toutes ces spécifications sont détaillées par **Maxence Hubiche** par exemple ici : **Les clauses de sécurité Access**

## 3 - Déploiement et empaquetage de votre application

Nous y voilà.

Vous pouvez envisager deux cas de figure pour le déploiement de vos applications. La section déploiement expliquée dans ce document concerne les deux cas avec ou sans **Runtime** puisque au final, le principe reste le même :

### 3-1 - Ambiguïté concernant la génération de packages

Pour ceux d'entre vous qui avaient Access 2010, vous aviez certainement vu comment générer un package d'installation où il ne fallait s'emmêler les pinceaux entre les deux rubriques qui existaient au sein de la fonction **Enregistrer et publier** (*Enregistrer sous dans 2013*) à savoir :

- **Empaqueter et signer** qui permettait de générer une version **signée** de votre application

- **Solution de package** qui permettait de créer un **package** d'installation que le projet soit signé ou non

Il est vrai que d'un point vu purement étymologique, la confusion en a laissé coi certains d'entre vous où vous m'écriviez soit dans le **forum**, soit en privé (*c'est pas bien !*) qu'aucun **MSI** n'était généré ; et pour cause, la rubrique était mal interprétée et pour ma part, mal traduite.

Mais aujourd'hui, dans Access 2013, **plus question de confusion** puisque l'outil de **Solution de package** n'existe plus d'une part et que **l'Assitant Package** que j'ai développé propose sa fonctionnalité d'empaquetage à partir du ruban dans le groupe **Compléments** de l'onglet **Outils de base de données**.

*Comme expliqué au début de ce tutoriel, je n'ai pas trouvé de solution pour faire avaler à ce satané Access que mon complément possède un **Menu Backstage**.*

**Qu'est qu'un fichier signé ?**

Un fichier empaqueté et **signé** possède une extension qui lui est propre : **ACCDC** (à ne pas confondre avec le groupe ;o)

Ce procédé requiert une **signature numérique** souscrite auprès d'un organisme agréé.

Pour la démonstration, je vous propose de voir la rubrique abordée dans le tutoriel traitant de **l'empaquetage de solution Access 2010**.

### 3-2 - Préparation en vue d'exploiter l'application avec le Runtime Microsoft Access 2013

Si vous souhaitez déployer votre application en vue d'une utilisation avec le Runtime Microsoft Access, quelques précautions sont à prendre :

1. Que vous ayez, avant d'avoir compilé et enregistré votre application en **.ACCDE**, testé celle-ci en *mode Runtime*.
2. Que vous fassiez en sorte que le **Runtime** soit installé sur le poste utilisateur (*cette option étant automatisable*).

#### 3-2-1 - Préparation en vue d'exploiter l'application avec la licence Microsoft Access

Si vous souhaitez déployer votre application en vue d'une utilisation avec une licence Microsoft Access, vous devez vérifier que la version de Microsoft Access du poste utilisateur est identique à la vôtre en l'occurrence la version 2013.

*Cet outil d'empaquetage n'accepte que le Runtime 2013 (version 15.0.4517.1004 ou ultérieure) d'une part et les bases de données au format 2007-2013.*



*Il est possible qu'une base de données au format **2013** lève des erreurs si elle est exploitée avec le **Runtime 2010**.*

#### 3-2-1-1 - Comment obtenir des outils nécessaires au déploiement de solutions Microsoft Access ?

Pour pouvoir bénéficier des outils nécessaires au déploiement d'applications Microsoft Access 2013, vous devez installer l'application **Package Wizard 2013** que j'ai développée en la téléchargeant directement depuis cet article.

Contrairement aux versions précédentes qui, pour les versions 2000 à 2003 nécessitaient d'être achetées, (*voir articles 2000 et 2003*), et aux versions 2007 et 2010 où il était fourni respectivement par téléchargement pour l'un et par installation complémentaire pour l'autre (*voir articles 2007 et 2010*), l'outil d'empaquetage et de déploiement pour Access 2013 a été **déprécié** comme de nombreuses autres fonctions d'ailleurs.

Vous pouvez consulter [la page à ce sujet ici](#).

Pour ce qui est de l'outil mis à votre disposition, c'est la même chose que pour les versions précédentes, il faut juste le récupérer d'abord et l'installer ensuite.

Les fichiers d'installation du Runtime Access 2013 n'ont pas besoin d'être téléchargés puisque je l'ai intégré dans le SETUP d'installation.

Notez au passage que tout comme pour les versions 2007 et 2010, il est également **GRATUIT**.

Ne soyez pas surpris par la taille du fichier (*il a beaucoup de code ! Euh., non je plaisante.*).

En effet, le programme d'installation complet pèse près d'un giga-octet du fait qu'il soit *français/anglais* et qu'il embarque les runtimes respectifs dans les deux langues d'une part et les deux technologie 32 et 64bits d'autre part, considérant que chacun d'eux pèse approximativement entre 215 et 290 Mo.

Vous pouvez bien entendu ne télécharger que la version sans Runtime qui ne pèse que 1 Mo.

Vous devrez alors télécharger si nécessaire le Runtime dans la bonne langue et la technologie souhaitée, depuis le site de Microsoft en cliquant [ici](#)

(Changez-le **fr-fr** en **en-us** pour la version anglaise).

Il est important de considérer que l'installation du Runtime 2013 requiert un système d'exploitation compatible.

#### **Systèmes d'exploitation pris en charge :**

- Windows 7
- Windows 8
- Windows Server 2008 R2
- Windows Server 2012

### 3-2-1-2 - Installer l'outil Assistant Package sur votre poste

**L'Assistant Package** est conçu pour les développeurs professionnels qui créent et souhaitent **déployer** facilement des applications **Microsoft Access 2013**.

Il peut se coupler au Runtime Access 2013 permettant de distribuer des applications Microsoft Access sur des postes dépourvus de la version complète de Microsoft Access 2013.

Avant d'installer **L'Assistant Package**, prenez la précaution de quitter tous les programmes encore actifs et notamment Microsoft Access, même si Windows Installer vous demande de relancer Access s'il était encore ouvert alors que vous aviez lancé l'installation.

Pour installer **L'Assistant Package**, veuillez vous rendre à la [section 5](#).

## 4 - Présentation de l'Assistant Empaquetage et déploiement

Le complément Assistant Package pour **Access 2013** est une application Access **32bits** et **64bits** (*la version 64bits a été créée en octobre 2014\**), qui s'utilise depuis Access 2013 dans le but de créer un package d'installation de vos applications que vous avez développées.

\* Il est donc possible d'utiliser ce complément avec Office 64bits.

Cet outil a été développé avec un aspect global qui reste similaire aux précédents avec un look orienté dans l'esprit coloré d'Access 2013 et quelques fonctionnalités supplémentaires qui n'existaient pas dans les versions antérieures. Pour ceux d'entre vous qui ont déjà utilisé les outils précédents, vous le constaterez rapidement.

L'outil est bilingue **Français/Anglais** (*Je pense ne pas avoir fait trop de fautes*) avec bien entendu, le français comme langage par défaut.

L'outil servant à générer vos packages se nomme **Assistant Package/Package Wizard**.

Son fonctionnement permet de générer des programmes d'installation de vos applications Access 2013 par la génération d'un fichier **MSI** et d'un programme d'installation (*nommé Setup.exe*) en vue de les distribuer.


J'ai relooké les formulaires par rapport à celui de la version 2010.

La mise en œuvre de packages en sera je l'espère, facilitée, notamment la disparition des ascenseurs qui vous permettra dès lors de voir le contenu complet de chaque fenêtre.

Si vous savez que les postes clients ne disposent pas de la version complète de Microsoft Access 2013, cet assistant vous propose d'inclure le **Runtime Microsoft Access 2013**.

Deux choix sont alors possibles :

- la case n'est pas cochée : nécessite que la version complète de Microsoft Access 2013 soit présente sur le poste ;
- la case est cochée : il sera installé selon la langue choisie ;

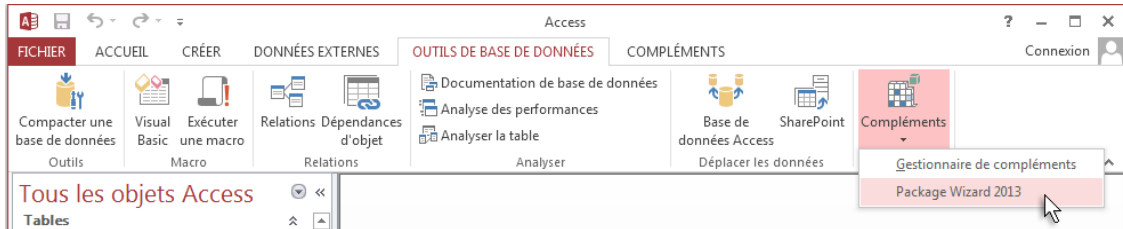
 Vous remarquerez qu'il est possible de définir un autre emplacement de stockage du programme d'installation du Runtime, celui-ci étant par défaut dans le sous-dossier **pkwRuntime** de l'Assistant Package lui-même. Si un chemin est inscrit, c'est qu'il a été détecté comme présent.

L'assistant regroupe tous les composants dont votre application a besoin pour fonctionner et automatise la plupart des tâches de création et de déploiement des fichiers tout en vous laissant la possibilité d'apporter les adaptations nécessaires pour les machines des utilisateurs finaux.

## 5 - Déploiement de votre application (avec ou sans le Runtime Access 2013)

Pour déployer vos applications avec un programme d'installation, installez d'abord l'**Assistant Package** si ce n'est pas fait.

Il est aisé de constater que module est disponible en cliquant simplement sur l'onglet **Outils de base de données**, puis le groupe **Compléments**.



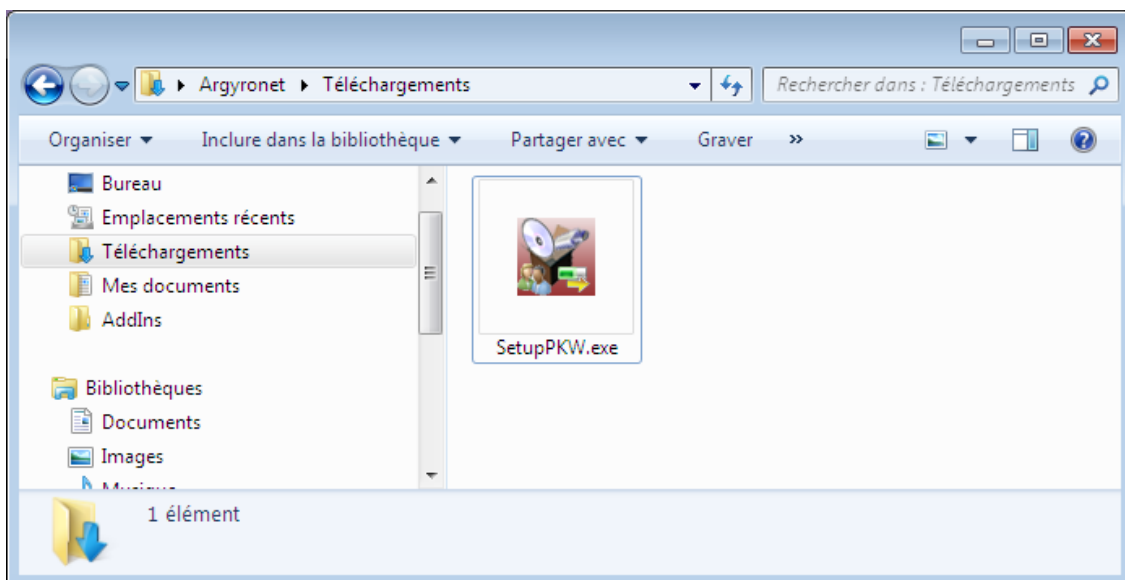
### 5-1 - Comment installer l'Assistant Package 2013

Une fois que vous l'avez téléchargé vous pouvez installer l'**Assistant Package**. Pour ce faire, vous exécutez alors le fichier **SetupPKW.exe** et suivez simplement les instructions.

Localisez l'icône du programme d'installation soit sur le **Bureau**,



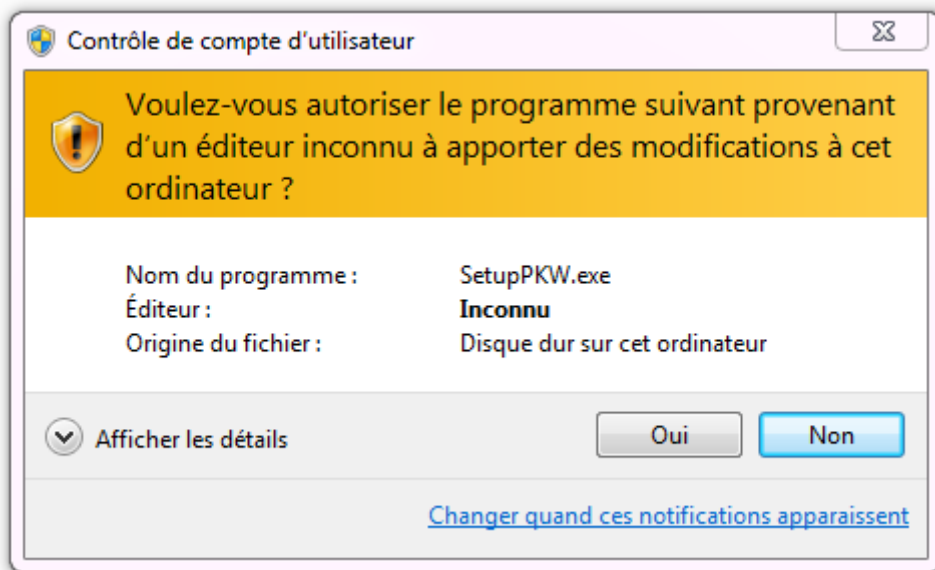
soit depuis le **dossier** où vous l'avez téléchargé.



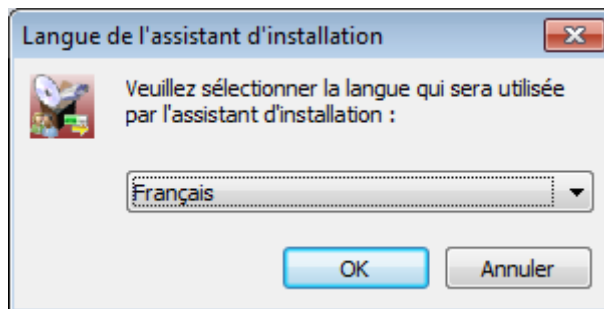
Si le message de sécurité survient pour vous prévenir de la possible dangerosité du programme que vous installez, je vous le dis tout de go, vous ne risquez rien.

Et pour certains d'entre vous, je ne suis plus un inconnu ;o)

En fait, cette fenêtre d'avertissement survient seulement en fonction du degré que vous avez défini pour le **Contrôle du compte utilisateur** dans Windows.



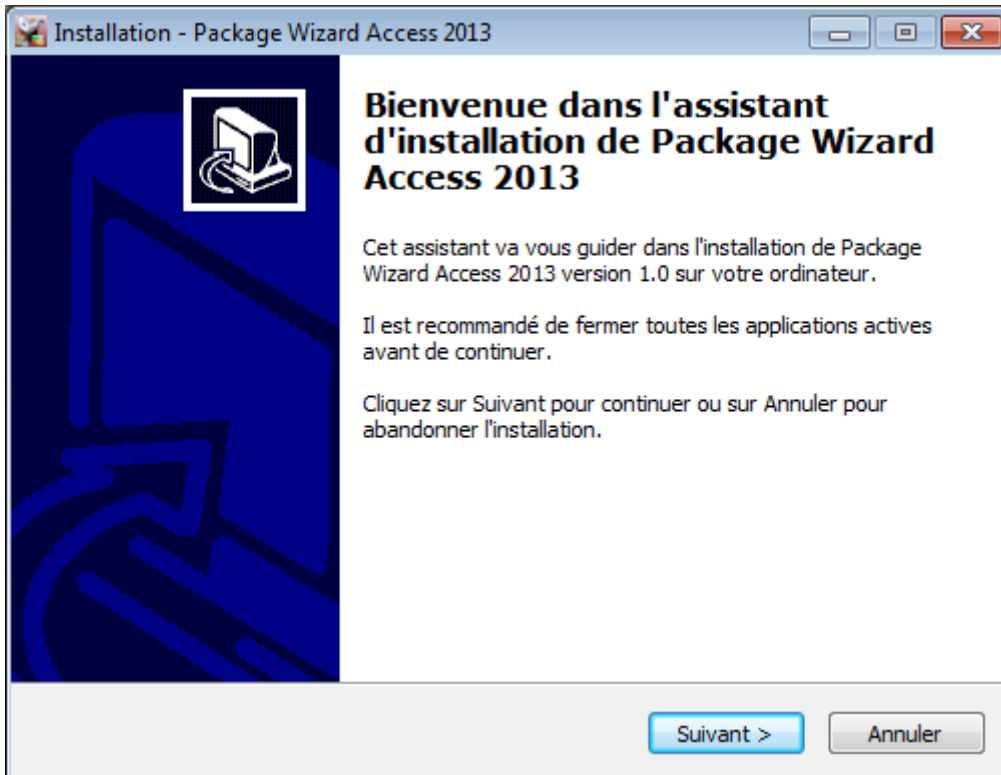
Deux langues sont alors disponibles avec le français par défaut ; cliquez sur **OK**



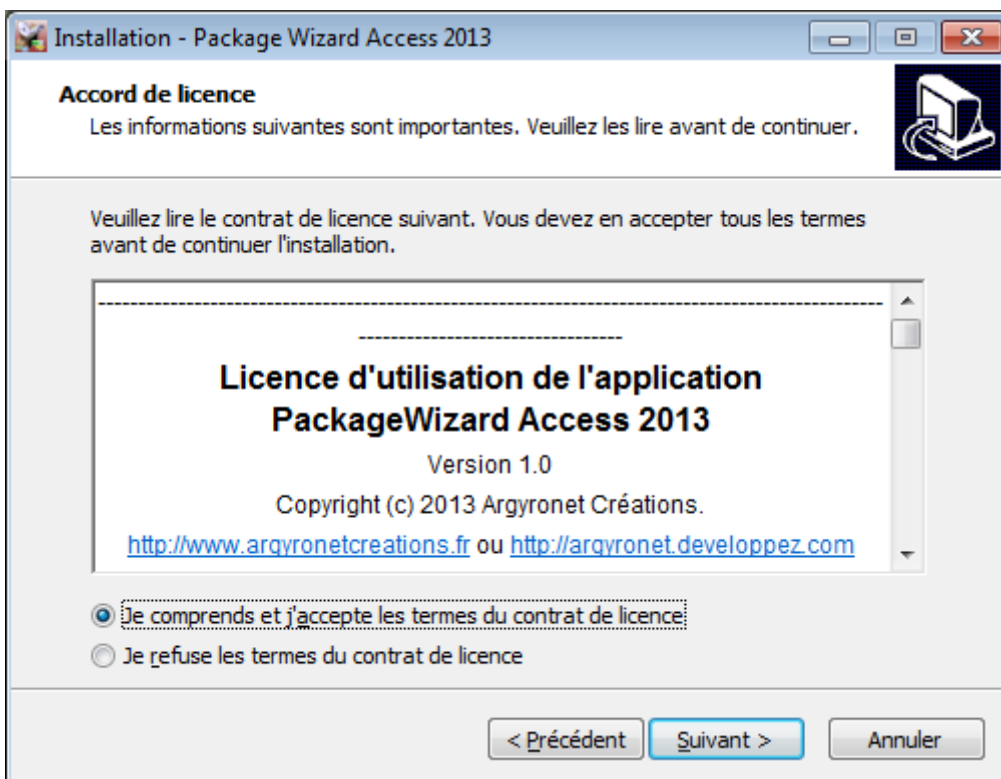
L'invite du programme d'installation s'affiche en vous présentant ce que vous allez installer. Humm, si ça se trouve, c'est même pas vrai !

Allez, cliquez sur **Suivant**.

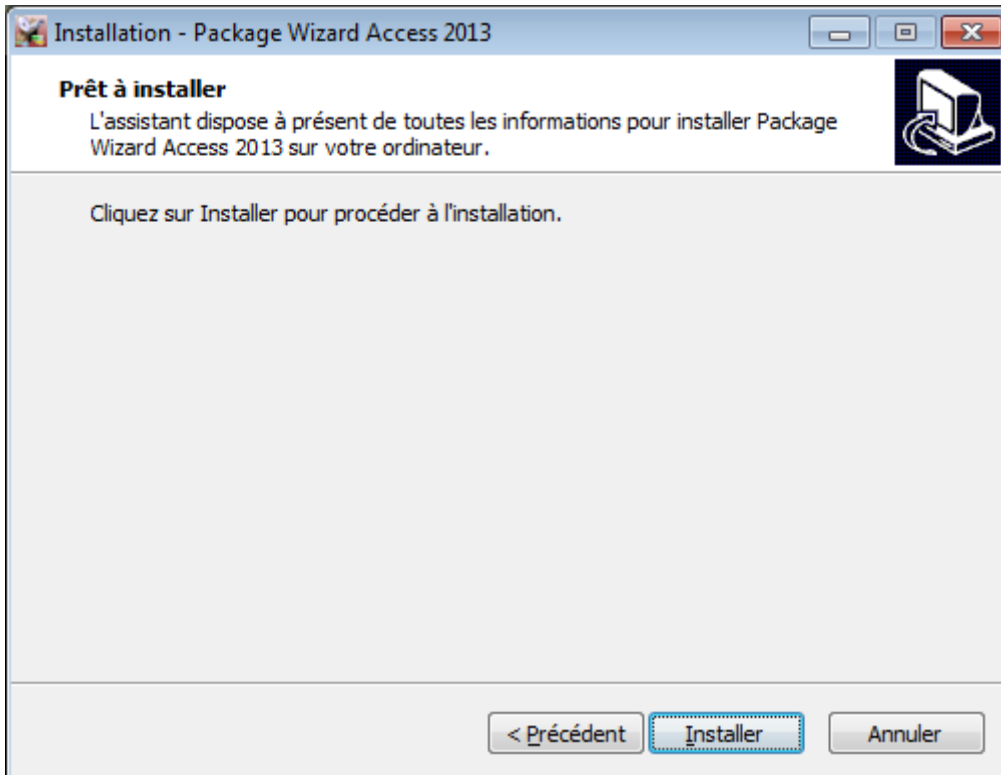




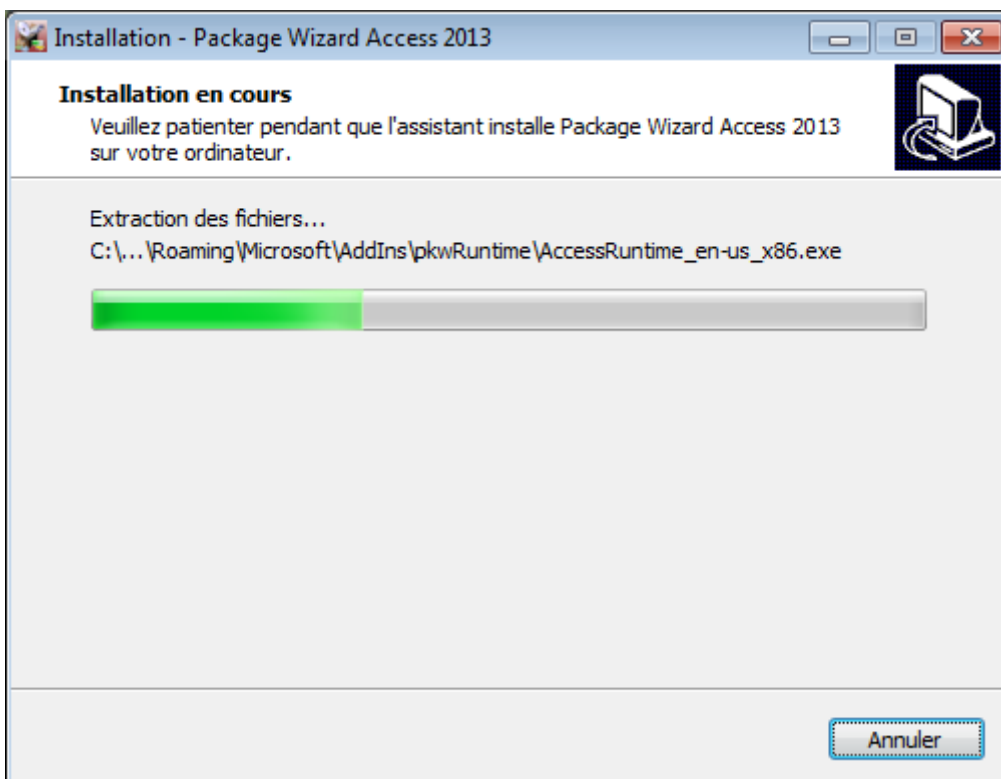
Cochez la case d'acceptation de la licence. Pour les plus curieux d'entre vous, vous pouvez la lire elle ne fait qu'une toute petite page.  
Cliquez sur **Suivant**.



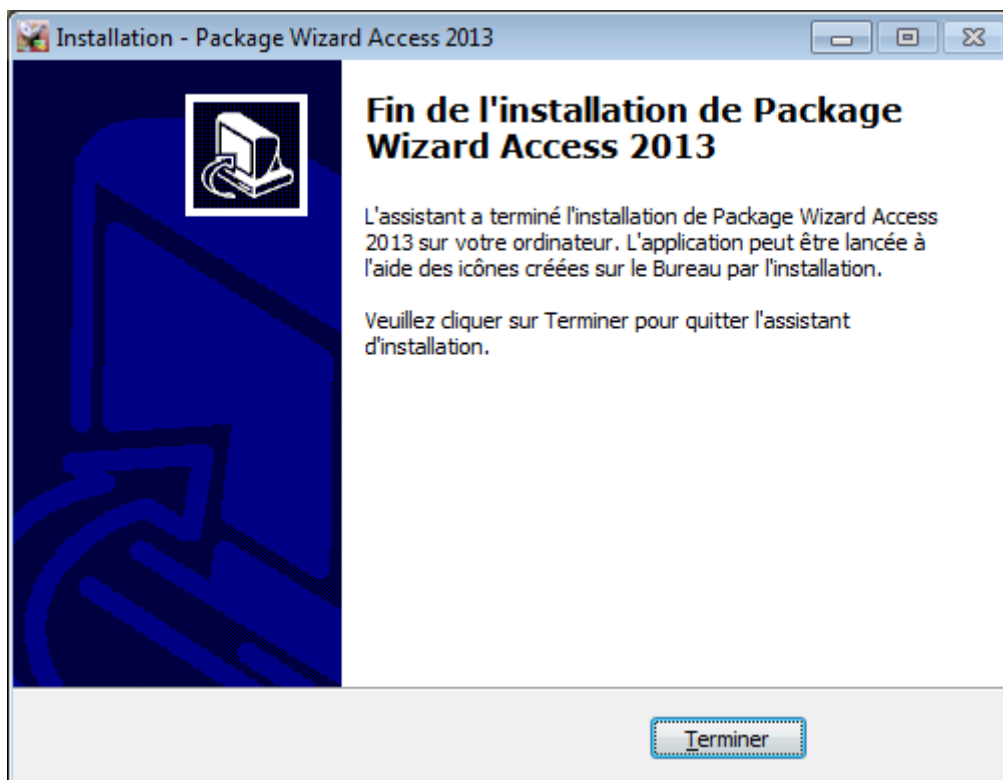
Bien là, pas grand-chose de plus si ce n'est que ça ne va pas tarder à se mettre en place.  
Cliquez sur **Suivant**.



Et ça progresse, et ça progresse. Bon, c'est un petit peu long, car il y a les deux Runtimes (FR et US), soit près de 370 Mo. Cliquez sur **Suivant**.



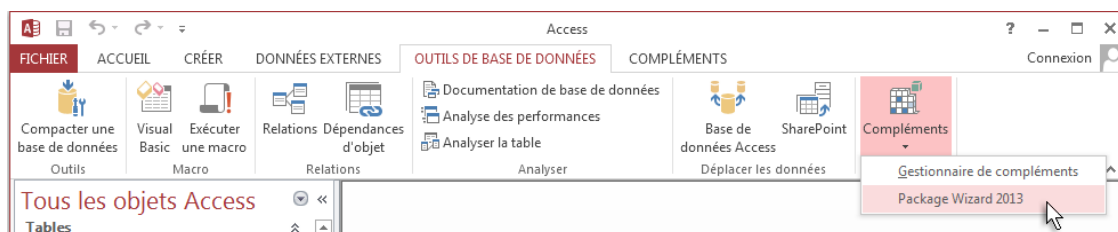
Et voilà, nous y sommes, c'est installé. Cliquez sur **Terminer**.



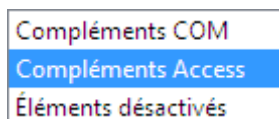
## 5-2 - Lancer l'Assistant Package

Une fois l'outil Assistant Package installé, ouvrez Microsoft Access 2013.

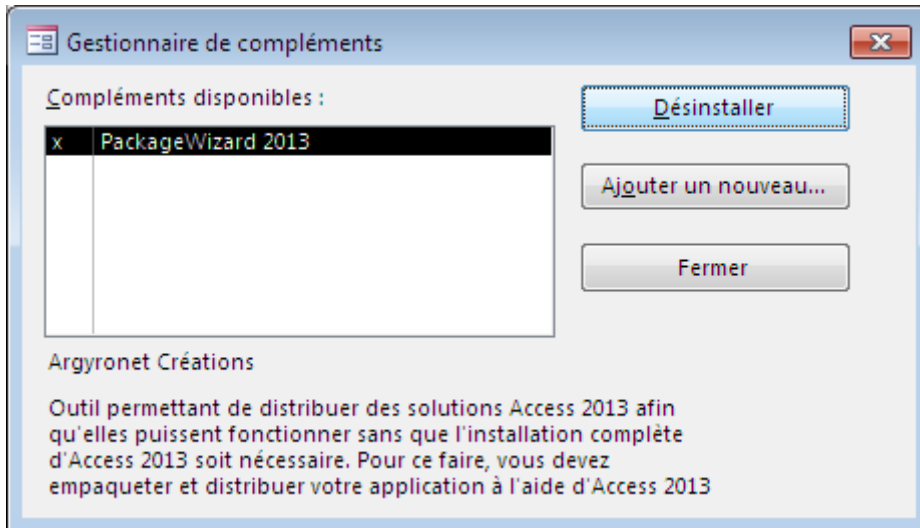
**i** Si tout s'est bien passé, l'installation du complément au sein de Microsoft Access est déjà effectuée.



**i** Dans le cas contraire, pour activer le complément depuis Microsoft Access, veuillez installer le complément manuellement via la rubrique **Options/Compléments/Compléments Access** de l'onglet **Fichier**.



Là vous choisissez **Installer** en visant le complément en question.



Puis vous validez., humm, en cliquant sur **Fermer**.

xxxxxxxxxxxxxxxxxxxxxxxxxxxx

### AVANT TOUTE CHOSE.

xxxxxxxxxxxxxxxxxxxxxxxxxxxx


Dans un fonctionnement standard, cet assistant package ne peut pas fonctionner en l'état **sans qu'il y ait une base de données ouverte**.

Vous devez donc **ouvrir** une base de données pour y accéder.

- Soit **celle** que vous souhaitez empaqueter.
- Soit une base de données **quelconque**.
- Soit une nouvelle base de données **vide**.

 *Toutefois, comme me l'a suggéré Thierry (alias **arkham46**) ; il est possible de ne pas se plier à cette obligation si vous dupliquez le groupe **Complément** dans la **barre d'outils Accès Rapide**.*

*Notez au passage que l'empaquetage s'effectue sur une **copie** de votre application de base de données et non sur l'originale.*

 *L'application à empaqueter est copiée temporairement dans le dossier d'installation que vous avez sélectionné.*

Vous pourrez alors utiliser l'Assistant Package à partir de cet instant, car l'onglet **Complément** n'est accessible que par ce biais.

Cliquez l'onglet **Outils de base de données**, puis le groupe **Compléments** pour choisir la commande **Package Wizard 2013**.

**L'Assistant Package** démarre alors.

## 5-2-1 - Démarrage de l'Assistant Package

Dès que la fenêtre fait son apparition, vous avez le choix de la langue, le **français** étant celle par défaut d'une part et le choix entre déployer un **nouveau package** ou bien d'en générer un à partir d'un **package existant**.

- Dans une étape de création d'un **nouveau package**, vous devez remplir ou saisir tous les champs obligatoires, ceux-ci étant repérables par un astérisque rouge \* (*s'il est activé*).
- Dans une étape de génération à partir d'un package existant, vous pouvez redéployer un package fondé sur un **modèle sauvegardé** qui vous évite toute ressaisie déjà effectuée ou bien de modifier certains éléments qui le constituent.

Choix de l'opération

## Assistant d'emballage pour Microsoft® Access 2013

Créez vos packages d'installation pour vos applications de base de données Microsoft® Access

Étape 1 sur 6

### Démarrage de l'Assistant Emballage

Français

English/United States

## Version 32 ou 64 bits

Bienvenue dans l'assistant d'emballage et de déploiement Microsoft® Access 2013. Cet assistant va vous aider à créer un package d'installation pour déployer et distribuer votre application.

Cliquez sur le bouton **Suivant** pour créer un nouveau package ou bien choisissez un package déjà enregistré auparavant... Vous pourrez également inclure le Runtime Microsoft® Access 2013.


**Que voulez-vous faire ?**

Créez un nouveau package

Utiliser et modifier un package existant

Liste des packages existant

À propos de... Quitter Suivant

 Lors de la génération à partir d'un package existant, vous disposez à l'étape 6 de l'Assistant de la possibilité de procéder à une sorte d'**enregistrer-sous** si toutefois vous changez le **nom du package**.  
Nous verrons cela un peu plus bas.

## 5-2-2 - Processus de déploiement

L'Assistant crée l'emballage et le programme d'installation (*setup.exe*) correspondant, en référençant tous les fichiers requis.

À l'issue de cette étape, un fichier **MSI** sera créé et accompagné des fichiers d'installation associés.

Le déploiement d'une application consiste à transférer une application emballée vers le support de distribution choisi ou vers un site Web depuis lequel elle pourra être téléchargée. Pour distribuer votre application, vous avez plusieurs possibilités :

vous pouvez copier manuellement les fichiers :

- sur un **support amovible**, un **lecteur local** ;
- ou sur **partage réseau** ;
- ou bien la publier manuellement sur un emplacement **Web** approprié.

**L'Assistant Package** propose des raccourcis et exécute automatiquement certaines des tâches qui vous incombent dans la procédure manuelle de déploiement.

### 1. Déploiement dans un répertoire

Si vous optez pour un déploiement dans un répertoire, le système vous demande de choisir un répertoire local ou réseau dans lequel copier les fichiers pour qu'ensuite les utilisateurs puissent accéder au programme d'installation.

### 2. Déploiement sur le Web

Vous pouvez, une fois votre package généré, envoyer votre empaquetage sur le Web via **FTP** par exemple.

Les fichiers et répertoires situés dans le répertoire local devront être enregistrés dans le répertoire Web avec la même arborescence que ce jeu de répertoires de base (*en respectant la casse des noms de fichier - UNIX le requiert*).

Dans le cas d'une installation directe via une page Web dédiée, il est préférable de mettre à disposition que **le fichier MSI** qui a lui seul encapsule l'ensemble des fichiers nécessaires à l'installation mais dépourvus du Runtime.

Dans ce cas, vous devrez prévoir de proposer une page Web suffisamment explicite pour avertir les utilisateurs qui ne possèdent pas la **version complète de Microsoft Access** qu'ils peuvent installer le **Runtime** et qu'il vous appartiendra de mettre un lien pointant sur l'exécutable de ce dernier.

Il n'est en effet pas possible d'exécuter le **SETUP.EXE** depuis le Web : cela génère une erreur traduite par l'impossibilité de trouver les fichiers requis - qui lorsqu'ils se téléchargent dans le dossier temporaire, se voient greffés du nom suivi d'un [1] soit par exemple, *setup[1].ini* au lieu de *setup.ini*.

Si vous avez un grand lot de fichiers embarqués en plus du MSI, il est recommandé de générer un fichier compressé (**ZIP ou RAR**) **auto-extractible** qui, par sa ligne de commande, pourra lancer **Setup.exe**.

## 5-2-3 - Étape 1/6 - Choix de la langue et du mode d'empaquetage

Lorsque vous lancez l'Assistant Package, **l'étape 1** consiste à sélectionner la langue qui par défaut est effectivement le français ; un peu normal me direz-vous.

Ici, vous sélectionnez l'option :

- **Création** d'un nouveau Package ;
- Utiliser et **modifier un Package existant** ;



La liste des packages existants n'est alimentée que si vous avez sauvegardé d'autres précédents packages.

À partir du moment où vous avez sélectionné un package existant, trois boutons qui jouxtent la liste se déverrouillent.

- Le 1<sup>er</sup> bouton vous permet de supprimer (*de façon irréversible*) un package dont vous n'avez plus l'utilité.
- Le 2<sup>e</sup> bouton vous permet de renommer un package existant (*un nouveau nom vous sera demandé*).
- Le 3<sup>e</sup> bouton vous permet de dupliquer un package existant (*un nouveau nom vous sera demandé*).

## 5-2-4 - Étape 1/6 - Création d'un nouveau package

Dans un premier temps pour exploiter ce tutoriel vous allez créer un nouveau package.

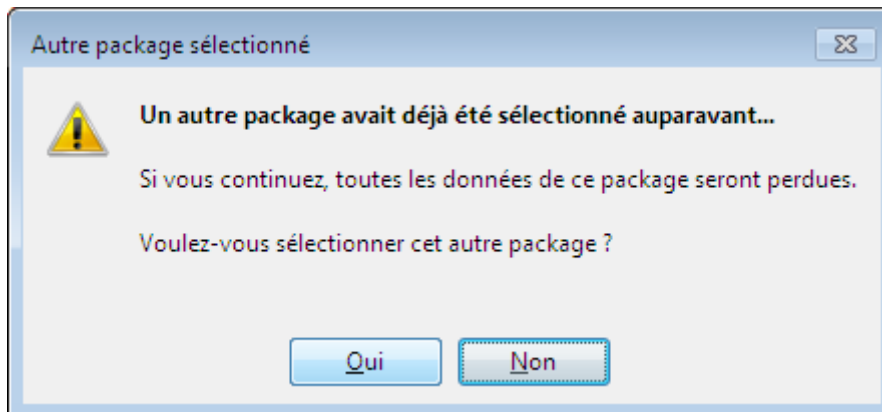
Pour ce faire, vous pouvez laisser ouverte la base de données que vous voulez empaqueter s'il s'agit de celle-ci mais c'est facultatif.

Cochez alors l'option pour définir que vous souhaitez créer un **Nouveau Package** puis, cliquez sur le bouton **Suivant** pour passer à **l'étape n°2 sur 6**.

Si vous souhaitez abandonner, le bouton **Quitter** vous permet de refermer l'Assistant après confirmation.



Attention, si un déroulement est en cours sur un nouveau package et que vous en sélectionnez un existant, vous verrez apparaître une alerte de confirmation.



### 5-2-5 - Étape 2/6 - Définition de l'application à empaqueter

La création d'un nouveau package implique le remplissage de toutes les zones, toutes n'étant remplies pas défaut. Le programme considère que c'est la base de données active qu'il faut empaqueter et donc, le chemin de celle-ci sera alimentée automatiquement.

#### **Choix du fichier à empaqueter.**

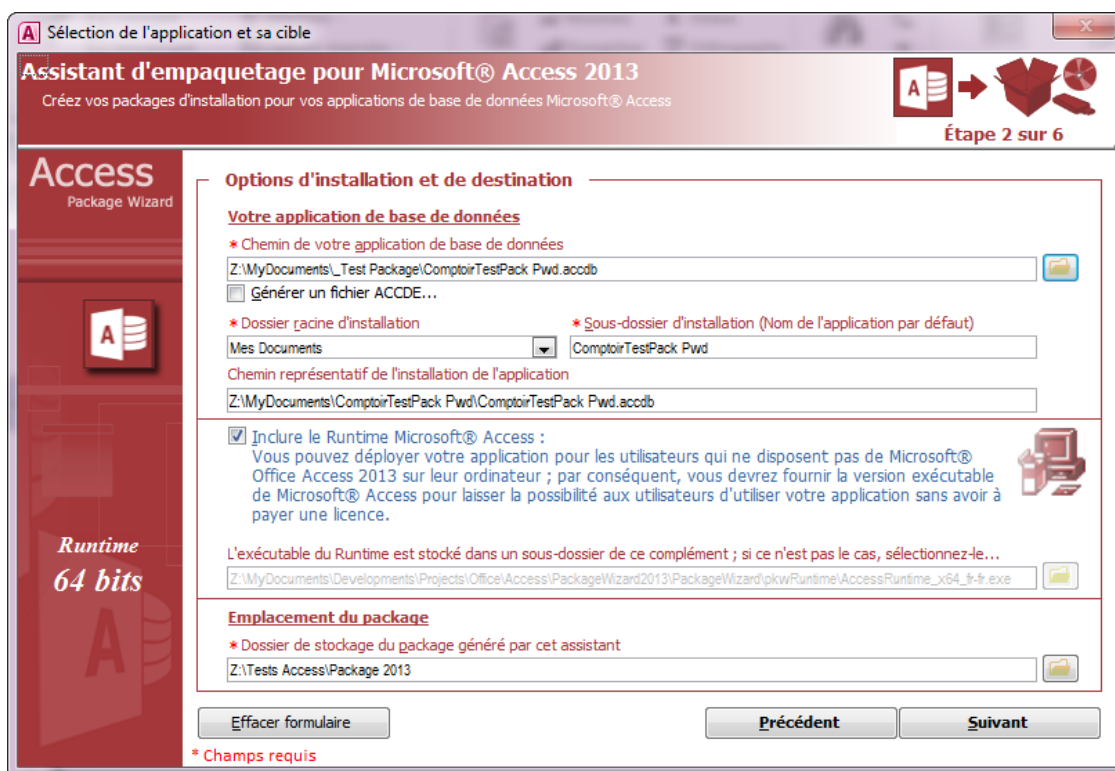
Si la base active n'est pas celle à empaqueter, alors vous définissez l'application à empaqueter à partir de la même zone de texte que vous alimentez par le biais du bouton **Parcourir** qui jouxte la partie droite de celle-ci.

Vous pouvez remarquer que certaines rubriques comportent **une petite étoile rouge \*** qui signifie que la rubrique est **obligatoire**.

Pour plus de commodité, nous allons considérer que la fenêtre est composée de trois zones :

- la première zone détermine le **nom du fichier** et le **répertoire** d'installation avec conversion en **ACCDE** ;
- la seconde zone détermine si vous voulez **inclure** ou non le **Runtime** ;
- la troisième zone détermine l'**emplacement** du package d'installation.





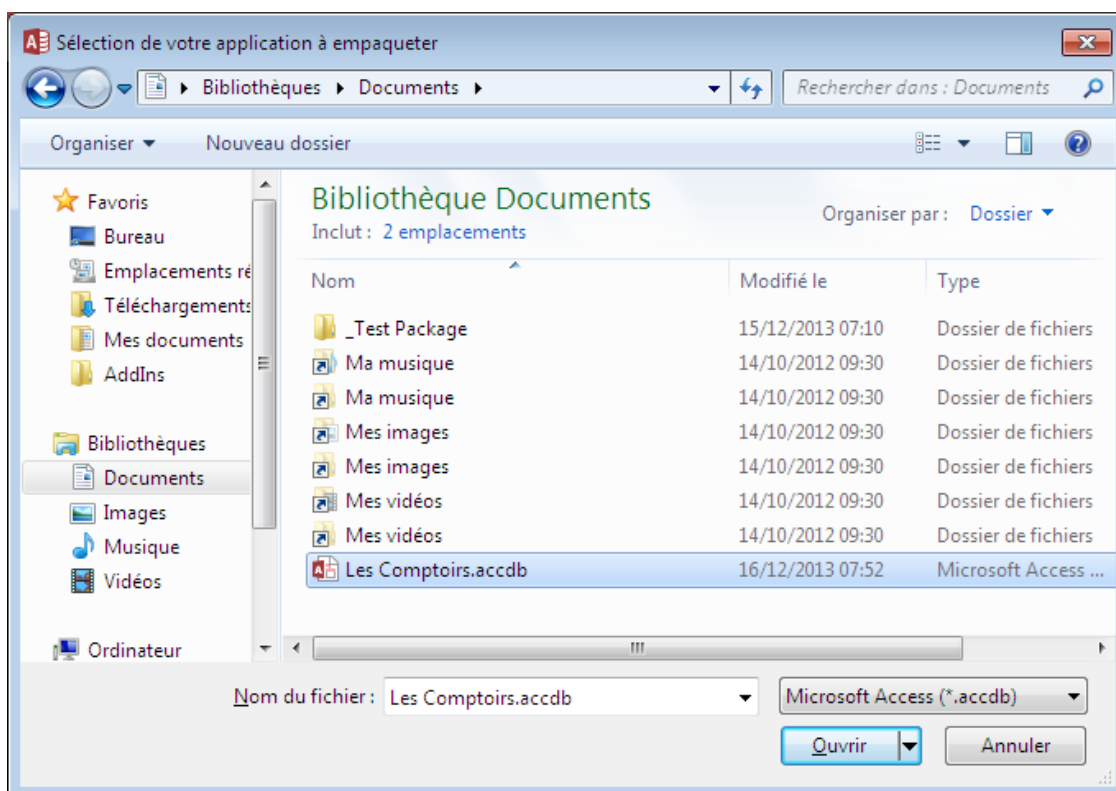
### 5-2-5-1 - Sélection de l'application Access à emballer.

Si vous souhaitez choisir une autre application de base de données que celle qui est stipulée, vous cliquez sur le bouton **Parcourir** afin d'aller chercher votre fichier (\*.accdb) dans le dossier correspondant.

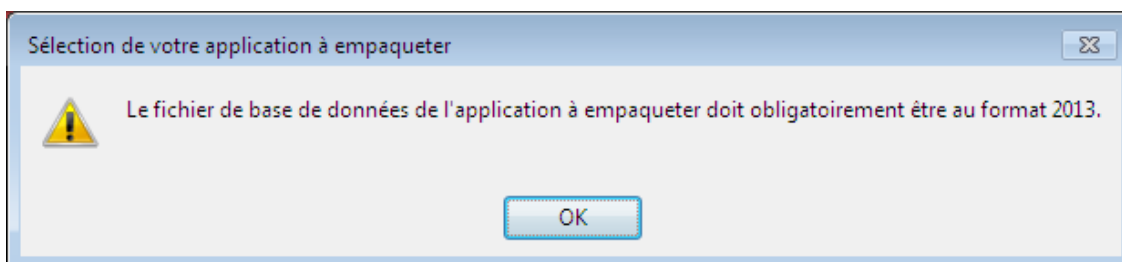
#### **Rappel :**

**!** La base de données active est choisie par défaut, et ce uniquement en cas de création d'un **Nouveau Package**.

Lorsque vous avez cliqué sur le bouton **Parcourir** pour sélectionner le fichier que vous voulez emballer, vous pouvez choisir tout type de fichier Microsoft Access ayant les extensions spécifiques qui caractérisent le format 2013.




Vous ne pouvez pas sélectionner de fichier d'un format antérieur, le processus vérifiant l'intégrité du fichier avant de le valider.



Normalement, vous êtes censé choisir un fichier de type **ACCDE** (ou **ACCDC** s'il est signé) que vous auriez préalablement créé à partir de l'application elle-même et après avoir contrôlé son bon fonctionnement comme il se doit.


Ceci dit, si vous avez à disposition un fichier ACCDE, cela peut signifier que vous avez une application qui compile correctement, euh, normalement.

Toutefois, dans cette version de l'assistant, vous pouvez demander à ce qu'un **ACCDE** soit créé à votre place. Ce dernier, une fois généré, se substituera au fichier **ACCDB** que vous auriez sélectionné.

 **Le fichier sélectionné est bien entendu celui qui fait office d'Application (Frontale).**

Pour ce qui concerne la base dorsale (*si toutefois votre base est scindée*) vous devrez l'inclure dans le package durant l'**étape n°4** s'il s'agit d'une application vouée à être utilisée par une seule personne ou plus exactement, sur un seul poste de travail.

En effet, vous ne pouvez pas stipuler que vous souhaitez installer un ou plusieurs fichiers sur un **lecteur réseau** depuis l'Assistant mais uniquement des **sous-répertoires**.

 *Il est dans bien des cas, recommandé de mettre en place une routine de **première utilisation** qui se traduit par la mise en œuvre d'une fonction qui référence et installe les tables liées à l'aide ou non de l'utilisateur au moment du premier chargement de l'application.*  
*Ici, toutes les fantaisies sont ouvertes :*

- concevoir une table locale qui contient l'ensemble des tables liées ;
  - inscrire le nom des tables dans un **fichier INI** ou dans le **Registre**
  - [...]
- et mettre en place le jeu de fonctions idoines pour le rattachement dynamique des tables.

Dans notre exemple, l'application *Frontale* sera installée dans le dossier **Racine** choisi et la Base de données *Dorsale* ne contenant que les tables, pourra être installée dans le sous-dossier **Data** ;

## 5-2-5-2 - Choix du dossier d'installation

Vous devez déterminer le chemin initial d'installation du programme appelé aussi **Dossier racine**.

### **Sélection du dossier Racine**

La liste déroulante proposant les dossiers racines est composée de 6 éléments :

- User AppData (ex : C :\Users\Argyronet\AppData\Roaming) ;
- Program Files (ex : C :\Program Files (x86)) ;
- Mes Documents (ex : C :\Users\Argyronet\Documents) ;
- Bureau (ex : C :\Users\Argyronet\Desktop) ;
- All Users (ex : C :\Users\All Users) ;
- Autre...


Vous pouvez ici choisir le dossier racine souhaité parmi ceux proposés dans cette liste.

Si vous choisissez l'élément **Program Files** le dossier d'installation a l'inconvénient de devenir en lecture seule.

Vous déterminez ensuite le sous-répertoire. Par défaut, le nom du **sous-répertoire** est celui que porte le **nom de votre application**.

Vous pouvez bien entendu le modifier.

Tant que les informations dans la **première zone** et la **troisième zone** sont incomplètes, la mention :

 **L'application source, le chemin cible et le sous-dossier cible sont requis.**  
persiste et bloque par la même, l'accès au bouton **Suivant**.



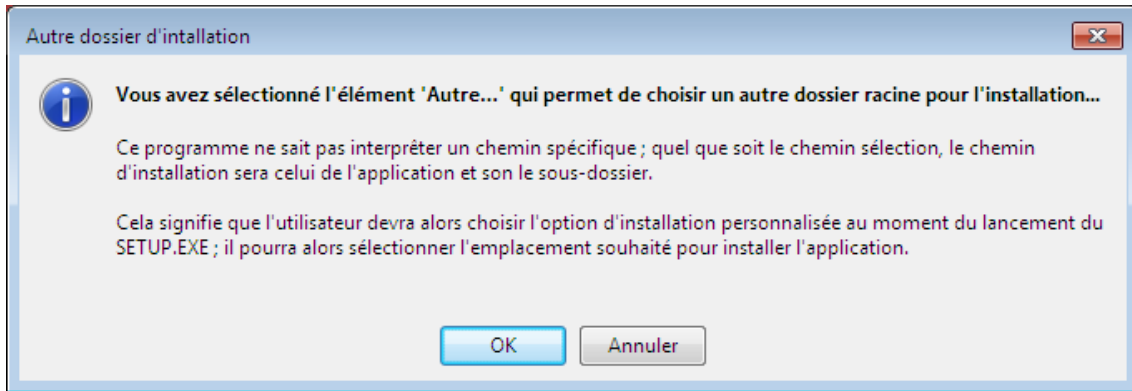
### 5-2-5-2-1 - Le choix du dossier Racine Autre.

Dans cette version de l'assistant, vous avez la possibilité de choisir un dossier racine «**Autre.**»

*En effet, nombreuses furent les fois où vous avez sollicité le forum pour savoir comment installer l'application dans un autre dossier que ceux proposés dans la liste.*

Bon, ce n'est pas encore au point à 100 % mais ça permettra aux utilisateurs de faire en sorte d'accéder à l'**option personnalisée** au moment de l'installation pour choisir le chemin souhaité.

Lorsque que vous choisissez l'élément de sous-dossier «**Autre.**», un message vous prévient en conséquence.



### 5-2-5-3 - Choix de l'inclusion du Runtime.

C'est également ici que vous allez pouvoir déterminer si vous souhaitez inclure ou non le **Runtime Microsoft Access 2013**.

**Si Microsoft Access 2013 est présent sur le poste cible**, ne cochez pas la case.

*Cette version de l'Assistant 2013 ne propose pas comme dans la version 2010 les trois options.*

*En cas de doute, il vaut mieux ne pas cocher la case et attendre les retombées de non-fonctionnement car si le Runtime est installé, il prendra la liberté de s'octroyer l'ouverture de tous fichiers Access ; si peu que l'installation soit effectuée sur un poste où une version complète autre que 2013 existe, ça risque de mettre la pagaille.*

En fait, il est considéré que si vous cochez la case, cela sous-entend que vous savez que le ou les postes devant recevoir votre package d'installation **ne possèdent pas de version d'Access 2013**.

Dans la partie basse de la seconde zone figure la zone de texte où est stipulée :

#### Chemin de stockage du runtime

C : \Users\Argyronet\AppData\Roaming\Microsoft\AddIns\pkwRuntime\AccessRuntime\_fr-fr\_x86.exe

qui est le chemin où est stockée la **version française** et anglaise du **Runtime Access 2013** si toutefois vous ne les avez pas supprimées.

Dès qu'un chemin est stipulé, c'est que le fichier a été localisé.

Si vous cochez la case et que la zone de texte est vide, vous devrez définir l'emplacement par le biais du bouton **Parcourir** afin de préciser l'emplacement du programme exécutable du Runtime Access 2013.

Mais normalement, vous n'avez pas à modifier cette option si vous avez installé cet assistant de façon conventionnelle.

**i** *Contrairement à la version 2010 du **Solution de Package**, le fait de cocher la case suppose que vous voulez distribuer le Runtime avec votre package mais ne provoque pas le changement du nom du fichier de l'application en ACCDR.*



#### Choix du Runtime dans l'Assistant Solution de Package 2010

**!** *Toute application ou tout programme Microsoft installés sur un poste de travail sont soumis à une licence. Bien que le **Runtime Microsoft Access** soit gratuit et permet la distribution et l'exploitation de vos applications développées à travers un mode de fonctionnement équivalent*

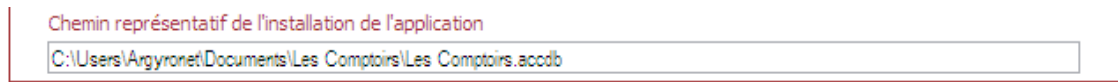
à celui de la **version complète de Microsoft Access**, il n'en est pas moins doté d'une licence qui se traduit par l'acceptation des termes de celle-ci lors de son installation.

### 5-2-5-4 - Résumé de l'application à emballer

Une fois la **base de données (Application)** à emballer renseignée, le **Dossier d'installation racine** sélectionné, et le **Sous-dossier d'installation** stipulé, la zone intitulée :

**Chemin représentatif de l'installation de l'application**

vous précise le chemin représentatif qui illustre l'emplacement de l'application une fois le package installé.

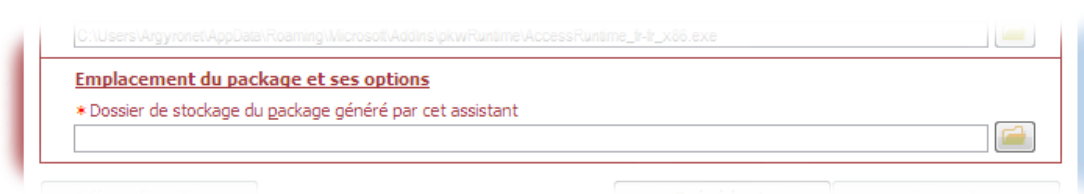


### 5-2-5-5 - Sélection de l'emplacement du package d'installation

Dans la troisième zone de cette étape, la zone **Emplacement du package** vous demande de renseigner le chemin dans lequel l'Assistant va stocker votre package une fois celui-ci généré.

À l'instar de la sélection de l'application, cliquez sur le bouton **Parcourir** pour sélectionner ce chemin.

Une fois toutes les options d'installation définies, la boîte de dialogue enrichie doit être similaire à celle ci-dessous.



Lorsque vous avez terminé, cliquez sur **Suivant** pour passer à l'**étape n°3**.

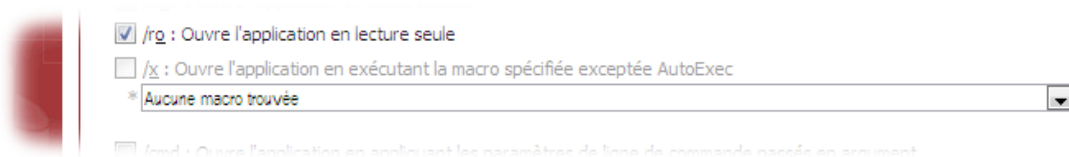
### 5-2-6 - Étape 3/6 - Détermination des options de raccourcis de l'application

Dans cette fenêtre, nous allons considérer comme précédemment qu'elle est subdivisée, en deux zones cette fois :

- première zone : **Emplacement du raccourci**
- seconde zone : **Paramètres supplémentaires**

Les **options de raccourcis** se définissent par **2 cases à cocher** et **2 zones de texte**.

Les **options paramètres supplémentaires** se définissent par **5 cases à cocher** et **2 zones de texte**.



#### **Options de raccourcis**

- **Menu Démarrer** : définit si vous souhaitez qu'il existe un raccourci dans le **Menu Démarrer**.
- **Bureau** : définit si vous souhaitez qu'il existe un raccourci sur le **Bureau**.
- **Nom du raccourci** : définit le **nom** que vous souhaitez donner à votre **raccourci** (pour qu'il soit plus parlant que le nom du fichier ACCDE.)
- **Icône** : vous permet de sélectionner une **icône** pour votre raccourci qui est par défaut celui de l'application Access définie en fonction de l'extension du fichier.

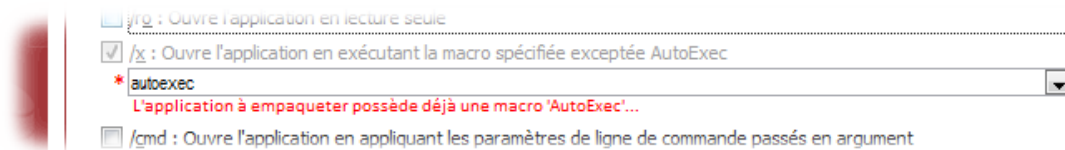
## Paramètres supplémentaires

- /excl : ouvre l'application en **mode exclusif**
- /ro : ouvre l'application en **lecture seule**
- /x : ouvre l'application en exécutant la **macro spécifiée** exceptée AutoExec
- /cmd : ouvre l'application en appliquant les **paramètres de ligne de commande** passés en argument

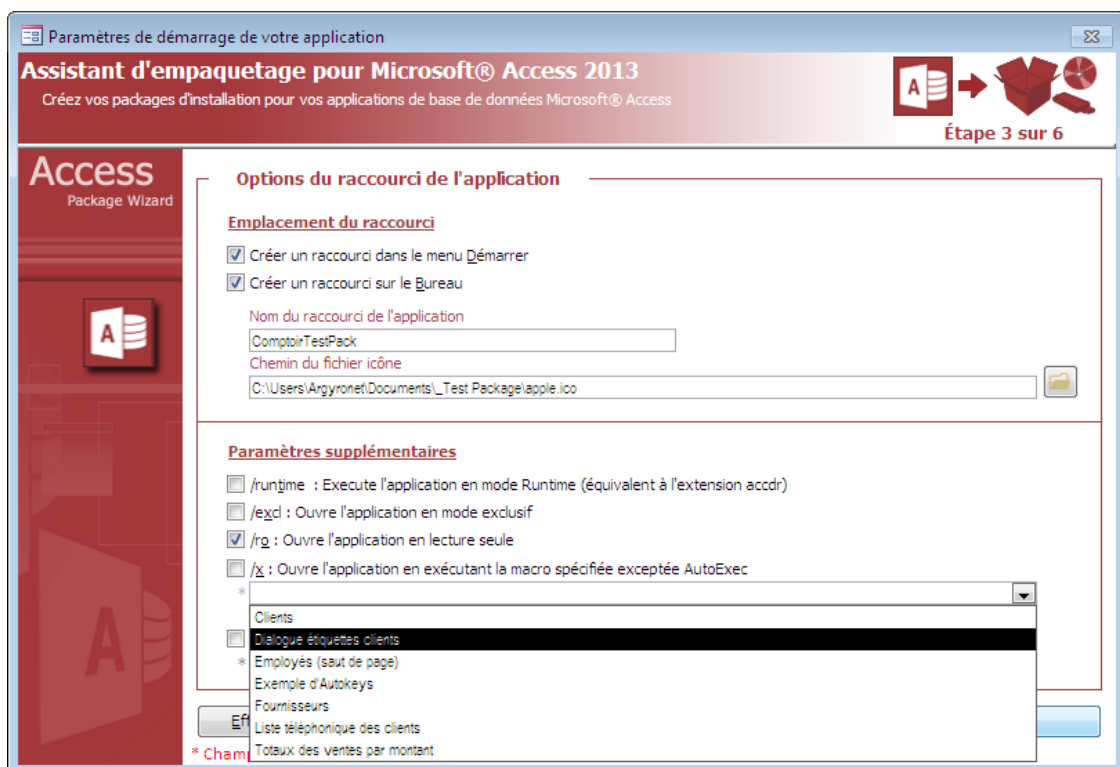
## Les macros

Cet assistant peut lister les macros de votre application, ce qui vous permet de choisir une macro particulière autre que **AutoExec**, qui dans ce cas ne doit pas exister dans la liste.

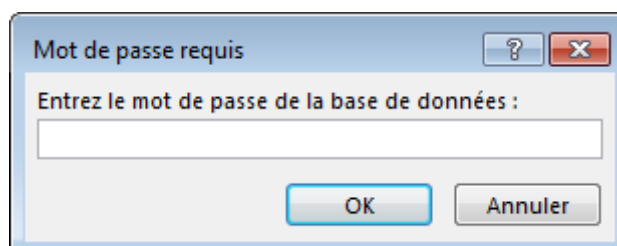
Si la macro **AutoExec** existe, alors la possibilité de choisir une macro de démarrage est verrouillée et cette macro est choisie par défaut.



Sinon, la liste est disponible et vous permet de sélectionner la macro que vous souhaitez affecter au raccourci.

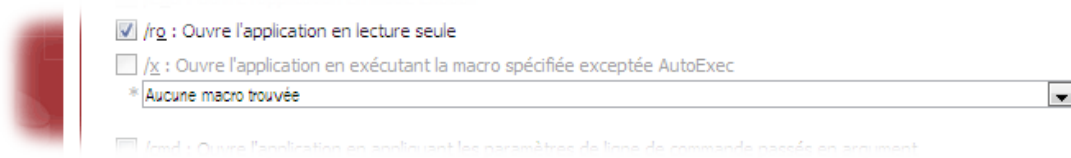


Si votre application est protégée par un mot de passe, il vous sera demandé de le fournir si toutefois vous souhaitez disposer de la liste de vos macros.





Si aucune macro n'existe, le formulaire vous l'affiche en conséquence.



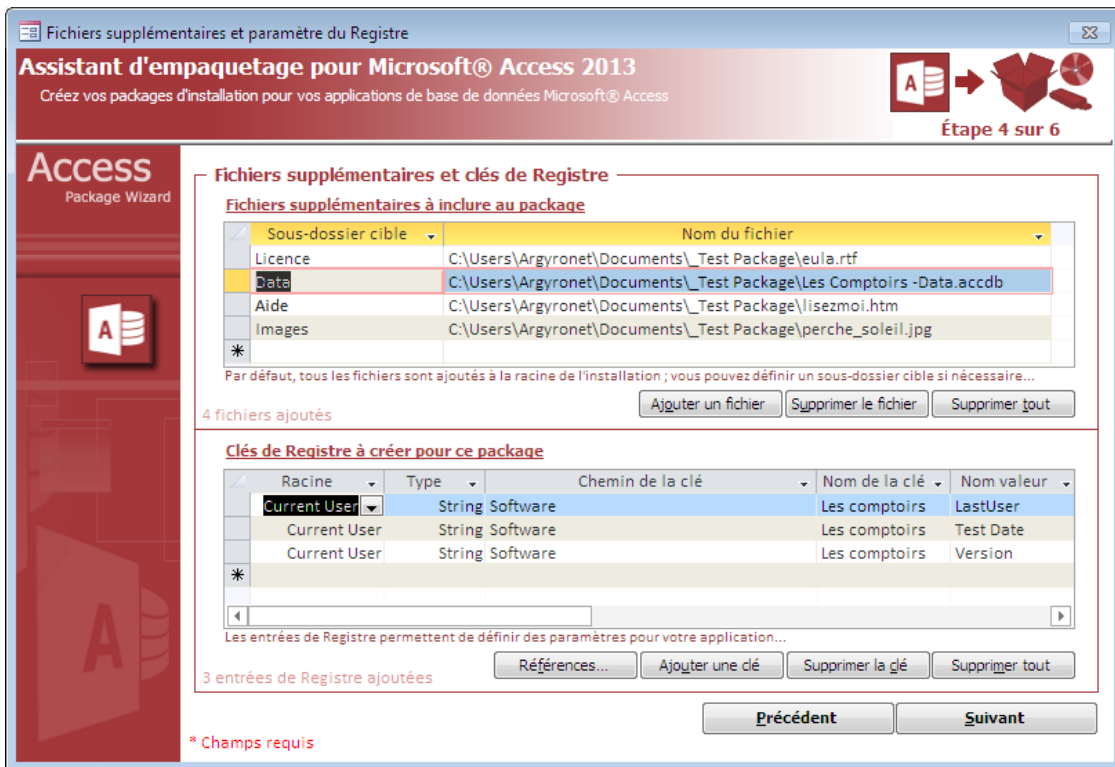
**La ligne de commande**

Inscrivez dans cette zone le paramètre de commande qui permettra à votre application de réagir en conséquence de la fonction **Command()**.

**5-2-7 - Étape 4/6 - Sélection des fichiers et définition des clés de Registre à ajouter à votre Package**

Dans cette fenêtre, nous allons considérer comme précédemment qu'elle est subdivisée en deux zones :

- Première zone : **fichiers supplémentaires à inclure au package**
- Seconde zone : **clés de Registre à créer pour ce package**



**Fichiers supplémentaires à inclure au package**

Si toutefois vous avez choisi **d'inclure le Runtime** dans le package, le fichier exécutable du Runtime sera ajouté dans le dossier «Files» du package d'installation mais ne sera pas intégré dans le MSI.

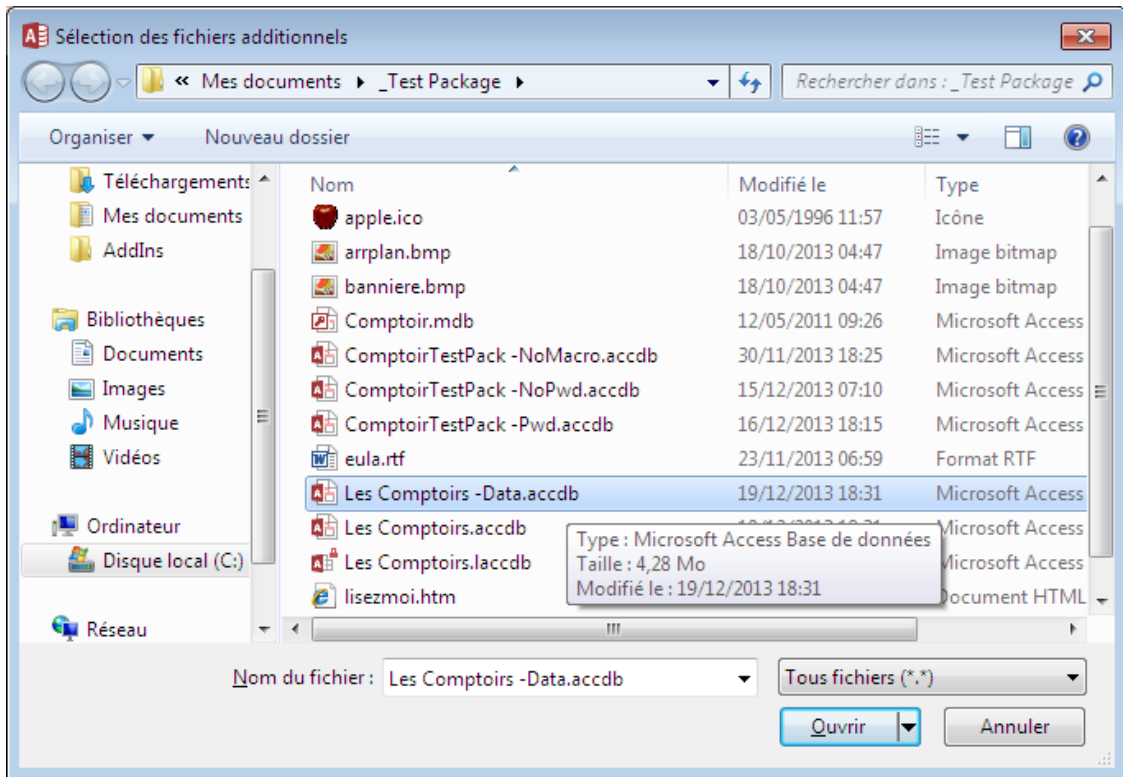
En revanche, les autres fichiers qui concernent plus particulièrement votre application, à savoir les fichiers d'aide, les éventuels documents annexes, les images, les icônes et tout autre type de fichiers dont votre application a besoin pour fonctionner ne sont pas prévus dans le package.

Il vous faut alors les inclure **manuellement** en les sélectionnant comme vous avez sélectionné préalablement le fichier à empaqueter.

- Pour pouvoir ajouter les fichiers il suffit de cliquer sur le bouton **Ajouter (Add)** et sélectionner, ensemble, les fichiers que vous voulez ajouter au package.



- Pour pouvoir supprimer un fichier qui a été ajouté à la liste il suffit de cliquer sur le bouton **Supprimer** (*Remove*) une fois l'élément sélectionné dans la liste.
- Vous pouvez également vider la liste complète en cliquant sur le bouton **Supprimer tout**.
- Si vous tentez d'ajouter un fichier déjà présent, même nom et même emplacement, un message s'affiche.



Une fois que vous avez sélectionné tous les fichiers à inclure avec votre package, ceux-ci apparaissent un à un dans la liste avec respectivement leur **chemin source**, leur **nom** précédé d'une zone dédiée à **l'installation dans un sous-répertoire spécifique**.

Lorsque vous sélectionnez les fichiers à inclure dans votre package, ceux-ci se trouvent dans un certain répertoire : par défaut, ces derniers seront installés dans le dossier de l'application, même s'il s'agit de composants tels que des *librairies* ou assimilés.

La **source** dans laquelle ils ont été puisés ne définit pas la **destination**, contrairement à la version de l'outil d'empaquetage 2000 (*pour les composants et librairies*) que, de loin, je considère être le mieux conçu, en tout cas sur ce point.

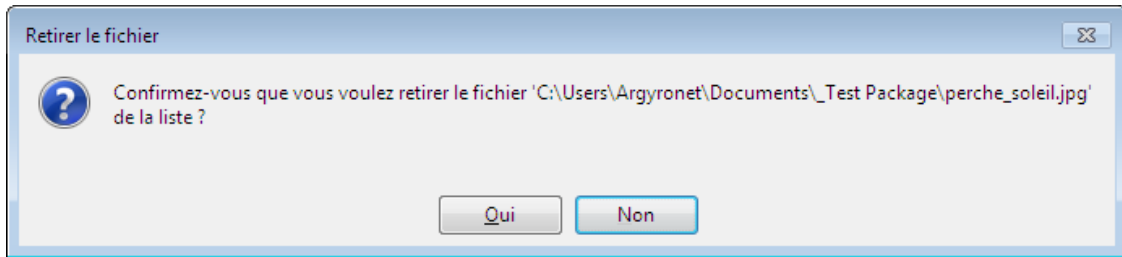
Donc, vous pouvez définir les sous-dossiers cibles pour chacun de vos fichiers inclus : par exemple, définir que vous souhaitez stocker les fichiers d'aide dans le dossier **.\Aide** et/ou les images dans le dossier **.\Images**.

#### Structure du chemin d'installation

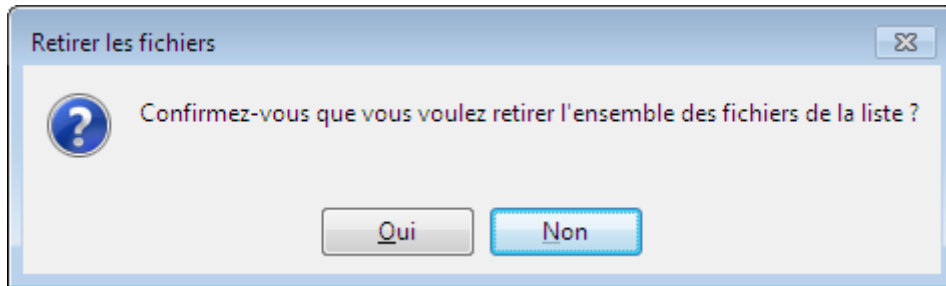
```
%Lecteur % \ %Dossier % \ %Nom du répertoire de votre Application % \ %Nom du sous-répertoire %
```

Dans l'exemple illustré, l'application exploite ses données depuis le sous-dossier **.\Data**. Si les fichiers sources sont situés dans différents répertoires, il vous faudra procéder à l'ajout de façon successive. Si vous ne spécifiez aucun sous-dossier d'installation, les fichiers ajoutés sont copiés dans le dossier racine de l'application

Vous pouvez supprimer un à un les fichiers qui ne sont plus à inclure (*cas de modification*). Cette action est confirmée par un message.



Vous pouvez aussi supprimer tous les fichiers qui composent la liste. Cette action est elle aussi confirmée par un message et toutes deux sont irréversibles si le package est sauvegardé à l'étape 6.



### Clés de Registre à créer pour ce package

Pour la partie concernant les **Clés du Registre supplémentaires**, un niveau **expérimenté** est requis. Pour plus d'informations concernant la Base de Registre, il vous faut comprendre le concept et la structure de cette dernière. Si toutefois vous ignorez totalement ou partiellement son fonctionnement, je vous conseille de laisser les options par défaut si vous ne savez pas comment procéder.

Il se peut que votre application ait besoin d'accéder à certaines entrées du Registre qui ne peuvent pas être ajoutées par votre application avant son installation.

De ce fait, le programme d'installation vous permet d'ajouter des entrées spécifiques connues à l'avance.

- Les clés racines sont **HKEY\_CURRENT\_USER** ou **HKEY\_LOCAL\_MACHINE** selon votre besoin.
- Dans la partie **Chemin de la clé**, inscrivez le chemin complet sans terminateur :

#### Exemple d'entrée de Registre

```
Software\Argyronet Créations\Outils Office
```

- La colonne **Nom de la clé** précise quelle clé ajouter au chemin en amont
- La colonne **Nom valeur** précise quelle valeur ajouter (*partie droite de l'éditeur*)
- La colonne **Valeur** précise la donnée correspondant à la colonne *Valeur*.

Exemple concret :

```
HKEY_LOCAL_MACHINE\SOFTWARE\Software\Argyronet Créations\Office\15.0\Access\Wizards\Assistant Package 2013\ [Installed] | 1
```

Dans la spécification ci-avant, j'ai défini la **Valeur Installed** dans la section **HKEY\_LOCAL\_MACHINE** à la clé **Assistant Package 2013** du chemin **SOFTWARE\Software\Microsoft\Office\15.0\Access\Wizards**.

### Attention !

Ne définissez pas d'emplacement approuvé ici.

La définition est définissable à l'étape 6.

**!** Définir un **Emplacement Approuvé** permet de faire en sorte que votre application ne soit pas vérifiée par le **Centre de gestion de la confidentialité**.

Cela s'applique particulièrement aux applications comportant des modules VBA, des connexions de données et des contrôles ActiveX (contenu actif).

Si vous souhaitez définir votre application comme provenant d'une source fiable, vous devez cocher cette case afin qu'un emplacement approuvé soit défini en fonction du chemin d'installation spécifié.

## 5-2-7-1 - Le package n'inclut et n'inscrit pas les bibliothèques référencées

Eh oui, ici aussi (**voir section** ), je n'ai pas encore prévu cette fonctionnalité.

**Celle-ci pourra faire l'objet d'une évolution dans une prochaine version.**

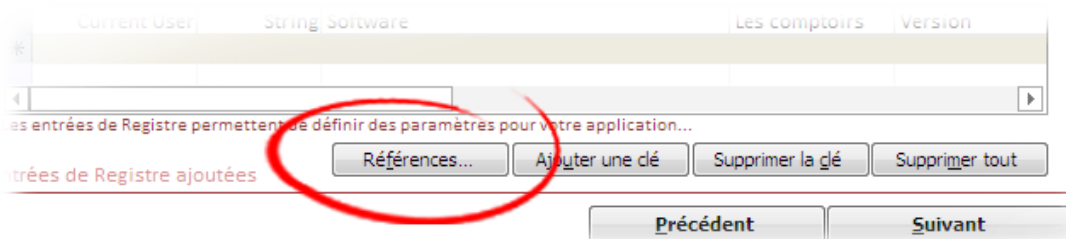
Toutefois, il peut s'avérer un peu superflu d'avoir à inscrire des composants pour cette version de Microsoft Access 2013 et ce, depuis la version 2007.

En effet, à l'époque, il était nécessaire de fournir des composants spécifiques de type **OCX** pour bénéficier par exemple du contrôle de texte enrichi (*RECHED32.OCX*) et du contrôle calendrier (*MSCAL.OCX*) pour les plus connus et les plus usités d'entre eux. Et ces derniers devaient être inscrits dans le Registre de Windows pour éviter l'erreur **429**.

Ces deux fonctions faisant désormais partie intégrante de Microsoft Access, il n'est plus nécessaire de distribuer ces composants qui, par ailleurs font l'objet du désagrément d'être «*killbité*» (*excusez mon anglicisme, mais il n'y a guère de traduction ad-hoc*).

Dans ce formulaire, j'ai ajouté un bouton vous permettant de visualiser les références de votre application.

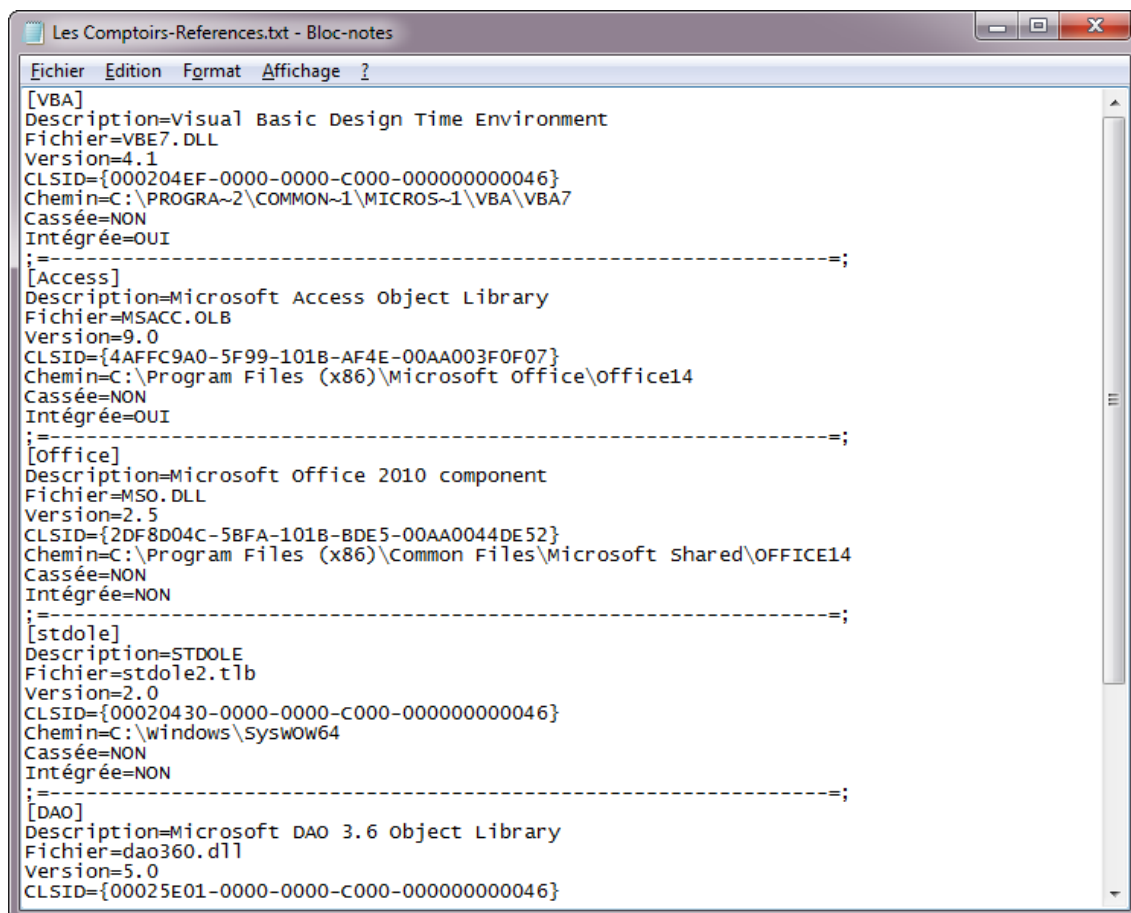
💡 Cela peut s'avérer nécessaire si vous avez besoin de connaître les caractéristiques d'une référence et vous aider éventuellement dans l'alimentation de la grille de gestion des entrées de Registre. Par ailleurs, vous avez la possibilité d'exporter la liste afin de pouvoir récupérer plus aisément des informations difficiles à mémoriser.



- Si votre base de données est protégée par mot de passe et que ce dernier a été renseigné durant l'étape précédente, il ne vous sera pas redemandé.
- Si vous avez refusé de le fournir, il vous sera demandé dans cette étape pour obtenir les références.
- Si vous ne le fournissez toujours pas, eh bien la liste ne sera pas visible.
- Sinon, après un court chargement, le formulaire liste les références comme suit :

Nom	Description	Fichier	Version	CLSID	Chemin	Cassée	Intégrée
VBA	Visual Basic Design Time Environment	VBE7.DLL	4.1	{000204EF-0000-0000-C000-000000000046}	C:\PROGRA~2\COMMON~1\MICROS~1\VBA\VBA7	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Access	Microsoft Access Object Library	MSACC.OLB	9.0	{4AFFC8A6-5F98-101B-AF4E-00AA0030F07}	C:\Program Files (x86)\Microsoft Office\Office14	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Office	Microsoft Office 2010 component	MSO.DLL	2.5	{2DF8D04C-5F9A-101B-BDE5-00AA0044DE52}	C:\Program Files (x86)\Common Files\Microsoft Shared\OFFICE14	<input type="checkbox"/>	<input type="checkbox"/>
stdole	STDOLE	stdole2.tlb	2.0	{00020430-0000-0000-C000-000000000046}	C:\Windows\SysWOW64	<input type="checkbox"/>	<input type="checkbox"/>
DAO	Microsoft DAO 3.6 Object Library	dao360.dll	5.0	{00025E01-0000-0000-C000-000000000046}	C:\Program Files (x86)\Common Files\Microsoft Shared\DAO	<input type="checkbox"/>	<input type="checkbox"/>
Scripting	Microsoft Script Runtime	scrun.dll	1.0	{420B2830-E718-11CF-893D-00A0C9054228}	C:\Windows\SysWOW64	<input type="checkbox"/>	<input type="checkbox"/>

Dans ce formulaire, vous pouvez exporter la liste des références dans un fichier texte ; vous pouvez alors récupérer aisément des informations relatives à telle ou telle référence.



```

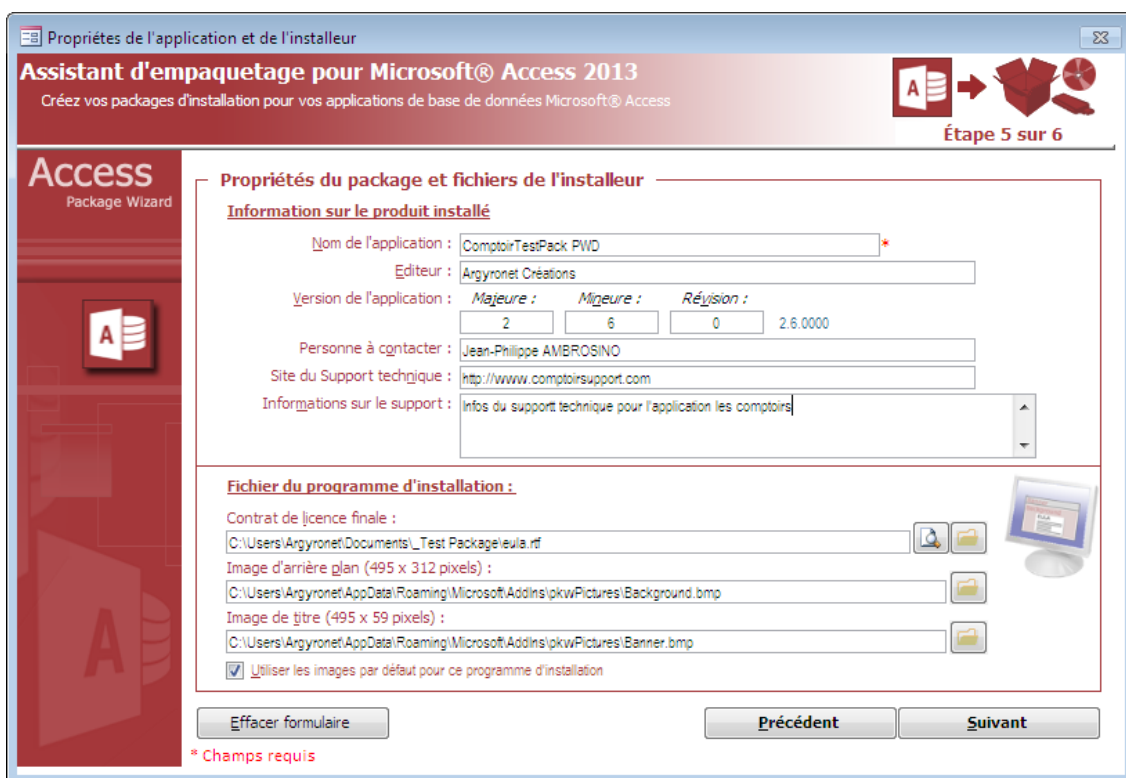
[VBAA]
Description=Visual Basic Design Time Environment
Fichier=VBE7.DLL
Version=4.1
CLSID={000204EF-0000-0000-C000-000000000046}
Chemin=C:\PROGRA~2\COMMON~1\MICROS~1\VBA\VBA7
Cassée=NON
Intégrée=OUI
;=====;
[Access]
Description=Microsoft Access Object Library
Fichier=MSACC.OLB
Version=9.0
CLSID={4AFFC9A0-5F99-101B-AF4E-00AA003F0F07}
Chemin=C:\Program Files (x86)\Microsoft Office\office14
Cassée=NON
Intégrée=OUI
;=====;
[Office]
Description=Microsoft office 2010 component
Fichier=MSO.DLL
Version=2.5
CLSID={2DF8D04C-5BFA-101B-BDE5-00AA0044DE52}
Chemin=C:\Program Files (x86)\Common Files\Microsoft Shared\OFFICE14
Cassée=NON
Intégrée=NON
;=====;
[stdole]
Description=STDOLE
Fichier=stdole2.tlb
Version=2.0
CLSID={00020430-0000-0000-C000-000000000046}
Chemin=C:\windows\Syswow64
Cassée=NON
Intégrée=NON
;=====;
[DAO]
Description=Microsoft DAO 3.6 Object Library
Fichier=dao360.dll
Version=5.0
CLSID={00025E01-0000-0000-C000-000000000046}

```

### 5-2-8 - Étape 5/6 - Propriétés du package et fichiers de l'installeur

Cette fenêtre est elle aussi subdivisée en deux zones :

- première zone : **Information sur le produit installé**
- seconde zone : **Fichier du programme d'installation**



### Information sur le produit installé

Ce sont les propriétés générales ; vous définissez ici le **nom de l'application**, l'**éditeur**, la **version**, la **personne à contacter** et le **site web** pour le support accompagné d'une zone **d'informations** supplémentaires.

Ces éléments seront utiles et visibles à partir du **Panneau de configuration** dans la rubrique d'installation ou de désinstallation des programmes.

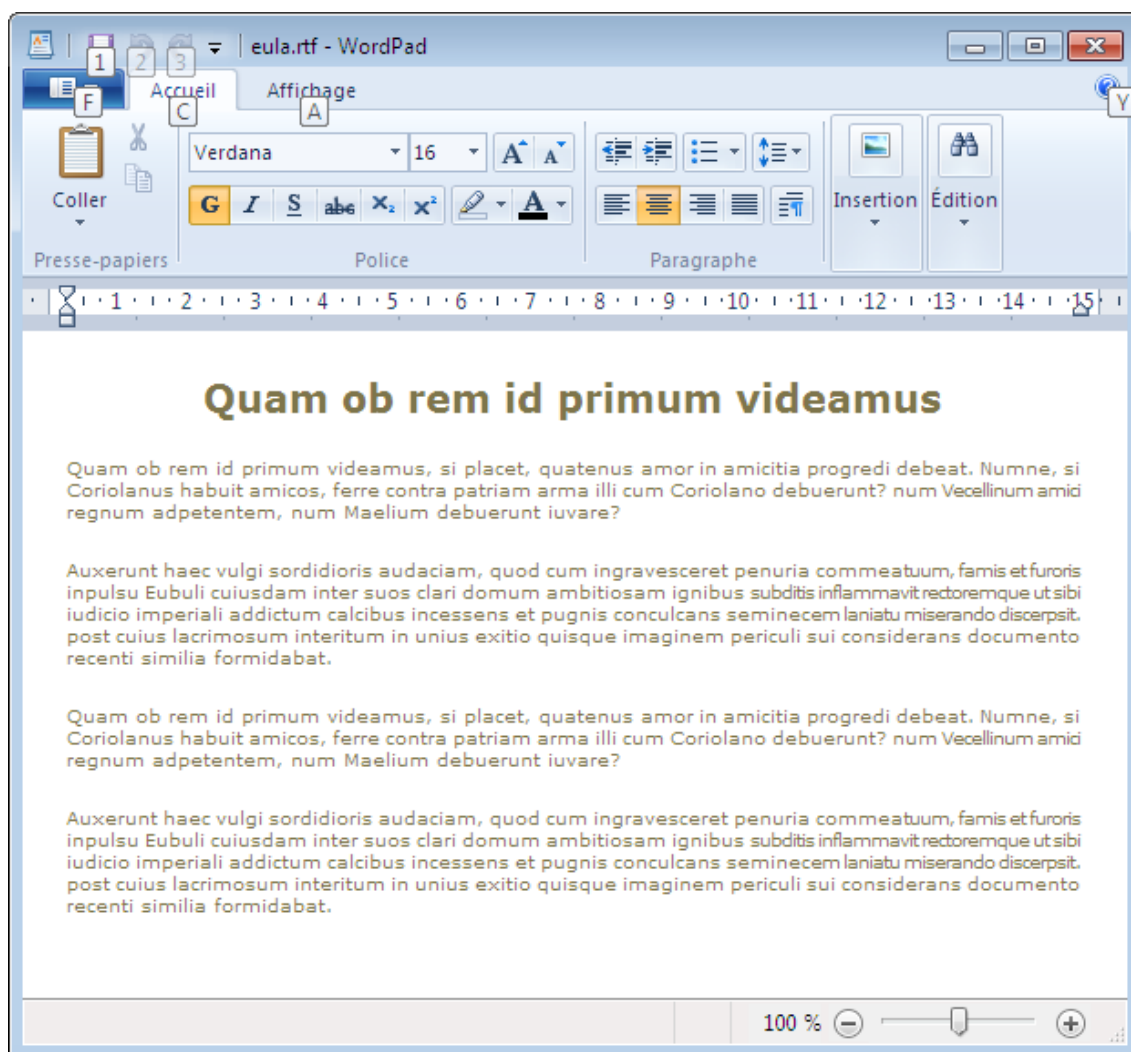
### Fichier du programme d'installation

Dans cette zone, vous devez si nécessaire, spécifier le document faisant office de clauses du contrat de licence de l'utilisateur final **CLUF** (ou **EULA en anglais**).

Cliquez sur le bouton **Parcourir** pour choisir le fichier **RTF** correspondant.

Il apparaîtra lors du lancement du **Setup** au moment où l'utilisateur devra ou non accepter les termes de la licence.

Par ailleurs, un bouton vous permet de visualiser le document RTF en question si toutefois vous vouliez le relire ou le modifier.



Les deux autres zones de texte suivantes représentent respectivement la définition d'une image d'**arrière plan** d'une part, et d'une image de **titre** d'autre part, toutes deux étant exploitées au moment de l'exécution du Setup.exe par l'utilisateur.

- Si vous voulez personnaliser votre **Setup**, choisissez deux images (*assez simples et plutôt pastels*) en respectant les **dimensions**.
- Si vous n'en voulez pas, l'écran du **Setup** restera à *dominante grise*.
- Si vous voulez que le Setup utilise les images **par défaut** fournies avec cet Assistant, **cochez** la case appropriée.

**Les images d'arrière plan et de titre sont fournies (*gracieusement ;o*) avec l'application :**



*Image d'arrière-plan*



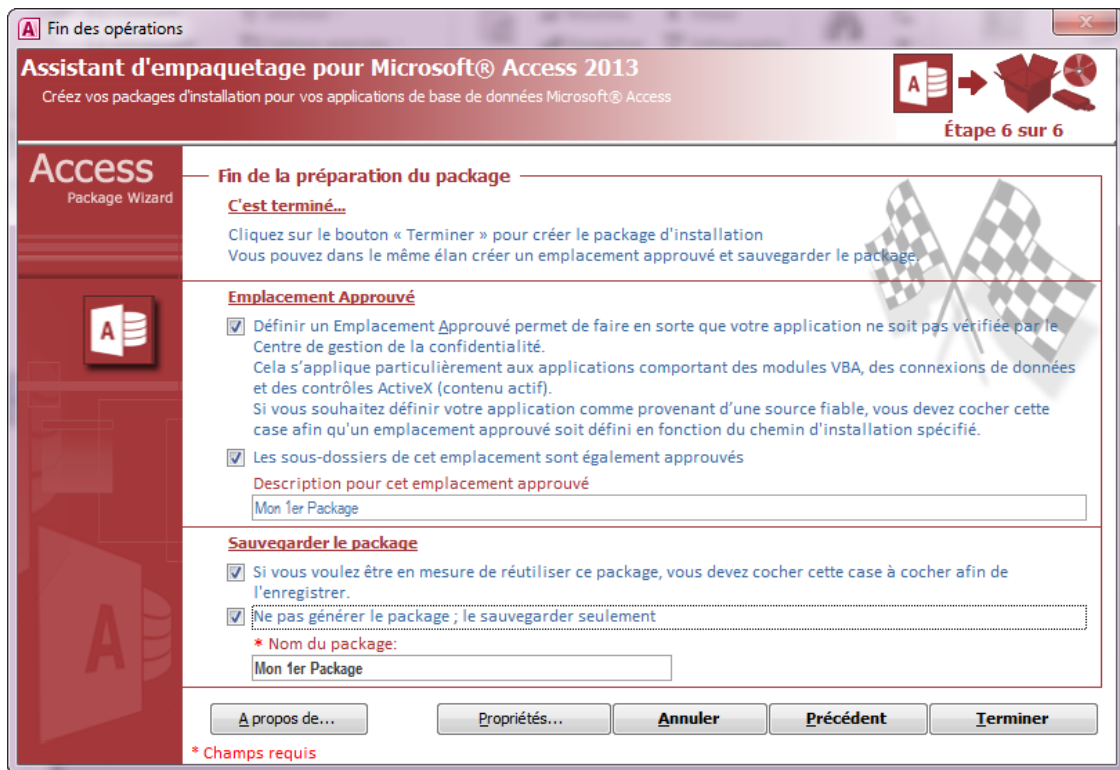
*Image de titre*

## 5-2-9 - Étape 6/6 - Fin de la préparation du package

Cette dernière fenêtre est elle aussi subdivisée en trois zones :

- première zone : **C'est terminé.**
- seconde zone : **Emplacement Approuvé**
- troisième zone : **Sauvegarder le package**





### C'est terminé.

Comme cela se comprend ici, vous avez terminé la préparation de votre Package en précisant les options possibles de cette dernière étape.

### Emplacement Approuvé

Définir un **Emplacement Approuvé** permet de faire en sorte que votre application ne soit pas vérifiée par le Centre de gestion de la confidentialité.

*Dans la version 2010 de l'Assistant Solution de Package était créé par défaut un emplacement approuvé.*

*Mais il n'était pas opérationnel, car il était défini dans la mauvaise clé à savoir **12.0** au lieu de **14.0**.*



*Je m'en suis aperçu que bien tard et n'ai pas pris l'initiative de vous en parler dans le tutoriel de la version **2010** puisque je l'ignorais jusqu'à aujourd'hui.*

*De ce fait, nombreux sont ceux d'entre vous qui avaient sollicité le forum pour connaître la marche à suivre pour en définir un.*

Cela s'applique particulièrement aux applications comportant des modules VBA, des connexions de données et des contrôles ActiveX (*contenu actif*).

Si vous souhaitez définir votre application comme provenant d'une source fiable, vous devez cocher cette case afin qu'un emplacement approuvé soit défini en fonction du chemin d'installation spécifié.

### Sauvegarder le package

Ainsi que je vous en ai parlé au début de ce tutoriel, vous avez la possibilité de sauvegarder le Package de deux manières.

La clé d'enregistrement de votre package étant basée sur son **Nom**, si vous avez créé un **Nouveau Package**, il n'y aura pas d'alerte.

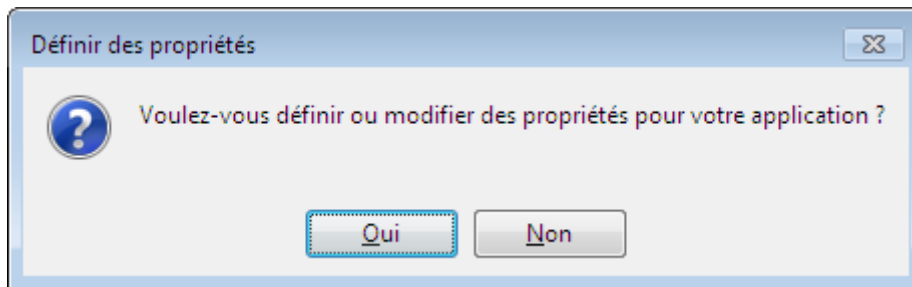
En revanche, si vous avez modifié un package existant, vous avez le choix entre le **sauvegarder (remplacer)** et le **sauvegarder-sous (Nouveau)**, la différence entre ces deux opérations s'établissant en changeant le nom :

- si ce nom n'est pas changé, et après confirmation, le package existant sera mis à jour avec les nouvelles modifications.
- si ce nom est modifié, toujours après confirmation, le package existant choisi sera conservé dans son état initial et le nouveau sera fondé sur celui sélectionné et modifié avec les nouvelles spécifications que vous avez effectuées durant les étapes du processus d'empaquetage.

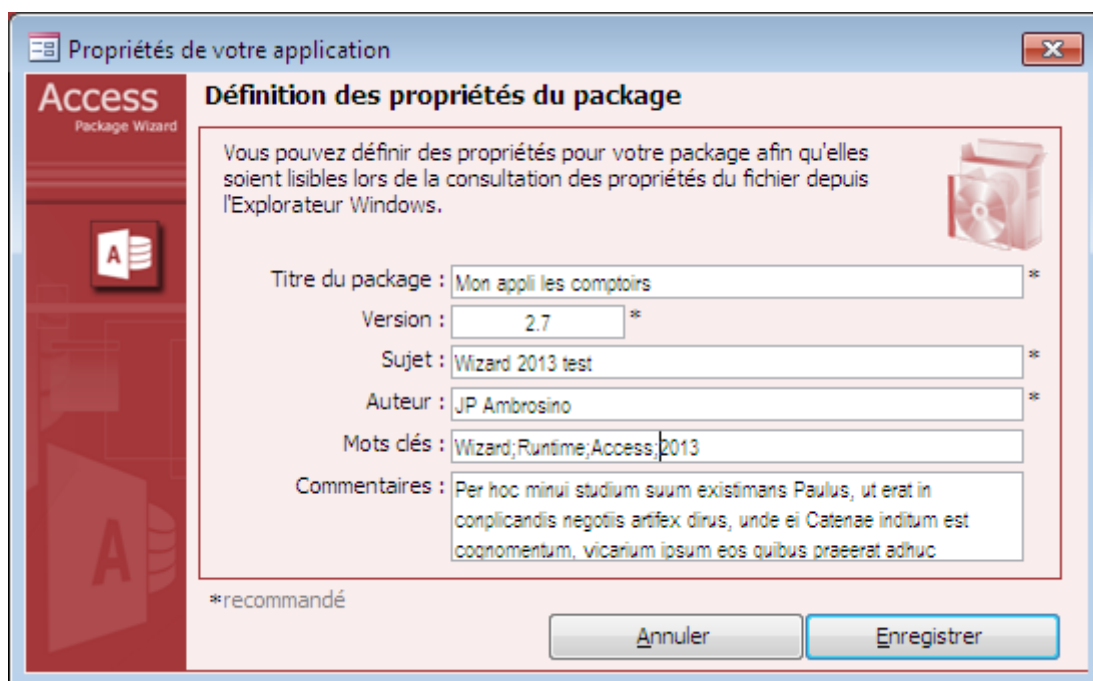
### Propriétés de votre application.

Dans le bas de la fenêtre, vous apercevez le bouton **Propriétés...**

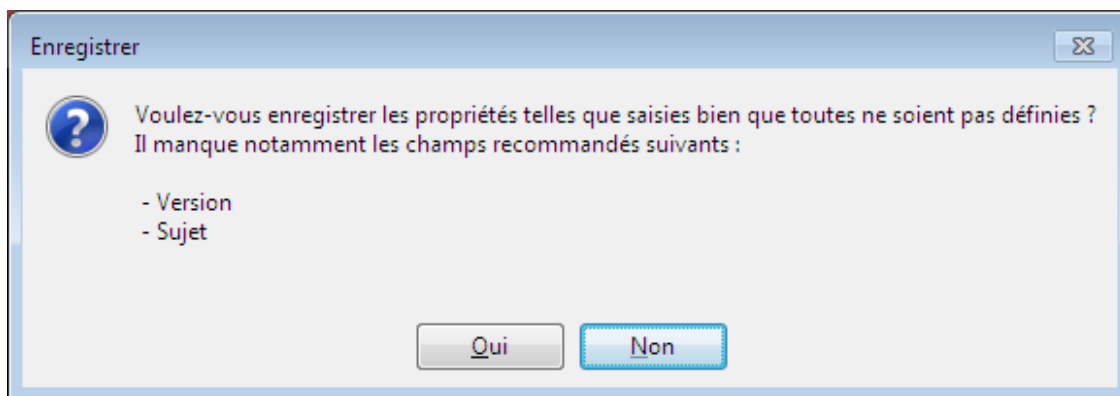
Vous pouvez définir des propriétés pour votre package afin qu'elles soient lisibles lors de la consultation des **propriétés** du fichier depuis **l'Explorateur Windows**.



Cette fenêtre comporte l'ensemble des champs qui pourraient être utiles pour informer quiconque du type et de la provenance du fichier.



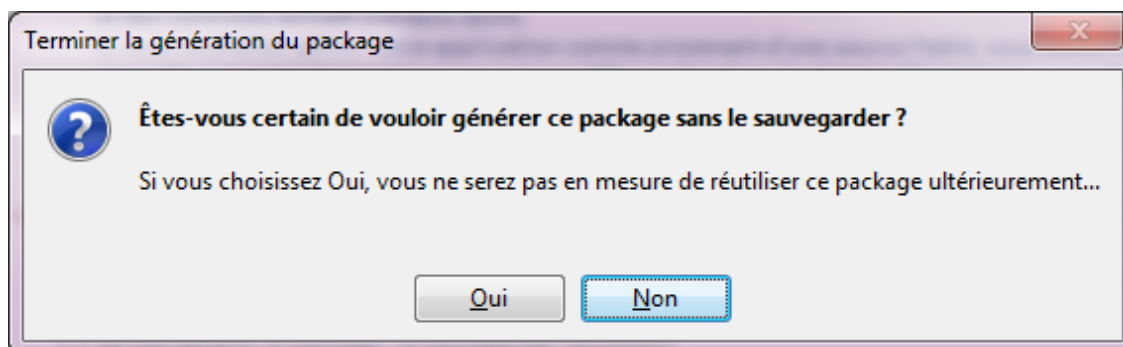
Dans la mesure du possible, si vous souhaitez définir des propriétés pour votre application, essayez de les renseigner toutes.



Lorsque vous avez terminé, cliquez sur **Valider** ; celles-ci resteront alors stockées sauf si vous ne sauvegardez pas le package.

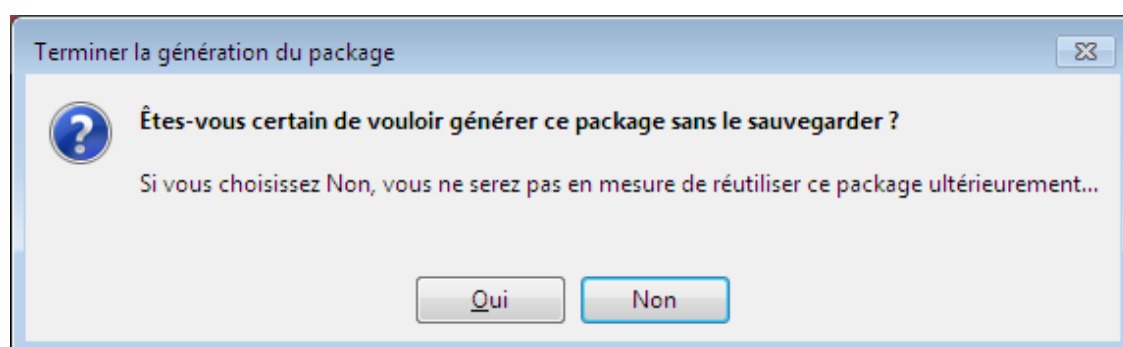
### 5-2-9-1 - Sauvegarde du modèle et génération du Package

Tous les éléments constitutifs nécessaires à la génération de votre package sont maintenant terminés ici. Si vous n'avez pas coché l'option de l'**emplacement approuvé**.



*Message de confirmation de l'emplacement approuvé omis*

ou bien celle de la **sauvegarde**.



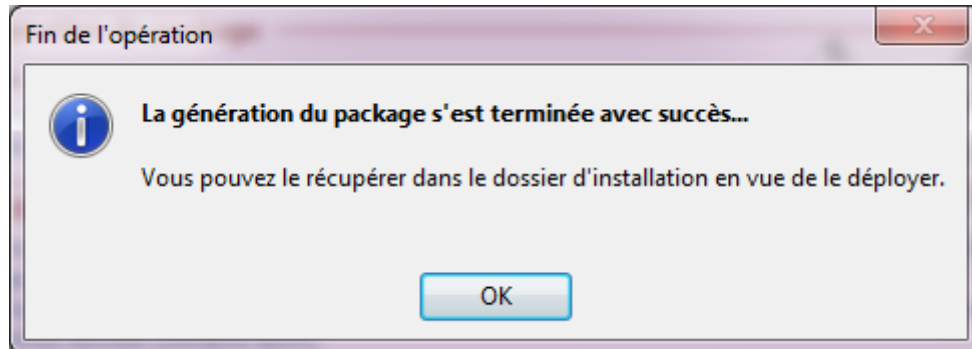
*Message de confirmation de sauvegarde omise*

Cela ne vous empêche pas de générer votre package mais une alerte vous prévient de manière à ce que vous puissiez vous raviser.

Notez que si vous n'enregistrez pas, il n'y a aucun moyen de récupérer la définition du modèle de Package. Cliquez sur le bouton **Terminer** pour générer votre package.

## 5-2-9-2 - Récupération du package (ou non selon l'option cochée)

Si la case de **sauvegarde seule** n'est pas cochée, au bout de quelques secondes, le package est généré dans le dossier que vous avez spécifié.  
Un message de fin de procédure fait son apparition.

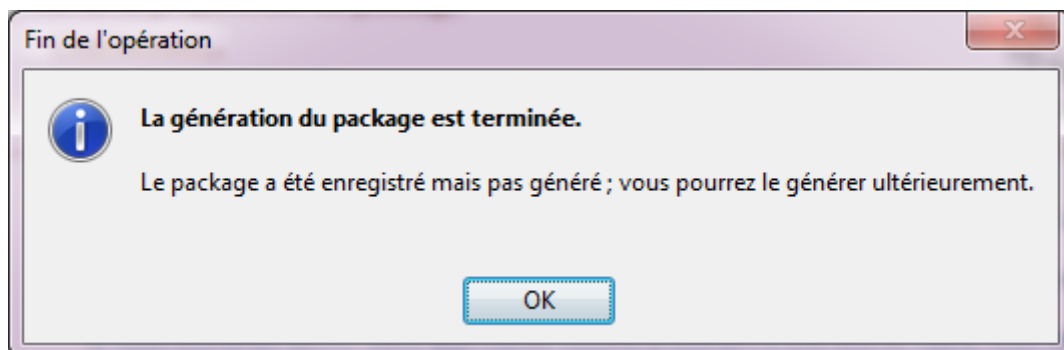


Puis, l'**Explorateur de fichiers** s'ouvre alors sur ce dossier vous présentant ainsi votre package.  
Le dossier visible inclut le fichier **MSI** et tous les fichiers nécessaires à son exécution.

Le modèle de votre package est dès lors sauvegardé.  
Une fois que vous avez cliqué sur **OK**, la fenêtre se ferme et libère l'application Access sur la base de données que aviez ouverte.

Si vous avez coché la case n'autorisant que la sauvegarde, le package est sauvé sans être généré (*nouveauté nov. 2014*).

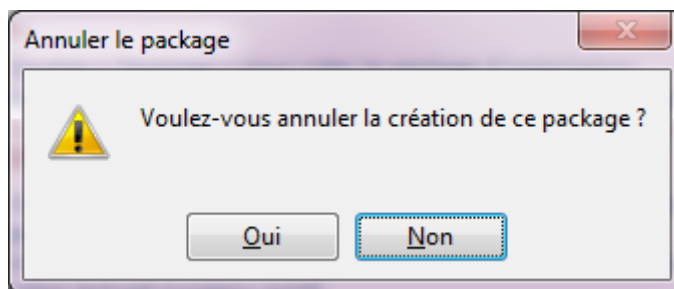
Le message de fin sera alors le suivant :



### 5-2-9-2-1 - Annulation du processus

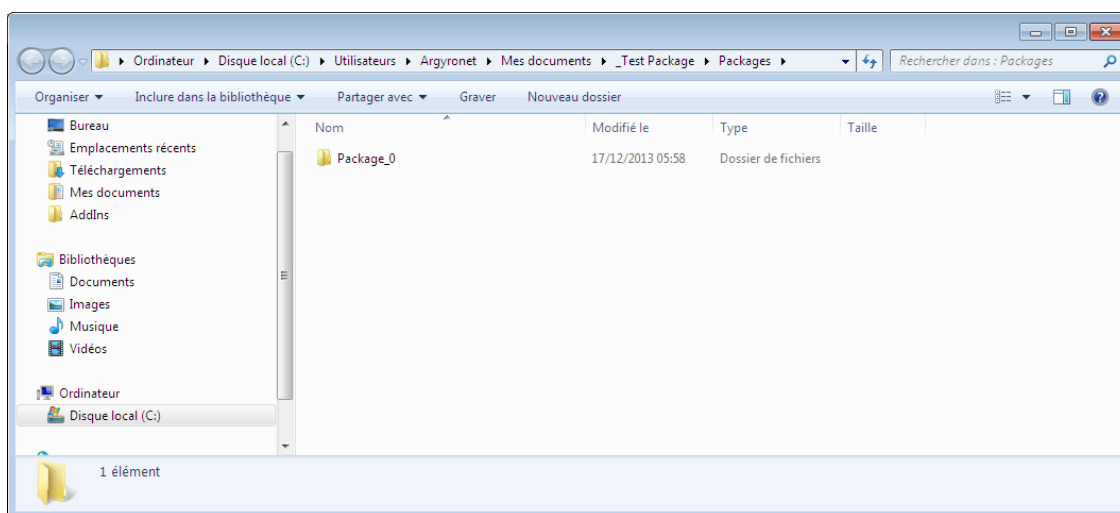
Dans la dernière version 32/64bits, une nouvelle fonction a été ajoutée : elle permet d'annuler la totalité du processus de création du package...

En effet, auparavant, il fallait cliquer cinq fois sur le bouton **Précédent** et c'était fort peu pratique.  
Un message de confirmation est toutefois en place au cas où...



## 5-2-10 - Vérification du Package généré

Dans le dossier que vous avez défini se trouve les fichiers nécessaires à l'installation de votre application.



Dans le sous-répertoire **Files** se trouve le fichier **MSI** qui porte le nom de votre application ainsi que l'exécutable du **Runtime** (dans la langue appropriée) si vous l'avez inclus à votre package.

## 5-2-11 - Le fichier AutoRun pour votre CD-Rom

L'Assistant Package génère systématiquement un fichier **AutoRun** pour votre Package pour le cas où vous proposeriez un package dans un **CD-Rom** pour l'installation.

Même si cela paraît évident, le lancement automatique de l'installation via un CD-Rom requiert que le **Service de Notification d'Insertion** de CD sur le poste client soit activé.

Ce contenu du dossier peut alors être stocké sur la racine du CD-Rom gravé avec votre projet.

### 5-2-10-1 - Pour concevoir un fichier Autorun manuellement

Si toutefois vous souhaitez personnaliser ou modifier le fichier **Autorun**, prenez un éditeur de texte comme **Notepad** ou autre et inscrivez-y au minimum les lignes de code correspondant à vos paramètres comme dans l'exemple suivant :

```
[autorun]
open = Setup.exe
icon = AppIcon.ico
```

Faites en sorte que le programme **setup.exe** tout comme l'icône **AppIcon.ico** se situent sur la racine du CD-Rom si vous stipulez le contenu comme tel.

Vous enregistrez ensuite ce fichier sous le nom **Autorun.inf** et le remplacez par celui existant dans la liste des fichiers de votre programme de gravure de CD-Rom.

## 6 - Évolution de votre Package

Tous les packages que vous avez sauvegardés sont disponibles dans la liste de **l'étape 1** avec la date. Lorsque vous sélectionnez un package existant, vous retrouvez toutes les informations que vous avez renseignées.

### 6-1 - Gestion des packages sauvegardés de l'Assistant

Lorsque vous travaillez avec l'Assistant Package, vous pouvez être amené à redéployer une version de votre application pour la simple raison que cette dernière a évolué ou que vous avez nécessité d'ajouter ou modifier des éléments qui l'accompagnent.

 *L'utilisation des modèles assure un gain de temps intéressant.*

Dès que vous choisissez un package existant, vous pouvez avoir à l'esprit deux initiatives :

- 1°) **Régénérer** un même package avec quelques petites modifications
- 2°) Générer un **nouveau** package à partir d'une sauvegarde du fait que celui-ci est très similaire à celui sélectionné.

#### 6-1-1 - Après avoir sélectionné un package, vous pouvez effectuer diverses opérations :

- Renommer un package
- Créer une copie du package sous un nouveau nom.
- Supprimer un modèle dont vous n'avez plus besoin.

 *Lorsque vous supprimez un modèle d'emballage, l'Assistant ne pourra plus générer de packages créés à partir de ce modèle*

Pour pouvoir déployer un package similaire, vous devrez reconstruire de nouveau ce projet. Veuillez donc à ne supprimer que les modèles dont vous êtes sûr de ne plus avoir besoin.

### 6-2 - Modification ou évolution de votre application

Il est très probable que vous ayez à faire évoluer ou modifier certaines fonctionnalités de votre projet.

Dans ce cas, il faudra le redéployer sur le poste client **si vous avez ajouté des composants ou référencé des bibliothèques qui n'y figuraient pas dans la version précédente.**

Ceci peut être un inconvénient, car il sera alors indispensable de désinstaller l'application initiale sur ledit poste.

En effet, si l'on souhaite travailler *proprement*, il ne faut ne jamais tenter de réinstaller une application sans l'avoir entièrement désinstallée d'abord.

Toutefois, ce générateur de package vous empêchera de le faire puisque le MSI généré est coiffé d'un système de contrôle de l'installation.





**i** Dans les versions antérieures, on ne pouvait généralement plus désinstaller (proprement) une précédente application si plusieurs installations hétérogènes se sont succédées sans qu'elles aient été désinstallées d'abord.

Ici, vous n'avez pas d'autre choix :

- soit vous réinstallez ;
- soit vous supprimez l'application.

Si elle est supprimée, alors vous pourrez envisager d'installer la nouvelle version que vous auriez modifiée.

**i** Il est à noter alors que si votre client ou les postes concernés fonctionnaient avec le Runtime Microsoft Access, la désinstallation de l'application ne supprime pas le Runtime.

Sinon, il est vrai que dans 90 % des cas, la simple fourniture du fichier ACCDE (ou **ACCDB** si vous n'avez pas créé de **ACCDE**) à copier dans le répertoire de destination est suffisante.

Là, vous pouvez donc procéder à la mise à niveau de votre application sur le poste client par la distribution à travers un petit **VBScript** ou un simple fichier **.bat** que vous rédigerez auparavant.

Cela reste une méthode plus rapide, surtout en réseau où de nombreux postes sont tous identiques.

## 6-2-1 - Modification de votre projet

Si vous êtes amené à modifier votre application de Base de données, il vous faut intervenir sur le projet **ACCDB**. Vous ouvrez donc votre base **ACCDB**, apportez les corrections ou les évolutions nécessaires, protégez ensuite votre projet au niveau du code **VBA** même si le projet doit être compilé dans un ACCDE.

De là, vous enregistrez le tout pour facultativement générer un nouveau fichier ACCDE pour vérifier que votre application est sans erreurs et que vous envisagez de nouveau de distribuer un package.


**!** *Humm.*  
Lorsque j'écris sans erreurs, ce n'est pas une règle ! Il est évident qu'un code écrit :

- sans instructions **Option Explicit**,
  - avec des variables **mal typées** voire toutes en **Variant**,
  - des **On Error Resume Next** un peu partout,
  - des passages de paramètres par **référence** (ByRef) au lieu d'un passage par **valeur** (ByVal) sans justification réelle,
  - aucune **gestion d'erreurs** digne de ce nom
  - et..
- va compiler sans problème et pourra générer un **ACCDE** en toute simplicité.

Le nouveau fichier **ACCDE** que vous avez sans doute généré, ainsi que tous les fichiers que vous avez modifiés (fichiers d'aide, images...), sont censés être restés dans le répertoire source d'origine.

Si toutefois vous avez modifié un fichier quelconque dans un autre répertoire, il vous faudra le copier dans le dossier source de l'emplacement défini dans le package afin de remplacer l'existant de manière à ce que l'Assistant Package le prenne en considération.

*Tous les fichiers que vous modifiez dans ce répertoire doivent porter le même nom que les fichiers ayant été ajoutés au package.*

 *Si vous ajoutez physiquement des fichiers absents du modèle de package, vous devrez les ajouter à la liste lors de la prochaine session de l'Assistant Package, sans quoi, ils ne seront pas pris en compte lors de la nouvelle génération du package.*

*Si toutefois vous avez déplacé, renommé ou supprimé un des fichiers quelconques qui constituaient le package, vous recevrez un message d'erreur conséquent qui se traduira par une **erreur 53**.*

## 6-2-2 - Rempaquetage de votre projet modifié

L'Assistant Package 2013 ne permet pas de régénérer rapidement vos jeux de fichiers **MSI** à partir d'un simple fichier **Batch** comme dans les versions 2000 et 2003, ce qui est fort dommage.

Pour régénérer un package, vous devrez alors lancer l'Assistant Empaquetage à partir **d'Access 2013** comme pour une première utilisation mais cette fois, en sélectionnant votre modèle que vous êtes censé avoir sauvé.

Ce fichier modèle doit porter le nom de votre projet ou plutôt celui que vous avez stipulé durant la dernière étape de génération de votre empaquetage. Il est accompagné de la date de génération, ce qui permet de le repérer sans doute plus facilement.

Si vous avez modifié votre application au niveau des DLL et autres composants, il est indispensable de fournir un nouveau **MSI** et non la simple copie du **ACCDE** mis à jour. Vous devrez par ailleurs envisager de modifier le programme d'inscription de ces composants dans la base de registre si toutefois ces composants nécessitent d'être enregistrés comme tels.

Ce programme d'inscription n'existe pas et il vous appartiendra de créer la routine correspondante et faire en sorte qu'elle s'exécute en même temps que l'installation de votre application à travers le package installation.

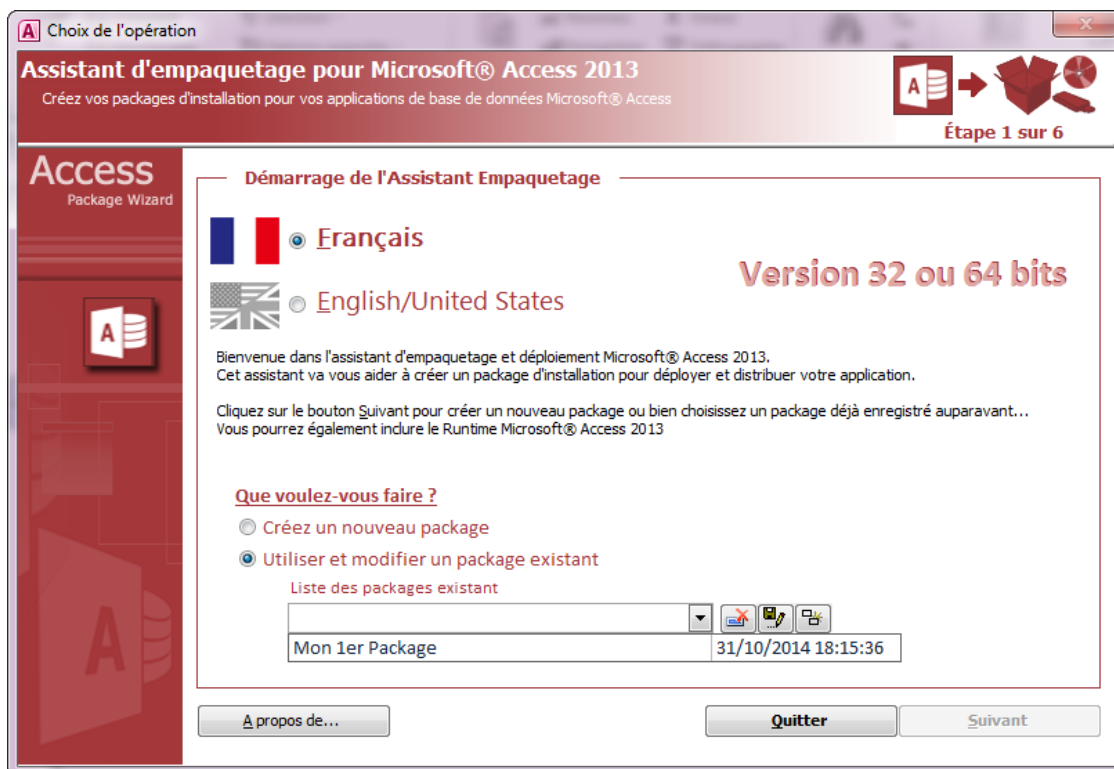
Si nombre d'entre vous le réclament et si le temps me le permet, je tâcherai de greffer dans la version suivante un module d'inscription des composants.

Il faudra enfin adapter le fichier **setup.ini** en greffant une nouvelle installation chaînée comme expliqué dans le **tutoriel 2010** à la section **5-2-4-1-1 Comment mettre en place un autre package ?**

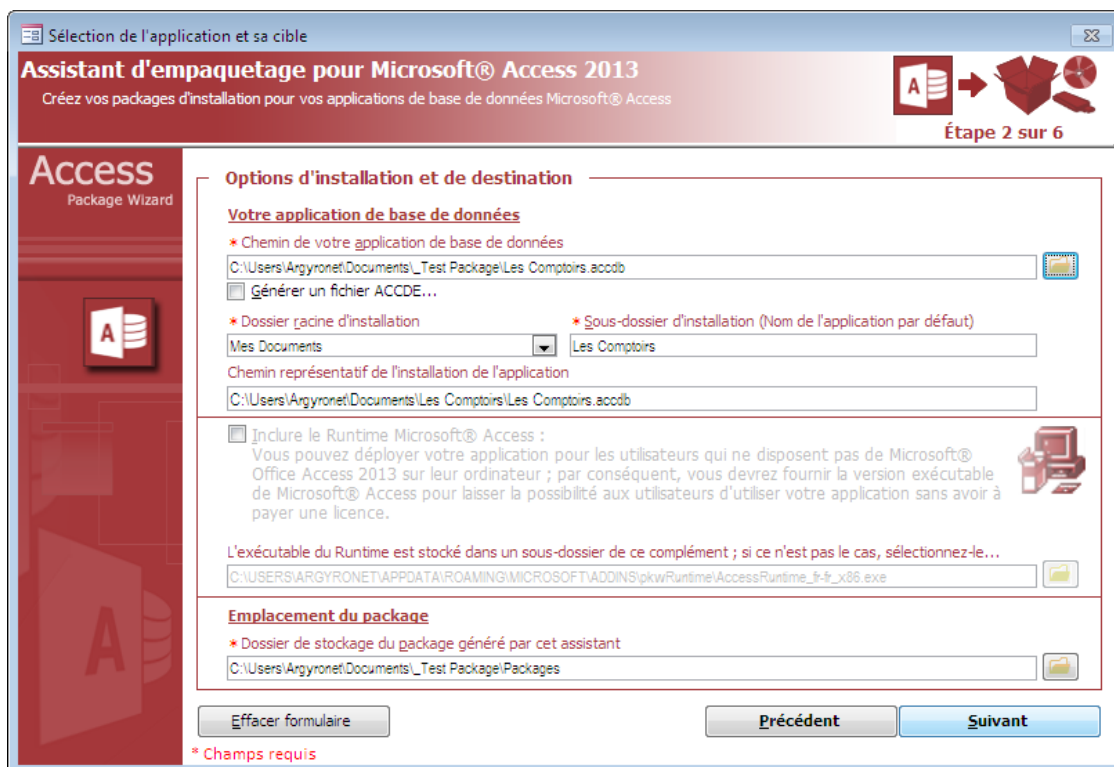
## 6-2-3 - Régénération du package après modification ou évolution

Pour pouvoir régénérer votre package, vous passez par le même déroulement, à savoir l'onglet **Outils de base de données** puis sur l'élément **Compléments** où vous cliquez sur **Package Wizard 2013** comme vu à la section : **Lancement de l'Assistant Package**.

Sélectionnez alors votre package à partir de la liste des **modèles sauvegardés** pour le charger.

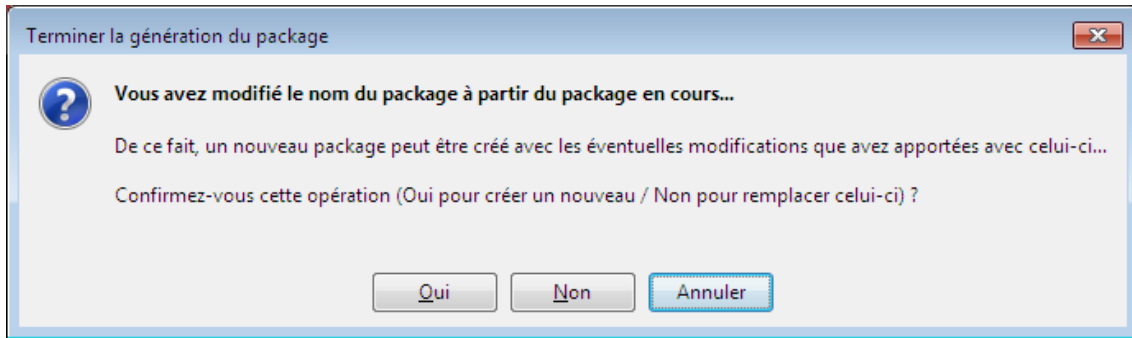


Vous retrouvez alors toutes les informations propres à ce package.



Vous passez en revue les six étapes en apportant *les éventuelles modifications*, notamment sur la version et cliquez sur **OK** pour terminer.

Vous pouvez alors comme nous l'avons vu, **mettre à jour** ce package ou bien **créer une nouvelle** version de ce package en changeant simplement le nom.



Tout comme précédemment, l'**Explorateur Windows** sera affiché avec le contenu du Package. Si le dossier du précédent package existe encore, le nouveau sera **incrémenté de 1**.

### 6-3 - Enregistrement de votre Package sur une source d'installation

Lorsque votre package est prêt, vous pouvez le stocker sur une source de votre choix afin de faire en sorte qu'il puisse être installé sur un poste client. Vous pourrez également inclure d'autres fichiers comme **Acrobat Reader** pour le cas où vous fourniriez un guide d'utilisation au format **PDF**.

En tout état de cause, votre package **doit contenir au minimum** les éléments suivants :

- **Setup.exe**

Le dossier **Files** dans lequel se trouve le fichier **MSI** de votre projet et un dossier **Setup** qui contient un fichier **INI** correspondant.

Et si vous avez prévu de proposer l'installation du Runtime, le dossier Files contiendra alors :

- **AccessRuntime\_fr-fr\_x86.exe** (ou sa version anglaise) ;

- et bien entendu, le fichier **MSI** de l'application à installer.

## 7 - Tester son programme d'installation

Une fois que vous avez terminé le processus d'emballage et produit des supports de distribution pour votre application, vous devez tester votre programme d'installation.

Pour ce test, choisissez une machine qui ne dispose pas déjà des fichiers de votre application ni des contrôles **ActiveX** requis par celle-ci. De plus, il est conseillé de tester l'installation sur **différents systèmes d'exploitation**, notamment Windows 8 et 8.1.

Vous pouvez pour ce genre d'opérations, envisager d'utiliser des **logiciels de Machines Virtuelles** qui s'avèrent être d'un grand secours pour des simulations multiples et ce, à travers différents systèmes d'exploitation.

### 7-1 - Pour tester votre programme d'installation depuis un CD-Rom ou un volume

- Insérez le **CD-Rom** dans le lecteur approprié ;

ou bien,

- Depuis le bureau, sélectionnez le menu **Démarrer**, choisissez **Exécuter**, puis tapez **D :\setup.exe** où **D :** est le lecteur source ;

ou encore

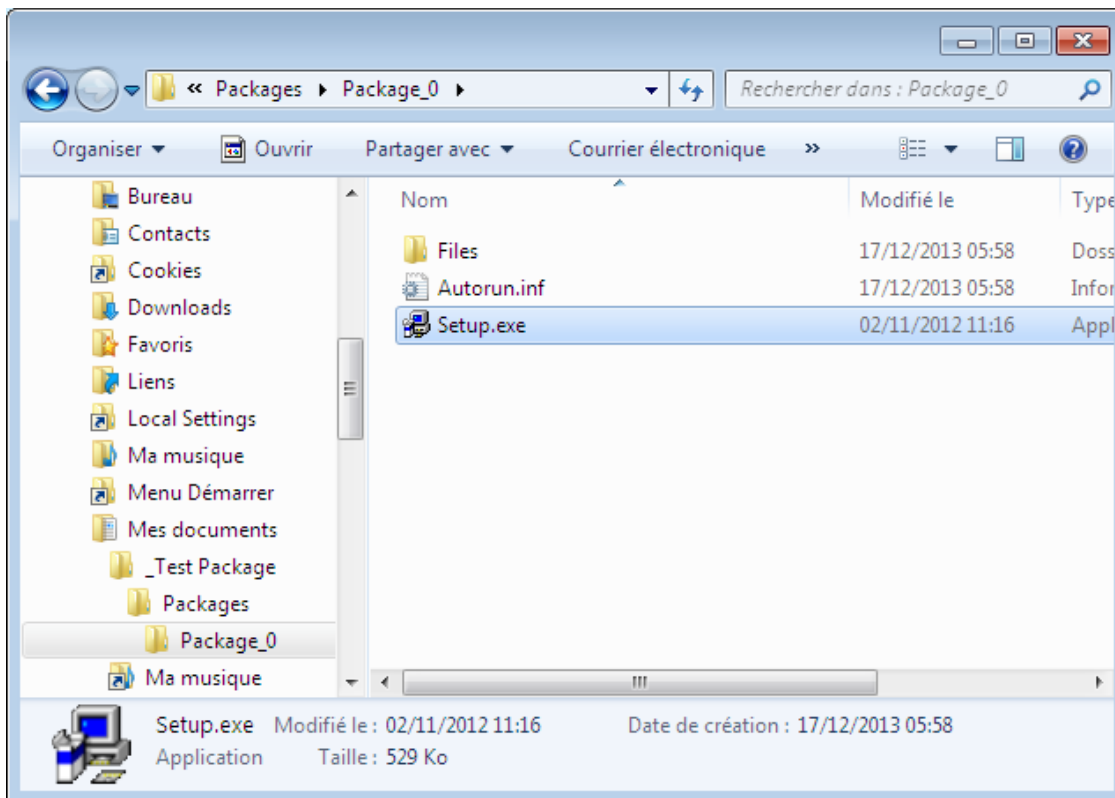
- Cliquez deux fois sur **Setup.exe** depuis l'**Explorateur de fichiers**.

Vous pouvez également ne lancer que le fichier **MSI**, le *setup.exe* étant réservé plus particulièrement aux lancements automatiques.

## 8 - Exemple d'installation sur un poste client

### 8-1 - Installation standard sans Runtime (Access 2013 présent sur le poste)

Une fois le package généré, embarquez le dossier *Package\_X* (où *X* représente le compteur) sur la machine cible. Dans notre exemple, j'ai utilisé une machine virtuelle Windows 7 SP1 avec Office 2013 Professionnel.

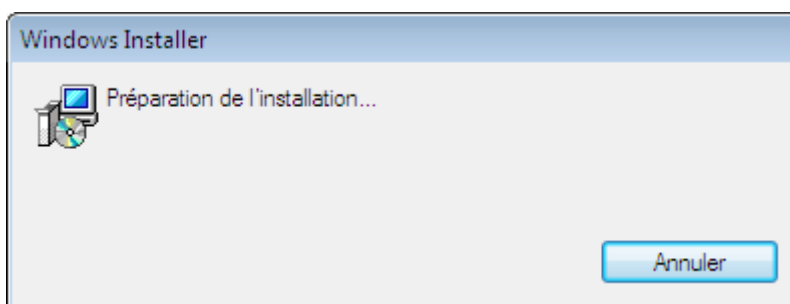


*Localisation du programme d'installation*

Ouvrez alors le fichier **Setup.exe**

La fenêtre de préparation de l'installation s'affiche alors un court instant.

Durant ce processus, l'installateur analyse votre système sous différents angles et notamment la version de Windows, l'éventuelle installation antérieure de cette application, etc.

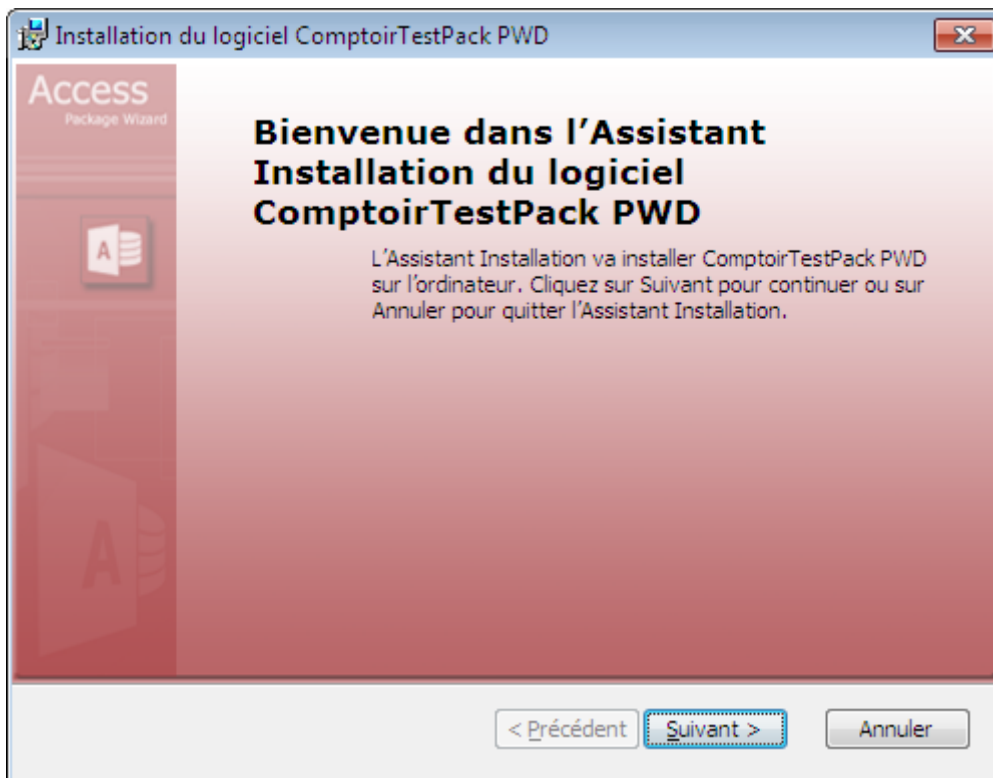


*Préparation de l'installation*

Si tout va bien, il vous est alors présenté ce qui va être installé.

Sinon, la fenêtre vous informe :

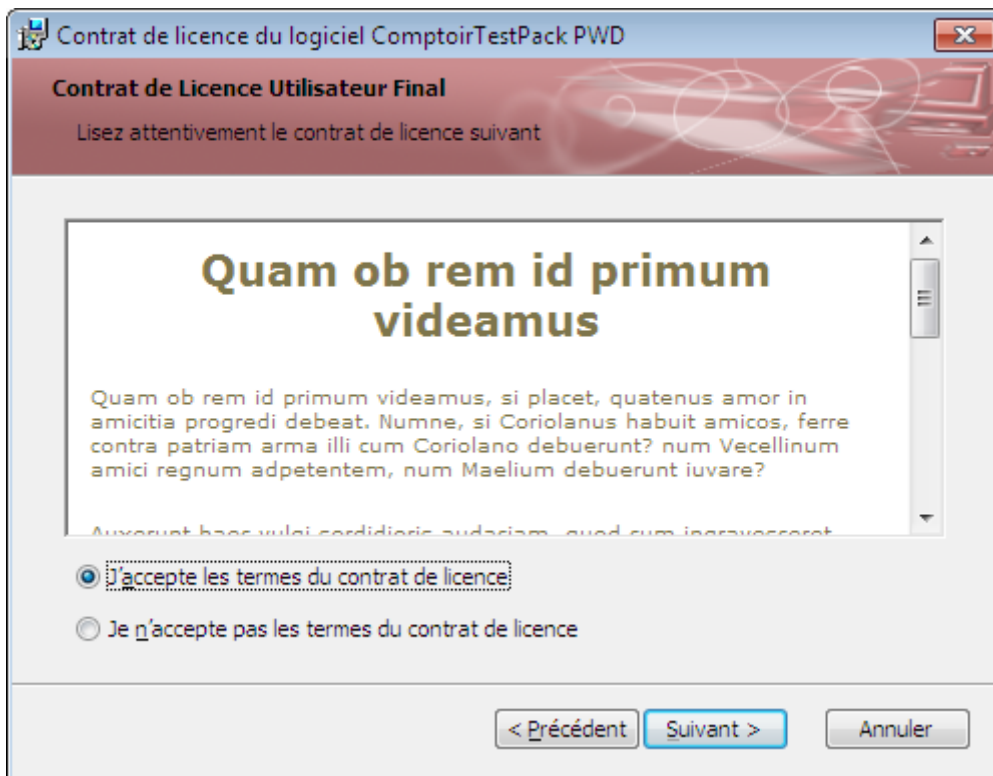
- soit de l'impossibilité de réaliser l'installation ;
- soit de **Réparer** ou **Supprimer** celle-ci (si elle a déjà été installée).



Présentation du programme installé

Cliquez sur **Suivant**.

Vous acceptez alors les conditions de la licence ; (vous remarquerez la similitude entre ce package d'installation et celui qui a permis d'installer le complément).

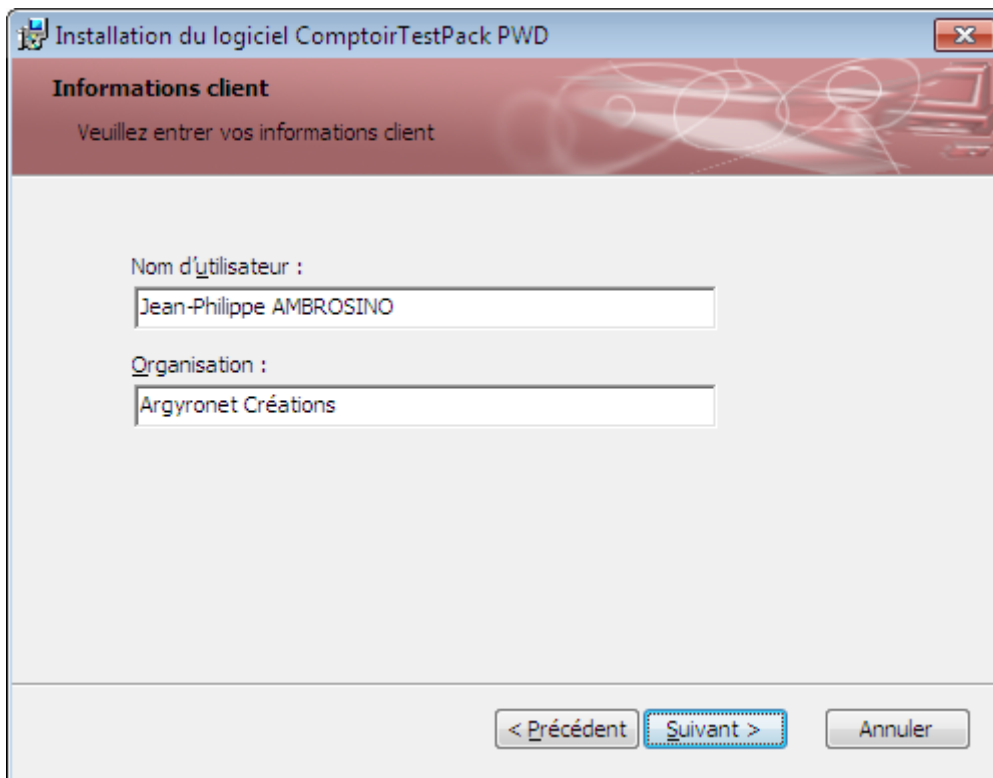


Accord de licence

Cliquez sur **Suivant**.

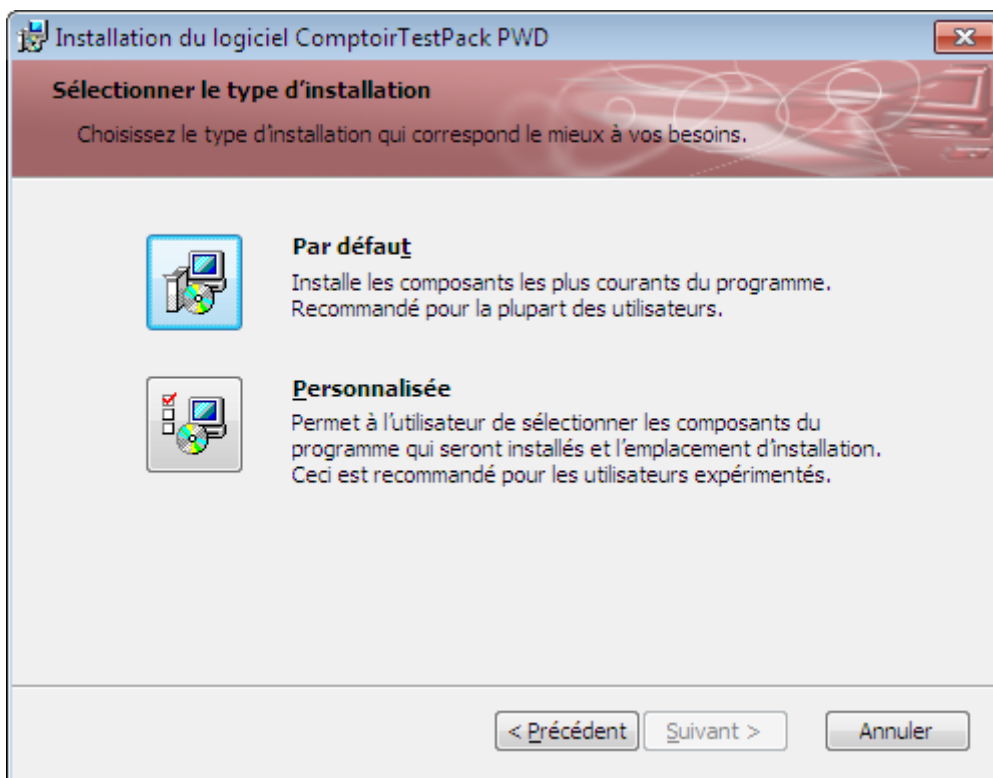


Entrez ici les noms et organisation.



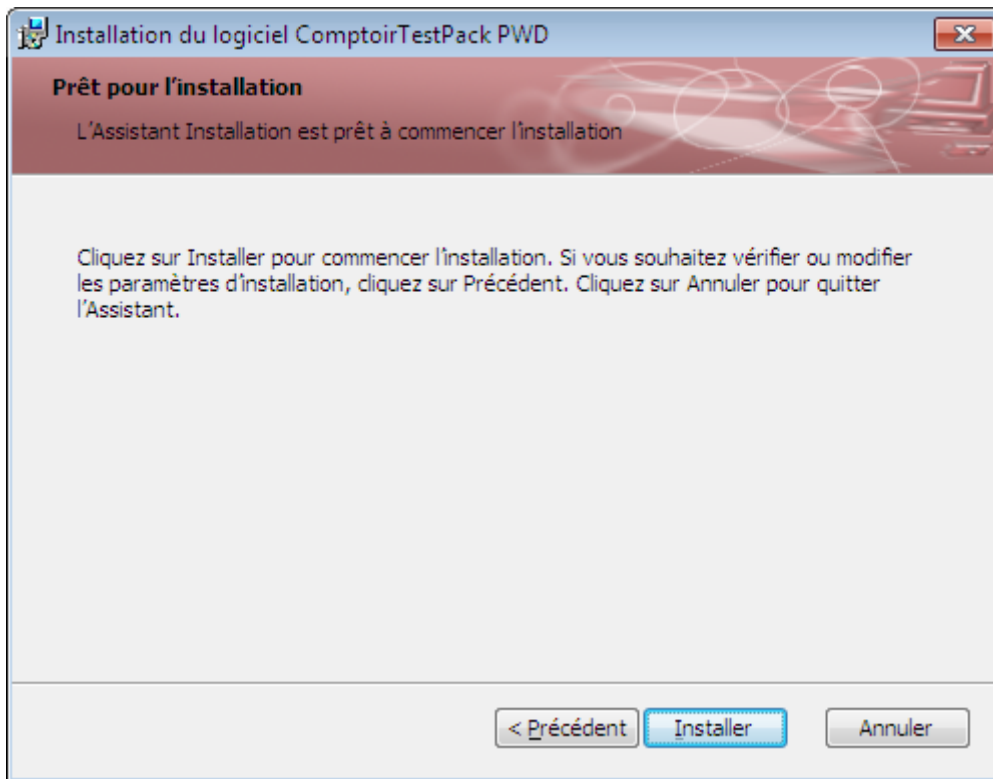
Informations sur l'utilisateur

Cliquez sur **Suivant**.  
Choisissez **Par défaut** ou **Personnaliser** selon.



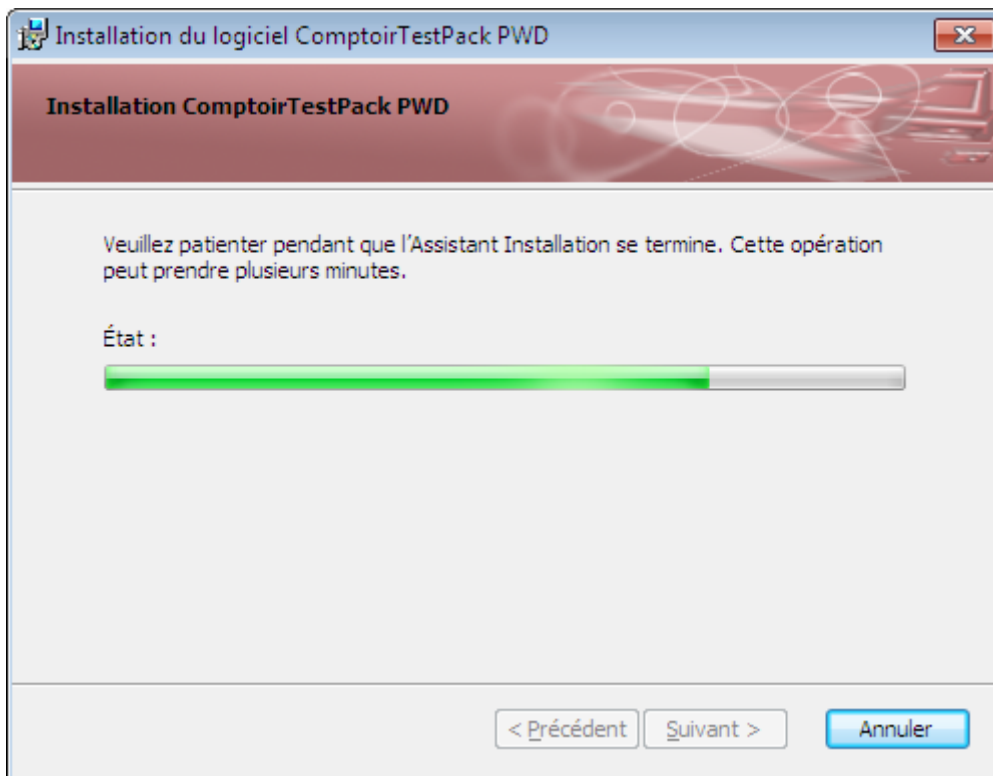
Mode d'installation

Cliquez sur **Suivant**.  
C'est ici la dernière étape avant l'installation effective.



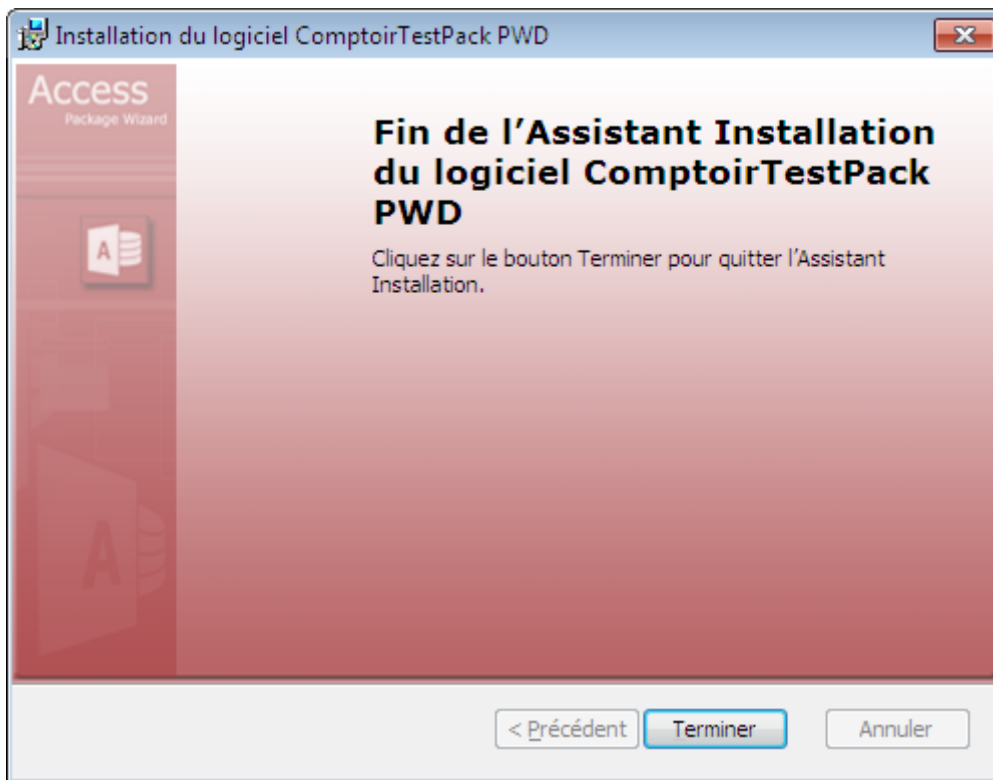
*Processus de validation de l'installation*

La progression de l'installation suit son cours.



*Installation en cours.*

Puis se termine ici après un certain temps (*tout comme met le fût du canon à refroidir.*).



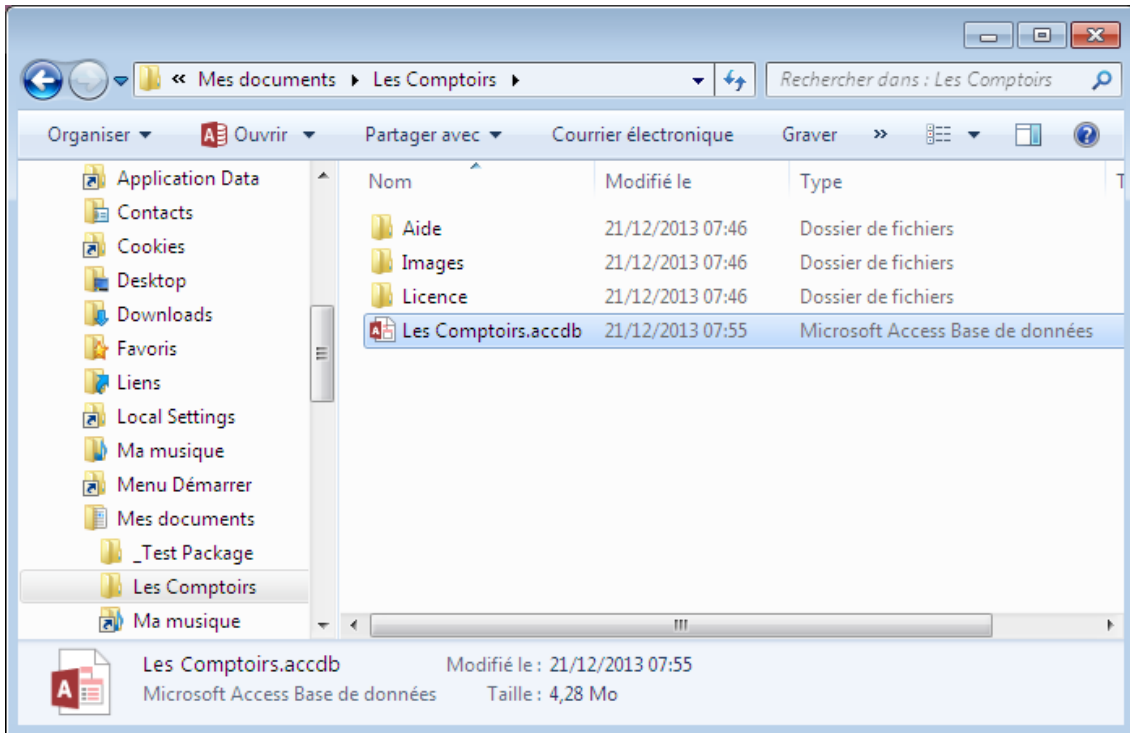
*Installation terminée*

Et l'icône est créée sur le **Bureau** si peu que vous ayez coché l'option durant l'étape 3 de l'assistant.



*Icône de l'application installée sur le Bureau*

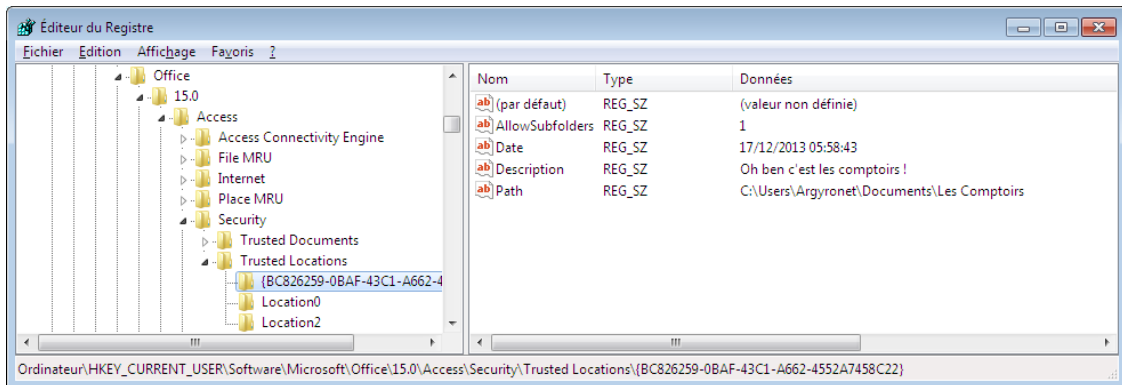
Dans l'Explorateur Windows, vous voyez le fichier de l'application Access et les sous-dossiers que vous avez déclarés durant l'étape 4.



*Emplacement physique de l'application*

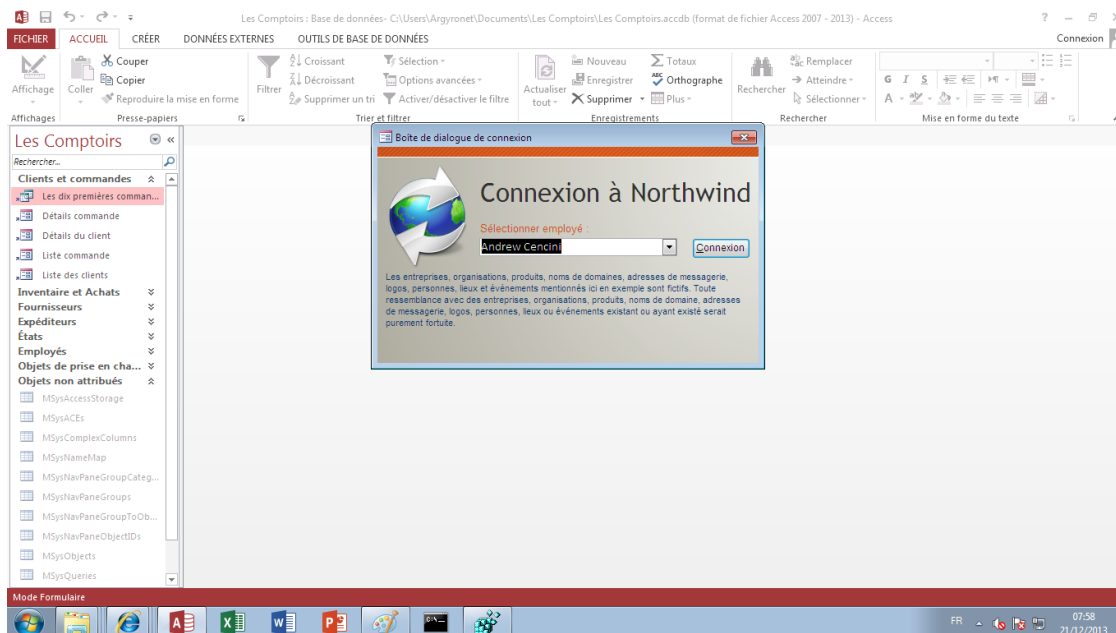
**Base de Registre**

Ainsi que prévu, si vous avez coché le fait que vous souhaitiez un **Emplacement approuvé**, il est créé automatiquement dans le Registre.



*Emplacement approuvé de l'application*

Dans ces conditions, l'application empaquetée se charge **sans alerte d'avertissement bloquant**.



Lancement de l'application

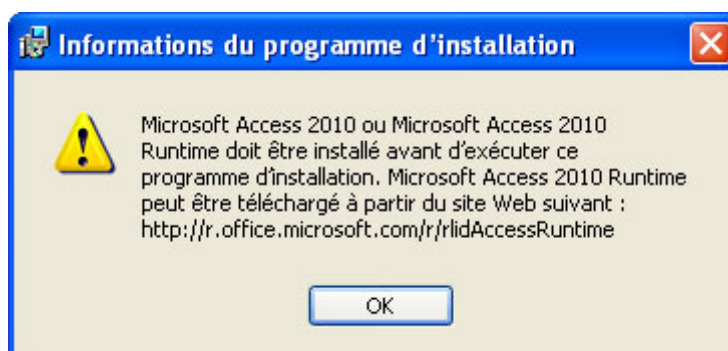
## 8-2 - Installation avec Runtime (Access 2013 absent du poste cible)

Dans ce programme, j'ai **intentionnellement supprimé** la vérification de la présence du programme Access ou de son Runtime avant que ne soit effectué l'installation effective de l'application.

*En effet, cela était **contradictoire** (notamment dans la version 2010 de l'Assistant de Solution de package) dans le sens où vous devez installer le Programme **Access** ou son **Runtime** avant d'installer l'application.*

*Or, le processus d'installation du Runtime s'effectue par **installation chaînée**, il est donc installé après que l'application ait été elle-même installée ce qui est tout de même étrange.*

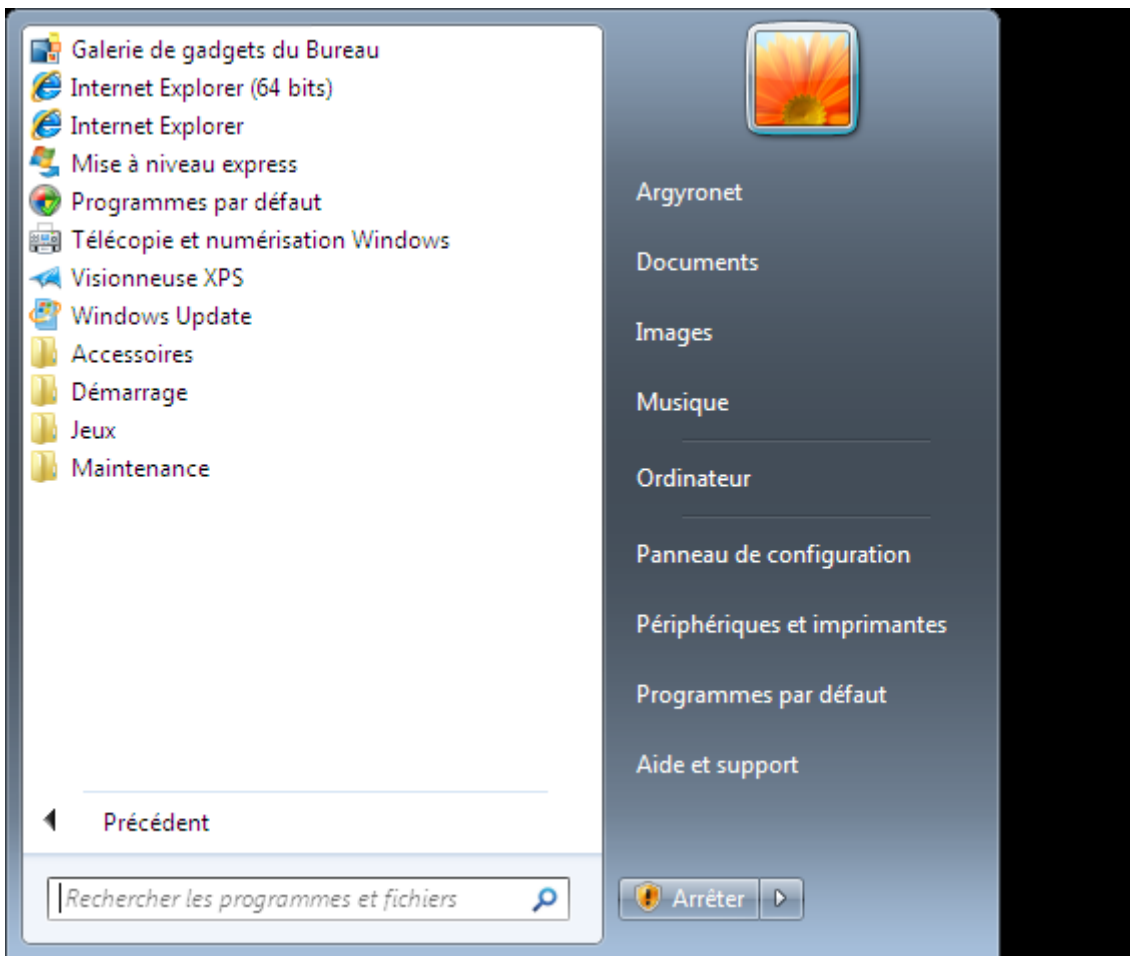
Du coup, ici, vous pourrez installer l'application mais elle ne marchera que s'il y a quelque chose pour la faire tourner, et donc **MSACCESS.EXE**.



Message d'erreur du MSI d'un package 2010

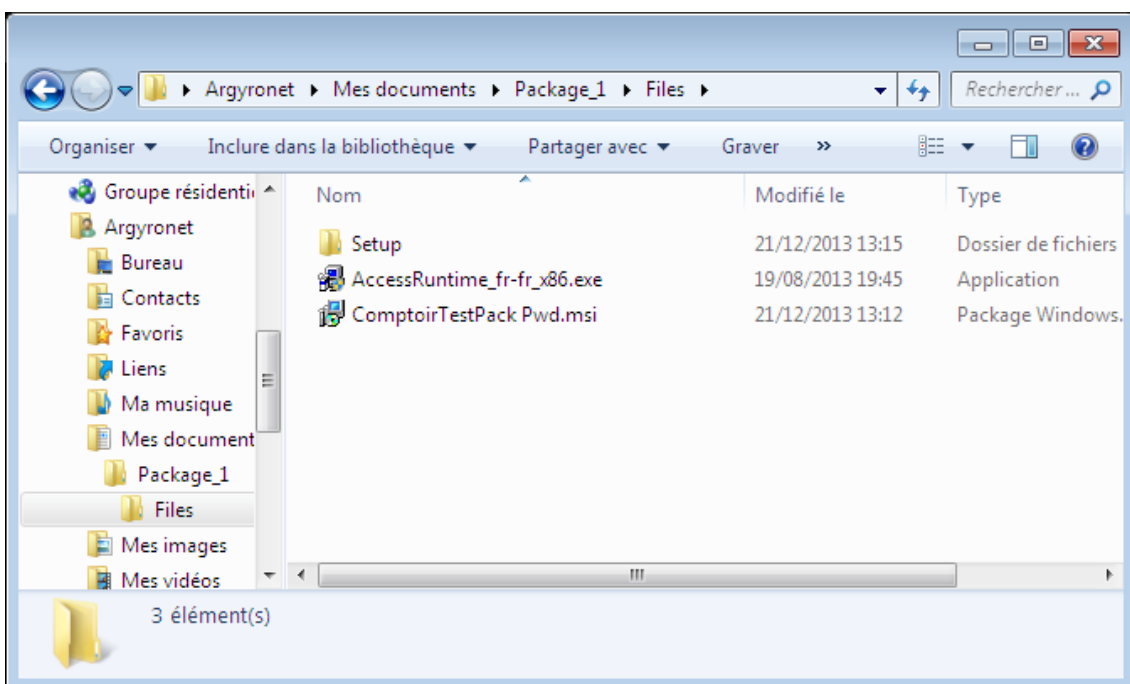
Une fois le package généré, embarquez le dossier Package\_X (où X représente le compteur) sur la machine cible. Dans cet exemple, j'ai utilisé une autre machine virtuelle Windows 7 SP1 sans Office 2013 ni le Runtime.





*Menu Demarrer de Windows 7*

Localisez le fichier d'installation **Setup.exe** comme vu à la section précédente.



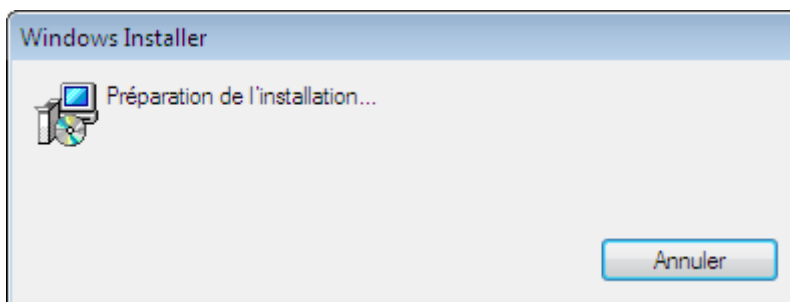
*Localisation du programme d'installation*



Ouvrez alors le fichier **Setup.exe**.

La fenêtre de préparation de l'installation s'affiche alors un court instant.

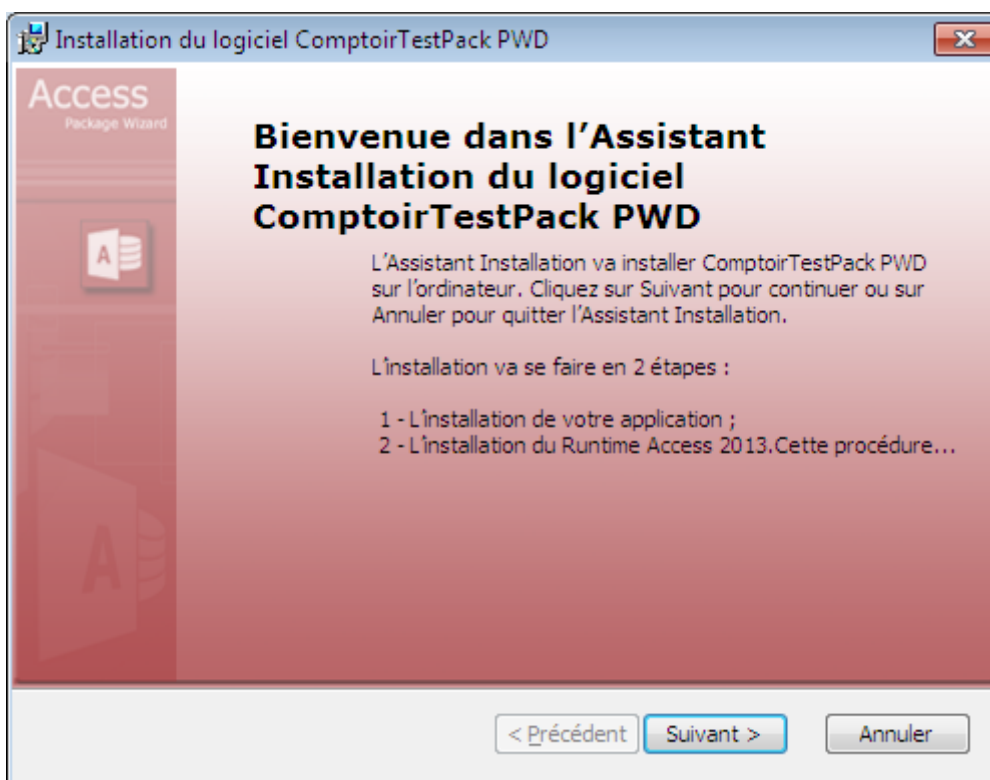
Durant ce processus, l'installeur analyse votre système sous différents angles et notamment la version de Windows, l'éventuelle installation antérieure de cette application, etc.



*Préparation de l'installation*

Si tout va bien, il vous est alors présenté ce qui va être installé.

- i** Remarquez ici que la fenêtre vous présente les deux étapes d'installation :
- l'application d'une part.
  - le Runtime d'autre part.



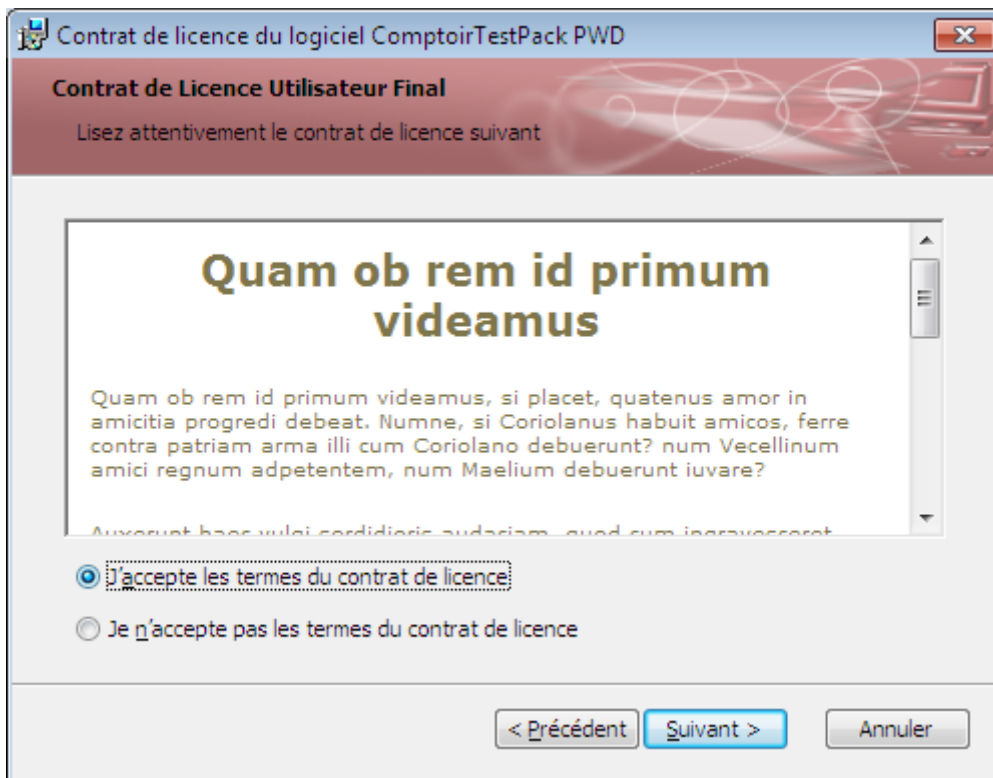
*Présentation du programme installé*

; Sinon, la fenêtre vous informe :

- soit de l'impossibilité de réaliser l'installation
- soit de **Réparer** ou **Supprimer** celle-ci (si elle a déjà été installée).

Cliquez sur **Suivant**.

Vous acceptez alors les conditions de la licence ; (vous remarquerez la similitude entre ce package d'installation et celui qui a permis d'installer le complément).



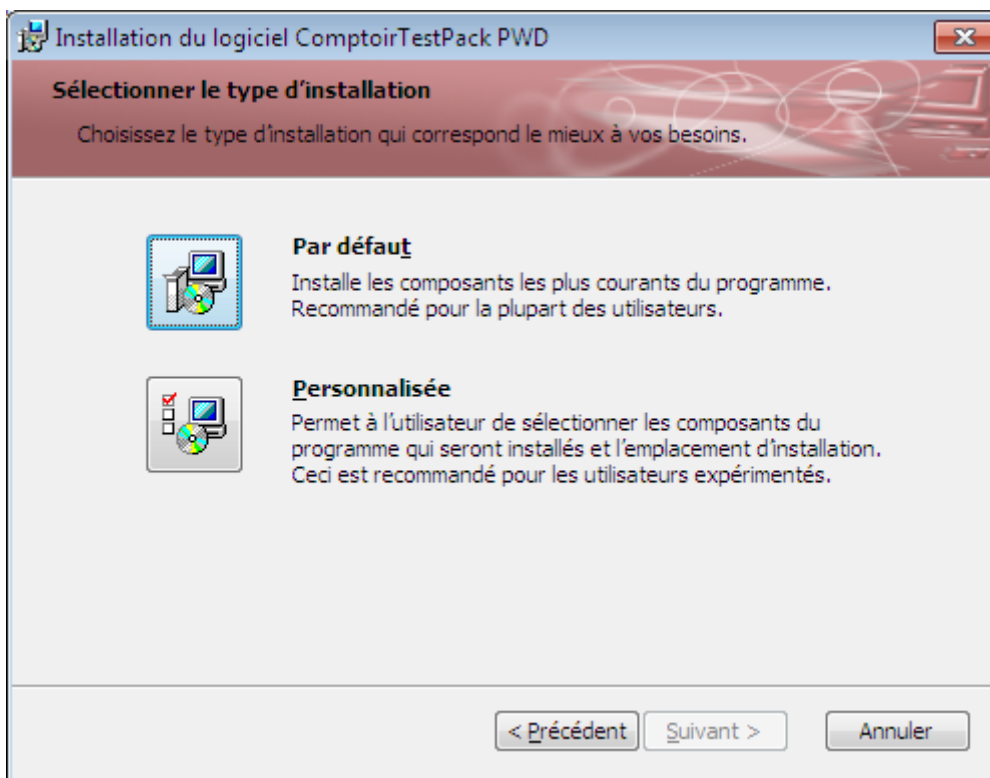
*Accord de licence*

Cliquez sur **Suivant**.  
Entrez ici les noms et organisation.



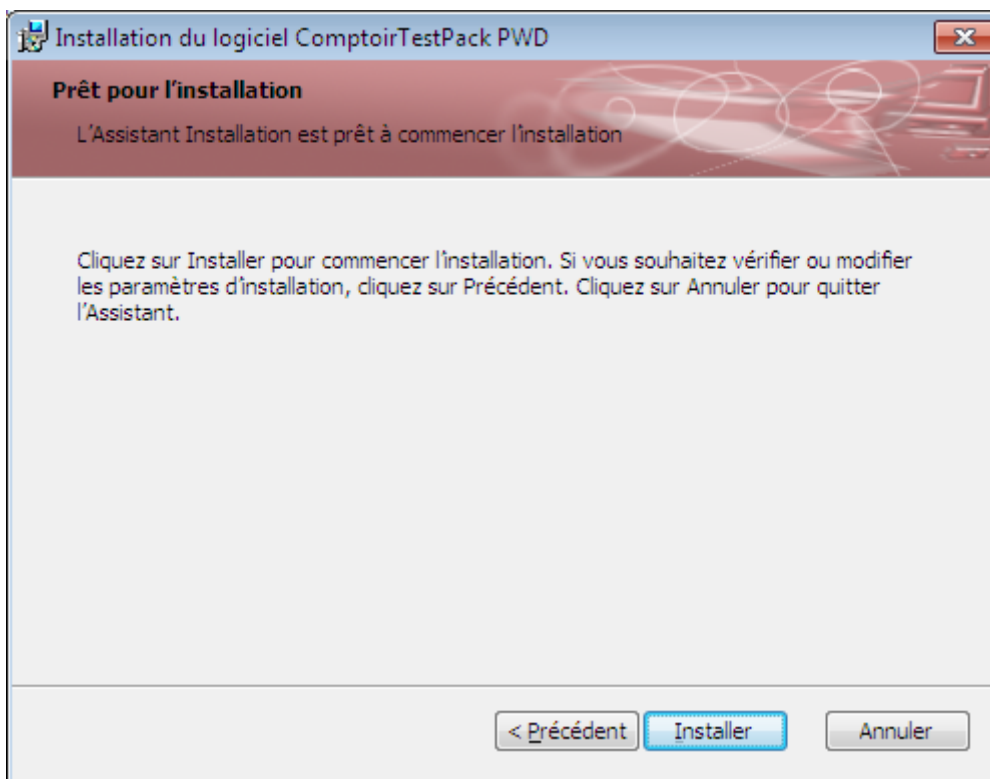
*Informations sur l'utilisateur*

Cliquez sur **Suivant**  
Choisissez **Par défaut** ou **Personnaliser** selon.



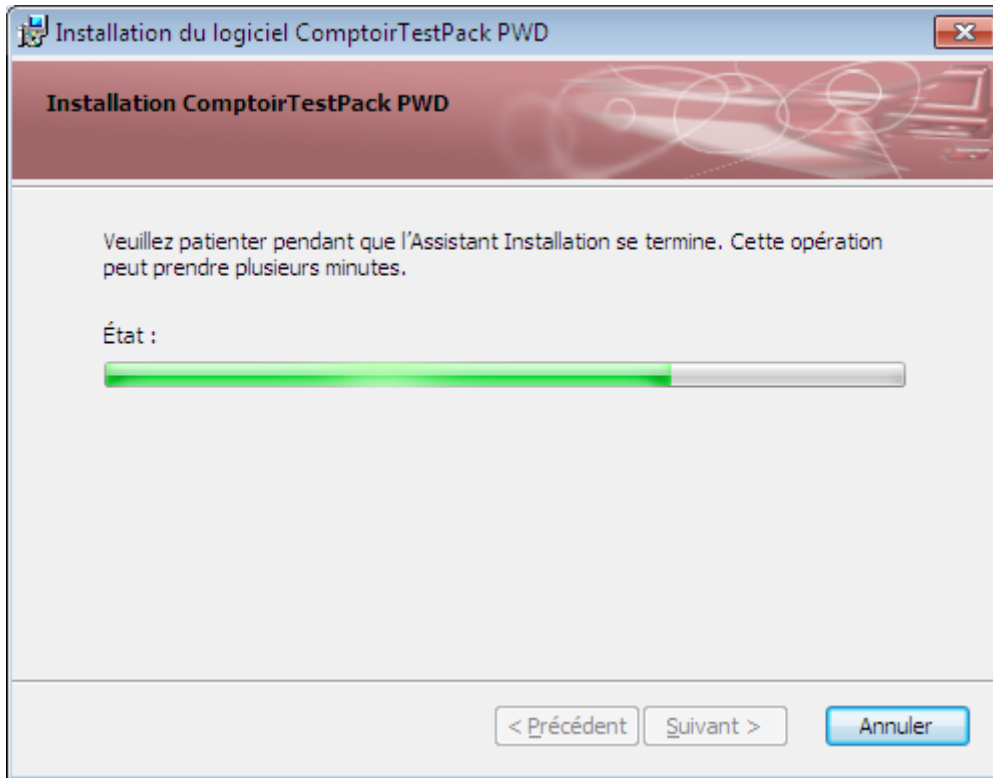
*Mode d'installation*

Cliquez sur **Suivant**.  
C'est ici la dernière étape avant l'installation effective.



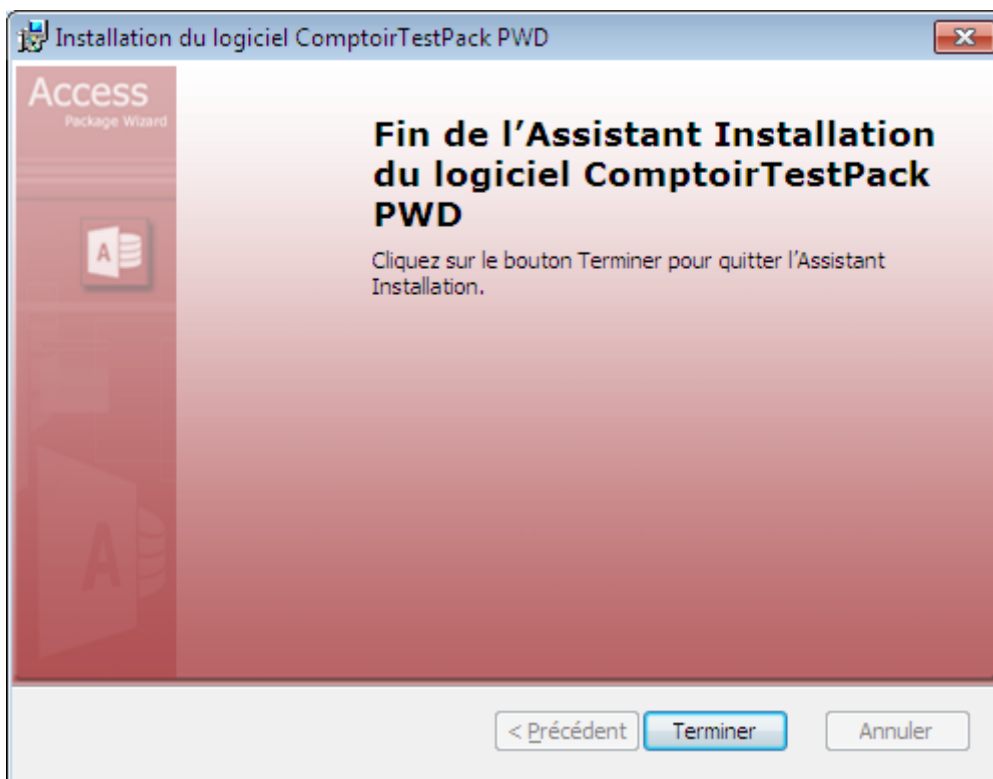
*Processus de validation de l'installation*

La progression de l'installation suit son cours.



*Installation en cours.*

Puis se termine ici après un certain temps (*tout comme met le fût du canon à refroidir.*).



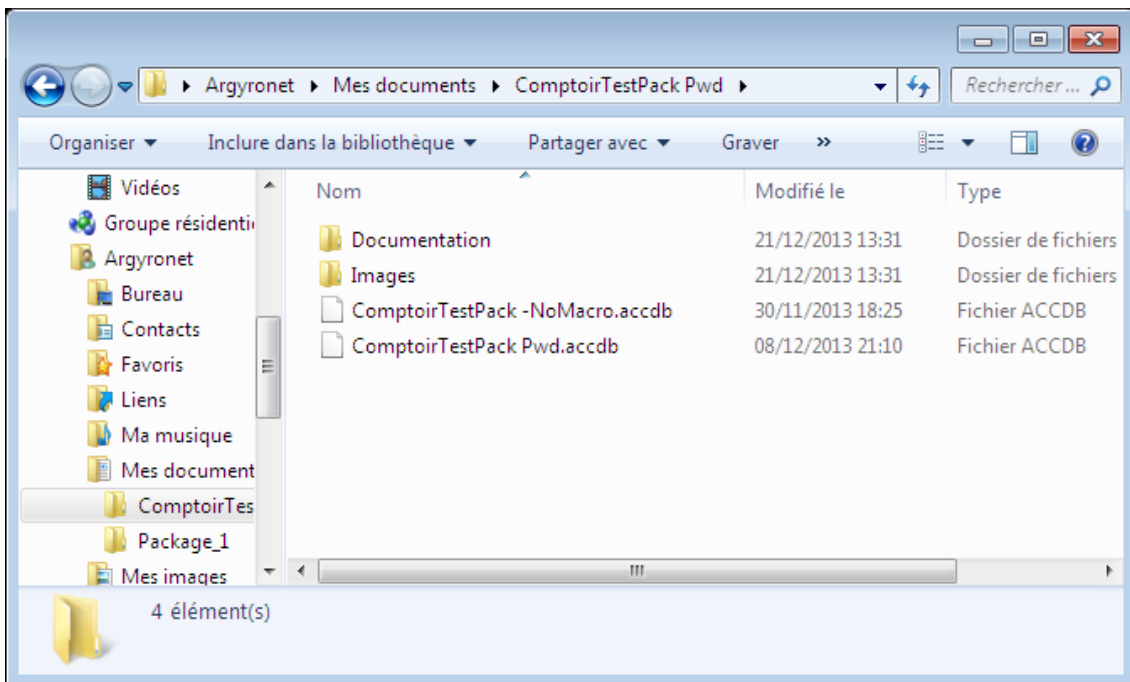
*Installation terminée*

Et l'icône est donc créée sur le **Bureau** si peu que vous ayez coché l'option durant l'étape 3 de l'assistant.



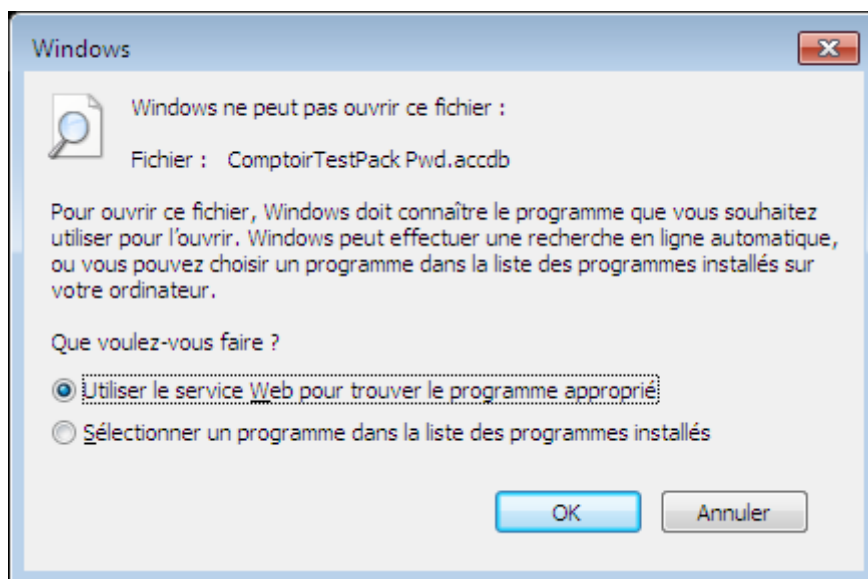
*Icône de l'application installée sur le Bureau*

L'installation étant terminée, ainsi que vous le constatez dans l'illustration ci-dessous, l'application est bien installée mais on voit bien que l'image de son icône reste blanche du fait qu'elle n'est associée à aucun programme connu. Normal, puisque le Runtime n'est pas encore installé.



*Emplacement physique de l'application*

D'ailleurs, si à ce stade vous chercher à ouvrir l'application, un message vous informe qu'il ne trouve aucun programme pour le faire.

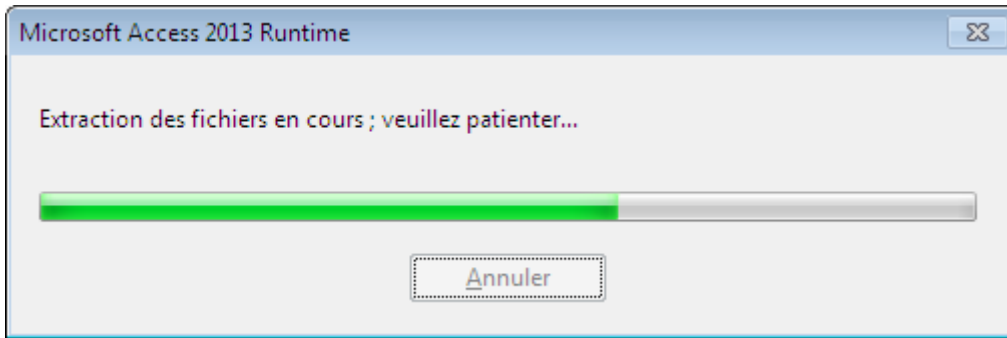


*Emplacement physique de l'application*

### ***Installation du Runtime***

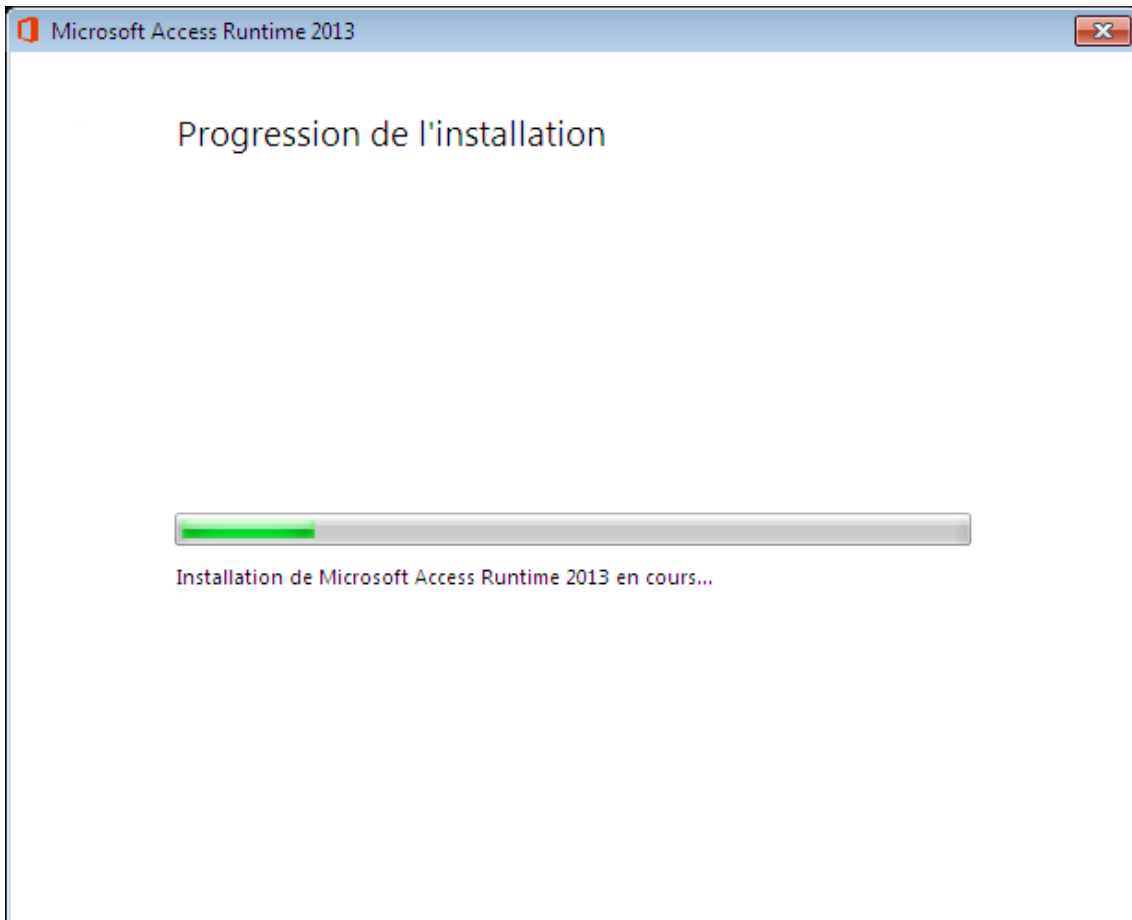
C'est alors que démarre l'installation du Runtime.





*Emplacement physique de l'application*

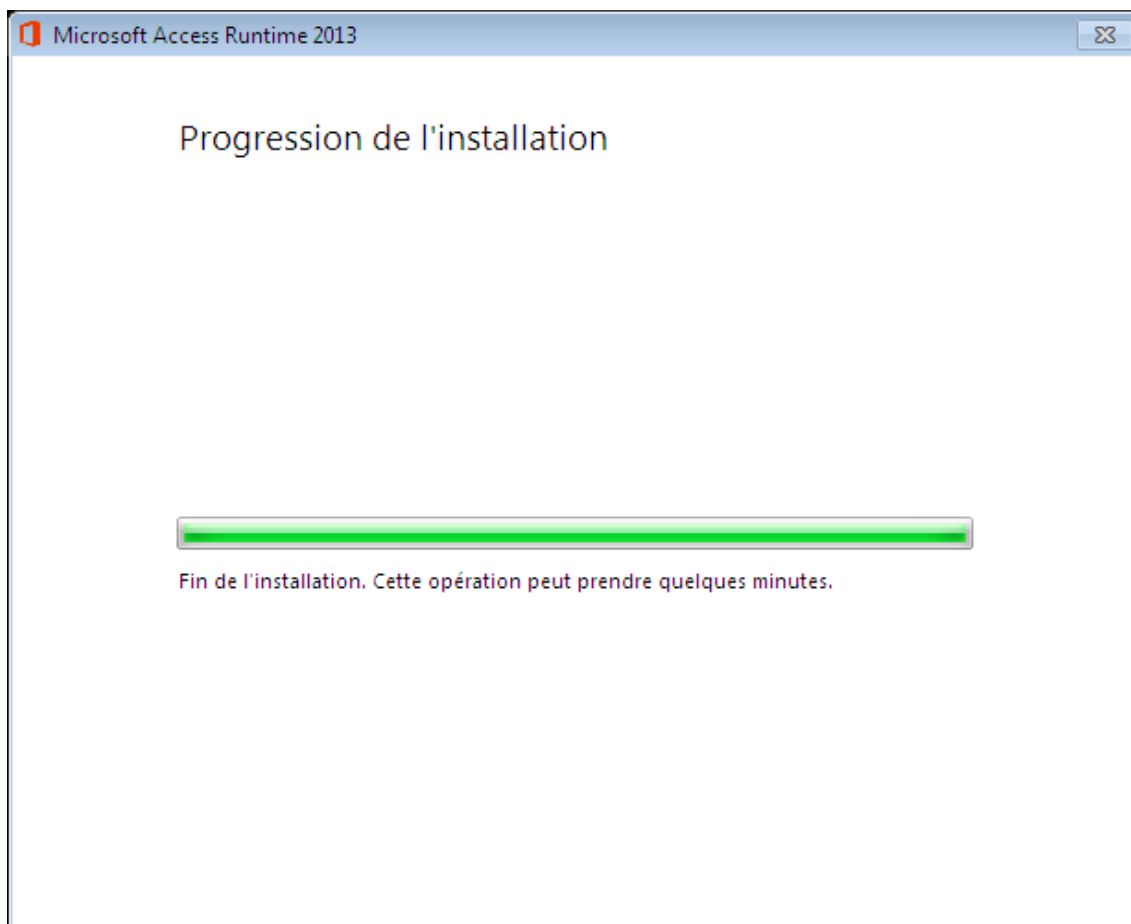
L'installation est assez longue à décoller.



*Emplacement physique de l'application*

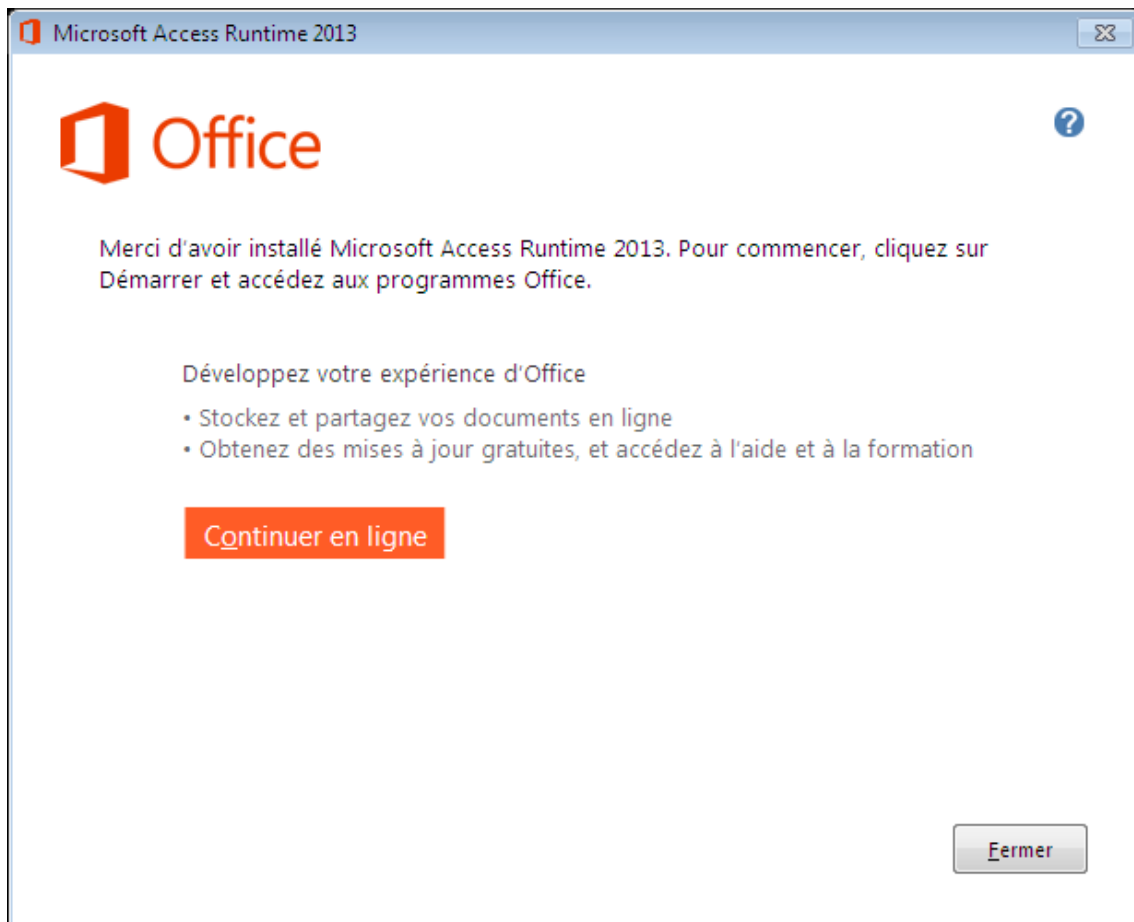
Et se termine au bout de quelques minutes.





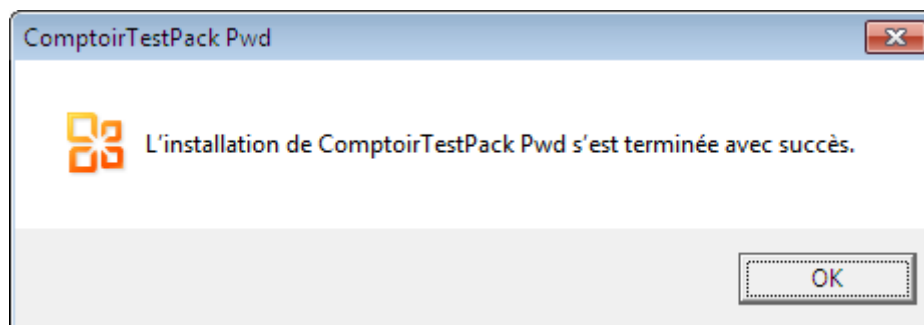
*Emplacement physique de l'application*

Le Runtime est alors installé.



*Emplacement physique de l'application*

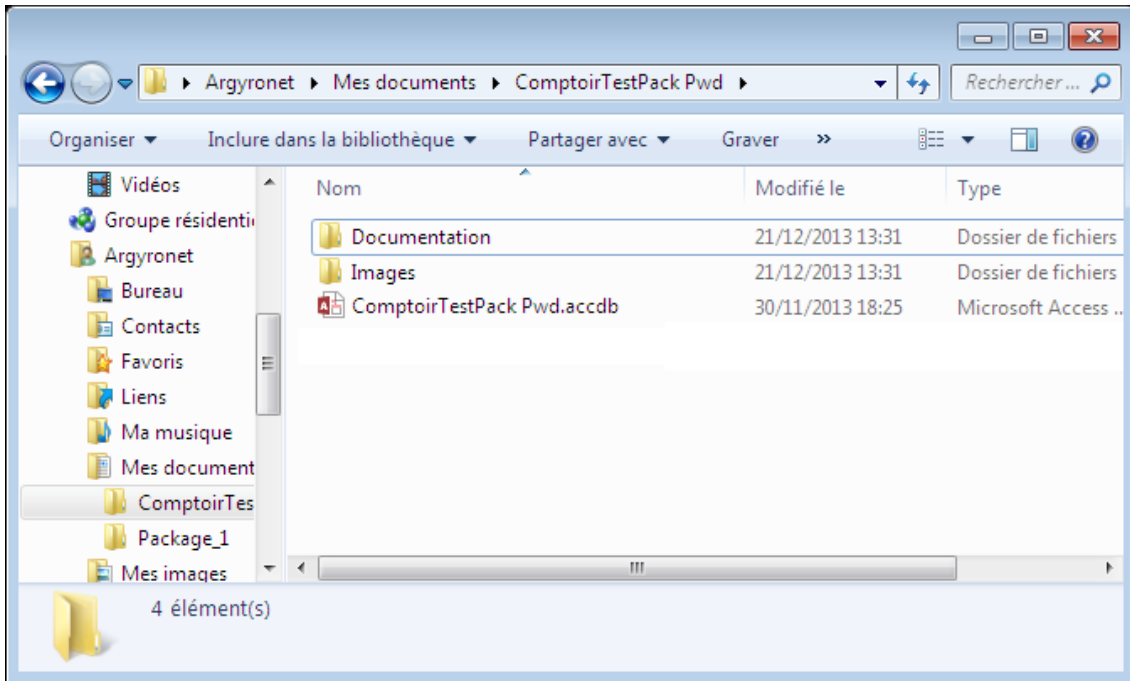
Et l'application aussi ; dans son entier.



*Emplacement physique de l'application*

Dans l'Explorateur Windows, vous voyez le fichier de l'application Access et les sous-dossiers que vous avez déclarés durant l'étape 4.

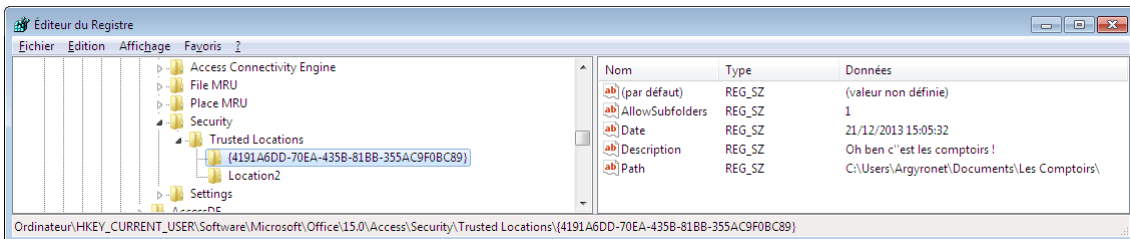
Remarquez que cette fois, l'icône (et surtout l'application) est bien affectée à MSACCESS.EXE (Runtime).



*Emplacement physique de l'application*

### **Base de Registre**

De façon identique, si vous avez coché le fait que vous souhaitez un **Emplacement approuvé**, il est créé automatiquement dans le Registre



*Emplacement approuvé de l'application*

## 9 - Les tests à effectuer sur un poste de test

Une fois l'installation terminée, exécutez le programme installé pour vous assurer de son bon fonctionnement. En général, vous n'êtes pas à l'abri de surprises inattendues et c'est le meilleur moyen de s'en rendre compte avant de délivrer le package à votre client ou à l'équipe d'utilisateurs de votre entreprise. Veuillez alors revenir à la **Section 2** relatant des tests préalables.

### 9-1 - Test de l'application installée sur un poste.

Le jeu de test consiste à exécuter votre application depuis le poste sur lequel vous l'avez installé.

#### 9-1-1 - Poste doté de la version complète de Microsoft Access

Il ne devrait pas y avoir de grande différence avec le poste de développement depuis lequel vous avez créé l'application.

Le jeu de test consistera alors à naviguer entre les différents formulaires, l'impression des états (*attention notamment à l'orientation des états et aussi à la notion d'imprimante par défaut*), à l'exécution de l'ensemble des tâches proposées par votre application.

#### 9-1-2 - Poste doté du Runtime Access seul sans aucune version de Microsoft Access

Si le poste client ne dispose pas d'une version complète de Microsoft Access, le **Runtime est alors résidant** sur le poste client et se comporte comme si une version d'Access était installée sauf que celle-ci est dépourvue de ses fonctionnalités de conception.

Dans ce cas, il n'y a pas plus de procédures de test à effectuer que celles qui sont stipulées en section 9.1.1 ci-avant quoique, certaines commandes étant désactivées, il est conseillé de faire utiliser votre application par un tiers qui ne la connaît pas.

#### 9-1-3 - Poste doté du Runtime Access conjointement avec une version différente de Microsoft Access

Bien que ce soit déconseillé, on considère que le poste client dispose d'une version de Microsoft Access **antérieure** à celle du **Runtime 2013**.

Dans ce cas, vous devez mettre en place une syntaxe particulière, car l'Assistant Package 2013 ne génère pas de raccourci approprié pour ce cas de figure.

Pour ce faire, soit vous le créez manuellement, accompagné des **options** de ligne de commande nécessaires pour démarrer Microsoft Access Runtime 2013 ou bien de modifier les associations de fichiers comme cela a été spécifié auparavant.

### Cas particulier pour un environnement réseau

*Je vous **déconseille** en cas d'utilisation en réseau de mettre votre application frontale sur un serveur.*

*Outre le fait que dans ce cas, plusieurs personnes peuvent partager les mêmes formulaires et autres objets, les temps de chargement et d'exécution seraient horribles !*

*Vous devez plutôt déplacer la base de données **dorsale** ne contenant que les tables sur le serveur dédié car le setup ne saura pas le faire nativement. L'application frontale qui est liée à cette base sur chaque poste utilisateur sera installée dans le dossier que vous avez choisi durant la génération du Package.*

*Au début du document, je vous ai recommandé de travailler dans le contexte réel lors du développement de votre application, à savoir développer*



*vos projets sur votre poste tout en ayant déposé la base de données (tables) dans un dossier dédié sur **le serveur** s'il s'agit d'une application*

multi-utilisateur ou dans le dossier **.\Nom de l'Appli\Data** s'il s'agit d'une application mono-utilisateur.

Toutefois, pour un confort de développement (**dû notamment aux lenteurs du réseau**), il peut être préférable de travailler totalement en local et envisager de scinder la base de données plus tard lorsque le projet tire à sa fin dans sa phase de développement.

En cours de développement pour un cas de figure multiutilisateur, cela permet de se rendre compte notamment des temps de réponse de l'ouverture de certains formulaires exécutant des requêtes lourdes, et vous invite par le fait d'envisager de les optimiser ou de comprendre pourquoi.

## 9-1-4 - .doté d'une version 2010 de Microsoft Office

J'aborde ici un sujet un peu plus délicat dans le sens où la version antérieure à Office 2013 est sortie il y a près de 3 ans.

Du fait que les fichiers de type **accdb**\* sont du même format lorsqu'il s'agit d'Office 2010 ou 2013, je serais en droit de me dire qu'il ne devrait pas y avoir de *problème de compatibilité*.

Oui, c'est exact en théorie ; il n'y a pas de problème parce que si vous ouvrez une base Access créée sous **Access 2010**, elle s'ouvrira sans problème dans **Access 2013** sans qu'il y ait d'avertissement d'antériorité de version.

Toutefois, Microsoft a préféré considérer le fait que la compatibilité **2013 vers 2010** n'était peut-être pas parfaite et souhaitant émettre des réserves sur ce point, il est vrai qu'il est préférable de s'assurer du bon fonctionnement de son projet et ce particulièrement avec le Runtime qui pourrait poser des problèmes de comportement.

Donc si vous distribuez un package déployé avec Access 2013 à un utilisateur dont le poste est pourvu de la version complète de **Microsoft Access 2010** ou de son **Runtime**, vous serez confronté à la problématique de fournir impérativement le Runtime 2013 mais aussi de faire en sorte de désinstaller toute version antérieure sans quoi, il sera sans doute impossible à l'utilisateur d'utiliser votre application.

### Résumé des situations :

- Si le poste est pourvu d'Access 2013 ou du Runtime 2013 => Package 2013 : Installation sans problème
- Si le poste est pourvu du Runtime Access inférieur à 2013 => Package 2013 : Installation à risque \*
- Si le poste est pourvu d'Access inférieure à 2013 => Package 2013 : Installation à risque \*

\* Il vous faut fournir le Runtime 2013 avec votre package

## 9-2 - Erreur d'exécution au cours des tests

Vous n'êtes pas à l'abri de surprises en matière de bugs et d'erreurs concernant votre application nouvellement installée sur le poste de test.

Si vous n'en rencontrez pas, tant mieux.

Les plus courantes sont en général liées à une omission ou une non-considération d'un élément de la part du développeur vis-à-vis de son application et/ou de son environnement.

### 9-2-1 - Quelques exemples :

- Vous pouvez très bien voir une erreur qui se traduit par le dysfonctionnement d'un ruban personnalisé. Vous devez alors vérifier, selon la façon dont il puise sa structure (*cas d'un fichier XML*) qu'il est présent et opérationnel.

- Une autre sera liée à l'exploitation d'un composant qui n'a pas été enregistré sur le poste de test. Ce cas de figure peut se rencontrer quand vous avez greffé un nouvel ActiveX® dans votre projet, que vous avez créé le fichier ACCDE (ou l'**ACCDB**) et fourni ou copié vous-même l'ACCDE (ou le **ACCDB**) sur le poste de test sans avoir pris la précaution d'embarquer le composant et de l'enregistrer dans la Base de Registre de Windows du poste concerné. (*Erreur 429*).

- Il est également possible que vous rencontriez une erreur qui est fort peu commune certes mais il m'est arrivé de la rencontrer de temps en temps.

Cette erreur se traduit par le message stipulant qu'une **bibliothèque n'est pas enregistrée** mais pour autant ne crée pas d'erreur numéro 429.

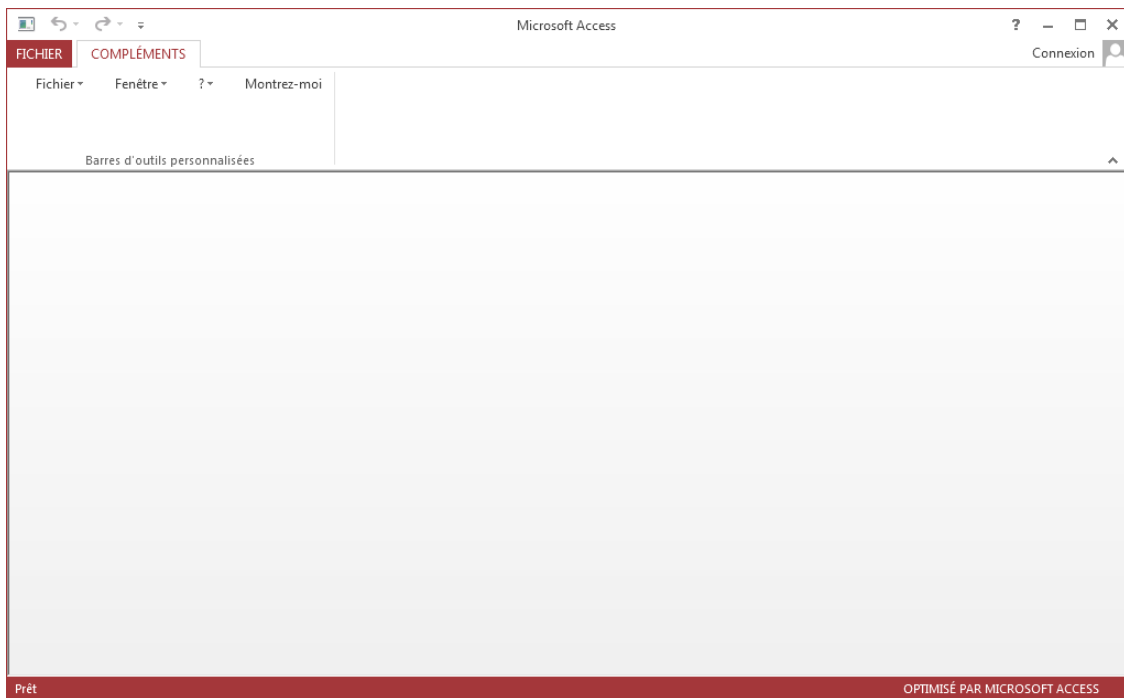
J'ai rencontré cette situation plusieurs fois avec la référence à Microsoft Scripting Runtime. La seule solution qui vous reste alors consiste à inscrire le composant par le biais de la commande permettant d'enregistrer ce type de bibliothèques dans la base de Registre.

*Pour de plus amples informations concernant la configuration requise pour les applications **Microsoft Office 2013** et plus particulièrement Access qui est très gourmand en mémoire, je vous invite à vous rendre **sur cette page** afin de vérifier que les postes utilisateurs répondent bien aux exigences de cette configuration.*

### 9-3 - Création d'un formulaire de démarrage

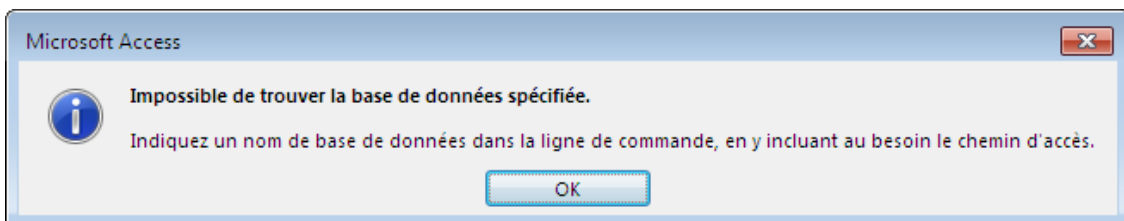
Il existe deux cas de figure qui concernent l'obligation de créer un formulaire de démarrage :

- si aucun **formulaire de démarrage** n'est stipulé et que votre application est un fichier **ACCDE** ou un **ACCDR** rien ne se passera sinon l'ouverture d'une fenêtre vide, absente de toutes possibilités sinon celles de fermer la base de données ou de quitter Microsoft Access ;
- si une macro **AUTOEXEC** est présente, elle se doit de spécifier l'ouverture d'un formulaire sinon, même phénomène que ci-avant se produira.



*Session Access inutilisable*

Dans le même esprit, vous ne pouvez pas lancer MSACCESS.EXE en tant que programme lorsqu'il s'agit du Runtime.



*Base de données obligatoire*

Il est donc conseillé de **créer** ce formulaire et parallèlement à cela, de **stipuler** que celui-ci **doit être chargé au démarrage**, qu'il soit **Menu principal**, **Fenêtre de connexion** ou **Splash screen**. Créer un formulaire de démarrage spécifique sous forme de « *Splash Screen* » permet d'obtenir un formulaire qui s'affichera de façon temporisée comme dans toutes les applications Microsoft au démarrage (§ *Word, Excel*).

Ce dernier pourra par exemple se charger de vérifier un certain nombre de paramètres comme le chargement des **Rubans** avant d'ouvrir le **Menu Principal**.

*Ceci dit, vous n'êtes pas tenu d'opter pour le choix des options qui définissent le formulaire de démarrage ; La mise en place d'une*

**i** **Macro AutoExec** peut très bien faire l'affaire mais dans tous les cas, un formulaire doit être affiché si votre projet tourne avec le Runtime.

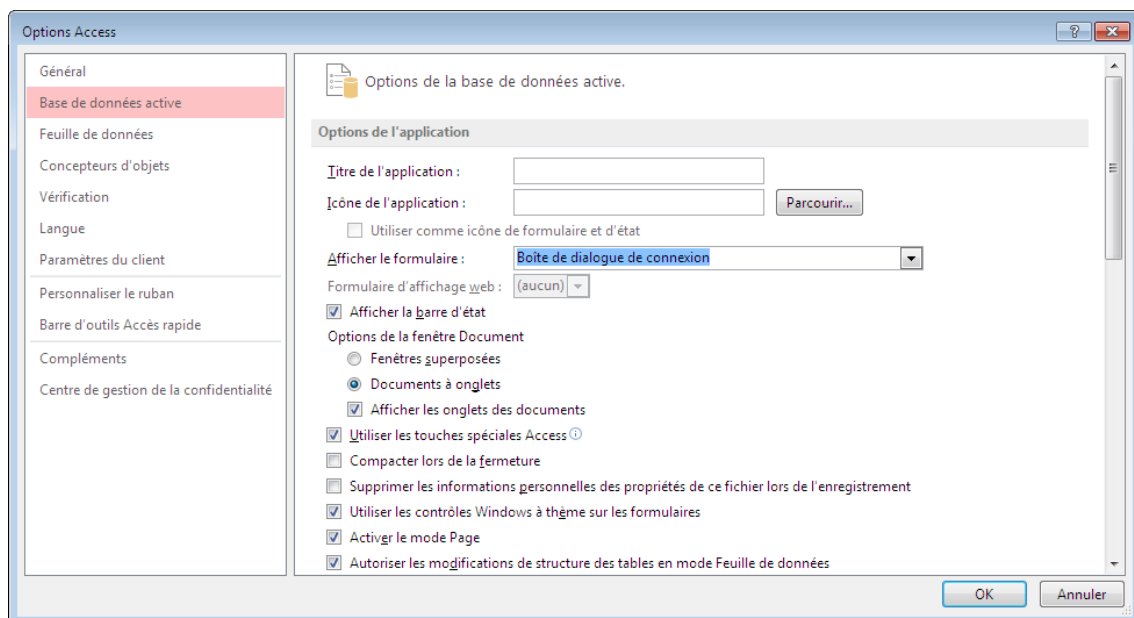
*Tout dépend des cas et des besoins.*

### 9-3-1 - Spécification d'un formulaire de démarrage et des options de démarrage

La boîte de dialogue **Option Access** vous permet de définir le formulaire à ouvrir au lancement de l'application que vous avez développée.

Pour y accéder, ouvrez les options d'Access depuis l'onglet **Fichier** et cliquez sur la rubrique **Base de données active**.

Spécifiez les options comme dans l'exemple, ci-dessous où vous sélectionnez le nom du formulaire à charger (ici **Boîte de dialogue connexion**).



*Options de démarrage*

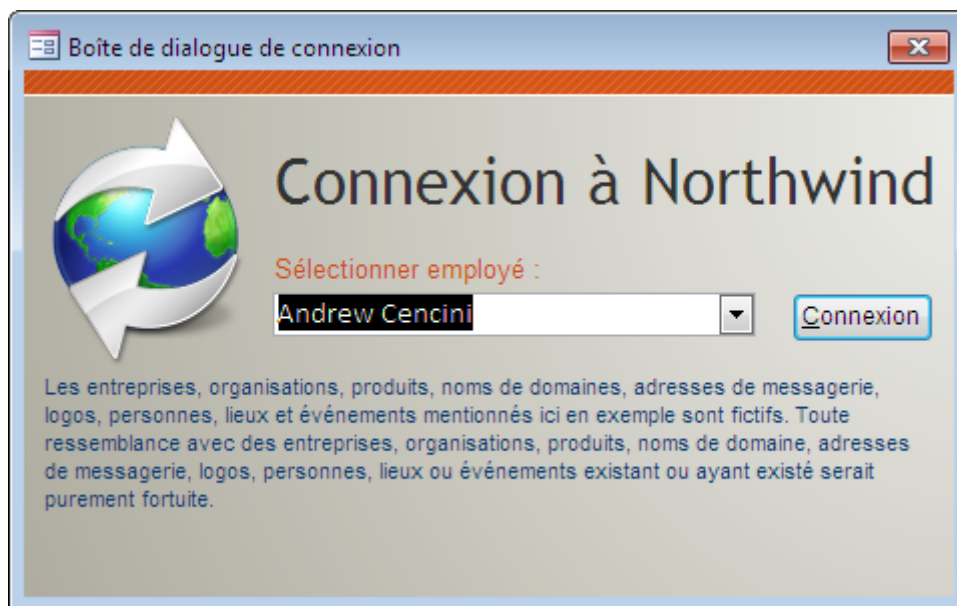
Pour une sécurité totale, vous pouvez décocher toutes ou partie des cases **mais cela reste risqué si vous n'avez pas prévu le code pour les rétablir** ou bien pour accéder à votre application en **mode Création**, par exemple avec un bouton caché ou une combinaison de touches particulière.

Sinon, nombreuses pourraient être les personnes qui posteraient sur le forum en écrivant qu'elles ont suivi ces conseils et que depuis, elles ne peuvent plus rien faire avec leur application.

### 9-3-2 - Exemple de formulaire de démarrage

Ce formulaire de démarrage qui n'est pas un «*Splash Screen*» pour l'application citée en exemple dans ce document. Un «*Splash Screen*» est en général un formulaire temporisé auto-refermable.





Exemple de formulaire de démarrage

### 9-3-2-1 - Génération d'un formulaire de démarrage temporisé

Pour concevoir un formulaire auto-refermable, vous pouvez exploiter l'événement **Timer** fixé à quelques secondes et parallèlement à cela, une routine se charge de faire quelques opérations de contrôle pour enfin afficher un formulaire de **Connexion** ou un **Menu Principal**.

Par exemple, il peut afficher un formulaire (*présentant rapidement l'application chargée, histoire de faire patienter l'utilisateur*) pendant le nombre de secondes défini avant de l'éliminer de l'écran et afficher ensuite le formulaire attendu.

*Dans l'exemple de formulaire de démarrage de la version des **Comptoirs** ci-dessus, un bouton **Connexion** permet de fermer le formulaire une fois l'utilisateur sélectionné. Selon le type d'application distribuée, ce formulaire de connexion avec **Login** peut également exiger un **mot de passe**.*

## 10 - Installation indépendante du Runtime Microsoft Access sur le poste client

Vous avez deux manières d'installer le Runtime Access, l'objectif se limitant à pouvoir exécuter des applications Microsoft Access.

Il n'y a pas d'intérêt à **installer manuellement** le Runtime si vous générez un package d'installation qui l'embarque, celui-ci étant installé automatiquement.

### 10-1 - Installation manuelle

Vous pouvez **télécharger le Runtime Microsoft Access 2013** directement ou le récupérer dans le dossier :

```
C : \Users\ %username% \AppData\Roaming\Microsoft\AddIns\pkwRuntime
```

puis l'installer aussitôt.

Vous installez le Runtime Microsoft Access comme vous installez l'application Access elle-même sauf si vous n'y êtes pas autorisé (*Droits d'administration*).

 **N'installez jamais le Runtime sur le poste développeur : cela est strictement inutile et resterait sans objet.**

Si vous devez installer manuellement le Runtime sur plusieurs postes en réseau, je vous conseille de copier le fichier **AccessRuntime\_fr-fr\_x86.exe** sur un serveur dédié, et de lancer son installation depuis celui-ci. Sinon, trouvez tout autre moyen de le faire (*CD-Rom, Clé USB etc.*) et lancez l'exécutable depuis le répertoire en question.

Vous pouvez revoir le processus d'installation du Runtime [ici](#).


### 10-2 - Installation automatique du Runtime

Comme vu précédemment, l'installation automatique est quant à elle supervisée par le **SETUP.EXE** d'installation que vous empaquetez lorsque vous générez un **Package** si vous avez coché la case appropriée.

Retournez sur la section concernée à la génération d'un **Package** à l'aide de l'**Assistant Package**, **plus haut dans ce document**.

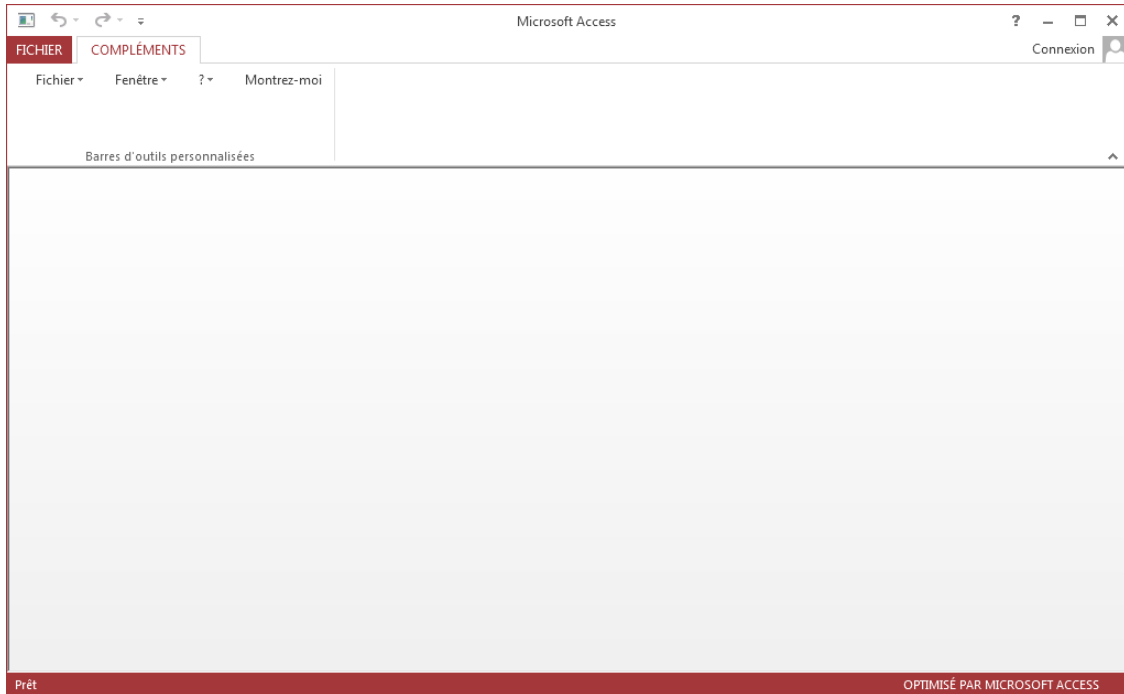
### 10-3 - Une fois le Runtime installé.

Une fois que les opérations d'installation manuelle ou automatique sont achevées, vous pouvez lancer une ou plusieurs applications Microsoft Access depuis un **Raccourci** ou depuis l'**Explorateur de fichiers**.

 **Rappel :** tout fichier **MDA/ MDB/ MDE** d'applications présentes sur le poste verront leurs associations mises à jour automatiquement.

## 11 - Différences entre Microsoft Access et l'environnement Runtime

Il est impossible de lancer le Runtime en tant qu'application. Si vous tentez d'exécuter MSACCESS.EXE alors que ce dernier n'est pas la version complète mais le Runtime (*ils portent le même nom*) vous recevrez un message d'erreur.



L'utilisateur ne peut pas accéder aux données dans la base de données autrement que par l'application elle-même. Il ne peut ni voir la structure des formulaires ni voir le code ou tout autre objet issu de l'application. Même si la base de données est située sur un serveur, l'utilisateur sera dans la même situation.

Si les applications qui fonctionnent avec le **Runtime de Microsoft Access** se comportent, à bien des égards, de la même manière que des applications qui fonctionnent avec la version complète de Microsoft Access (**Microsoft Access Runtime** est une version du logiciel Access ôtée des objets permettant la conception), il existe certaines différences qui peuvent avoir une incidence sur les approches de conception et de développement :

- le **Volet d'Exploration** (équivalent à la fenêtre Base de données des anciennes versions) et **Visual Basic Environment** sont masquées ;
- les **barres de Ruban** intégrées ne sont pas gérées mais vous pouvez néanmoins ajouter à votre application des barres de **Ruban personnalisées** ;
- certains menus (*y compris les menus contextuels*), fenêtres et commandes sont masqués ou désactivés ;
- la gestion d'erreurs de Visual Basic est nécessaire. **Les erreurs qui ne sont pas gérées par le code arrêtent l'application** sans aucun avertissement préalable. Pour cette raison, l'emploi des macros (*même incorporées*) **est plutôt déconseillé sauf si vous ne maîtrisez pas Visual Basic..**

Certaines séquences et combinaisons de touches sont désactivées :

Raccourcis clavier	Correspondance
<b>Ctrl+Pause</b>	Interrompt l'exécution du code ou de la macro.
<b>Maj</b>	Empêche l'exécution d'une macro AutoExec et ignore les

	propriétés de démarrage des bases de données lorsque vous ouvrez une base de données.
<b>F11</b>	Affiche la fenêtre du Volet d'exploration.
<b>F12</b>	Affiche la boîte de dialogue Enregistrer sous.
<b>Ctrl+G</b>	Affiche la fenêtre Débogage.
<b>Ctrl+N</b>	Ouvre une nouvelle base de données.
<b>Ctrl+Entrée</b>	Ouvre un objet en mode Création.

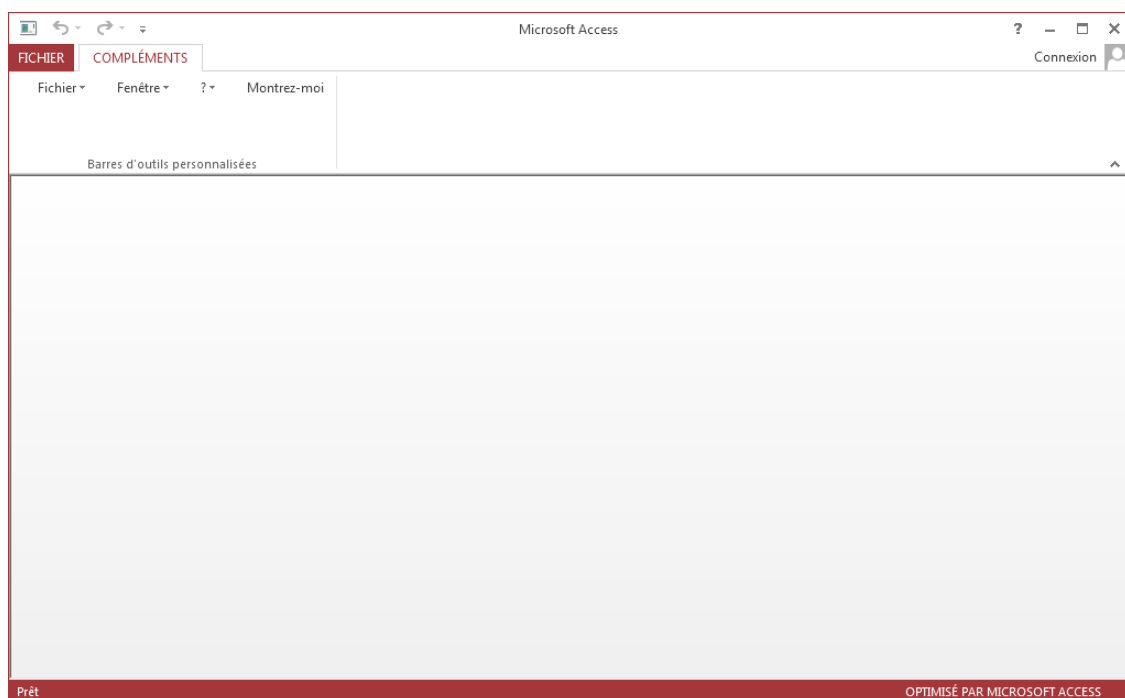
## 11-1 - Comment se présente le programme Runtime ?

Vous ne pouvez pas installer le **Runtime Access 2013** sur un poste pourvu de la version complète de Microsoft Access 2013 tout simplement parce que le **Runtime** est installé dans le même répertoire que Microsoft Access ; il ne sera pas installé si cette version est détectée.

Par ailleurs, si une version **antérieure complète de Microsoft Access** est présente sur le poste devant recevoir le **Runtime Access 2013**, il est fortement conseillé de procéder à des essais préalables, même si théoriquement, aucun conflit n'est intercepté excepté les associations de fichiers dans l'Explorateur Windows.

### 11-1-1 - Tentative de lancement de Microsoft Access Runtime

Il n'est pas possible de lancer le programme **MSACCESS.EXE** puisqu'il s'agit de la version **Runtime**.



Il est impératif de spécifier une base de données pour pouvoir l'utiliser.

## 12 - Les erreurs liées à l'installation

### 12-1 - Erreur d'exécution 429 lors de l'exécution d'un programme Access

#### Symptômes

Vous installez une application d'exécution Microsoft Access Runtime ou bien vous exécutez une application Access et le message d'erreur suivant apparaît :

**Erreur d'exécution '429' : Le composant ActiveX ne peut pas créer l'objet**

#### Cause

Vous avez installé l'application d'exécution sur un ordinateur où Microsoft Access n'a jamais été préalablement installé. Cette erreur peut se produire lorsqu'un composant n'a pas été enregistré dans le **Registre** du système. En effet, lorsque vous introduisez dans votre application un certain nombre de références externes et que votre programme contient des instructions **exploitant ces dernières** via VBA, ce message apparaîtra et votre application ne pourra fonctionner.

Dans l'exemple de code ci-dessous, il est supposé que vous avez tenté d'instancier la classe **clsPDFCreator**.

```
Set oPDFClass = CreateObject("PDFCreator.clsPDFCreator")
```

#### Résolution

Dans ce cas précis, seule l'installation de **PDFCreator\_1\_7\_X\_Setup.exe** résoudra le problème.

Concernant les composants de type COM nécessitant une inscription au Registre, vous devez les enregistrer manuellement ou via un programme dédié. Pour ce faire, vous devez localiser l'implantation physique de ces composants dans le système et également dans le Registre de Windows.

Vous devez alors faire en sorte que le poste cible dispose d'une version qui corresponde à la version exploitée par votre projet.

Dans le menu **Outils / Références** de **Visual Basic Editor**, vérifiez si les références sont correctes (Pas de terme **MANQUANT**) et que les références cochées correspondent bien à des bibliothèques correctement enregistrées (*s'il ne s'agit pas d'un cas avec le Runtime*).

Vous pouvez également exploiter la propriété **IsBroken** d'un objet **Reference** dans une boucle **For** pour les lister depuis une procédure externe d'une application de test dédiée à ce type de diagnostics.

#### Pour plus d'informations.

Pour plus d'informations, vous pouvez aussi jeter un œil ici à la page **Résolution de l'erreur 429 lors de l'automatisation des applications Office**.

### 12-2 - Erreur liée à l'utilisation de MouseWheel.dll

Depuis la version 2007 de Microsoft Access, vous n'avez plus besoin de ce composant, car la molette est désactivée nativement.

En admettant que vous l'ayez tout de même utilisé par le fait que vous ignoriez cela, vous êtes tenu de fournir le composant **MouseWheel.dll**.

Comme nous l'avons vu juste avant, une **erreur de type 429** est levée lorsqu'un composant ou une référence n'est pas correctement enregistré dans le Registre de Windows.

Concernant l'exploitation de la DLL **MouseWheel.dll**, c'est un peu particulier.



*Vous pouvez en cas de distribution de votre application sur différentes versions d'Access antérieures à 2007, vous rabattre sur la version qui ne **nécessite pas d'inscription** qui est disponible sur ce site ; elle a été développée par Thierry Gasperment alias **Arkham46**. Il vous propose un tutoriel sur l'exploitation de la **molette de la souris** sans avoir à inscrire le composant et donc plus d'erreur 429 pour ce cas précis.*

En revanche, si votre application est au format 2013, je vous conseille tout bonnement de supprimer ces appels devenus inutiles.



## 13 - Conclusion

Outre le fait que la politique de Microsoft vis-à-vis d'Access 2013 est d'entériner peu à peu les applications dites de bureau au profit de projets applicatifs *météorologiques* par temps gris ;o) - Il n'en est pas moins, et j'en sais quelque chose, un panel de développeurs qui ont été surpris de la suppression de cet outil d'emballage livré gracieusement jusqu'alors dans les versions 2007 et 2010.

De fait, ils continueront sans doute tout comme moi-même à développer des applications si petites soient-elles pour des petits groupuscules d'utilisateurs où la notion de souscription à une plateforme comme le *Cloud* peut paraître un tant soi peu surdimensionné.

En conséquence, j'ai pris la liberté (*et le risque aussi*) de me lancer dans ce projet en espérant vivement qu'il vous apporte un plus et anéantira votre déception.

Certes, il n'est pas abouti comme je l'entendais à cause de bugs et dysfonctionnements intrinsèques à Access 2013 mais il fonctionne et c'est au final ce qu'on lui demande.

Ainsi que vous avez pu le lire au cours de ces différents paragraphes, cet Assistant Package 2013 a été un peu amélioré en fonction de ma vision des choses.

Je suis bien évidemment disposé à faire évoluer cette application si vous pensez que de nouvelles fonctionnalités seraient les bienvenues.

## 14 - Correctifs de la version MSACCESS.EXE 2013

Dans la première version **15.0.4505** de MSACCESS.EXE existait un bug (*entre autres*) qui depuis a été corrigé dans un article publié le 9 avril 2013 sous la forme d'un hot-fix (*inclus dans le « Cumulative Update d'avril 2013 »*) :

<http://support.microsoft.com/kb/2768334> <http://support.microsoft.com/kb/2768334>

Aujourd'hui, il est inutile (*sauf en cas d'incompatibilité*) de télécharger ce hot-fix d'avril, puisque Microsoft a publié deux autres mises à jour d'Access 2013 (*KB2752093 en septembre et KB2768008 en octobre*).

Actuellement, la version **32 bits** recommandée par Microsoft est la version **15.0.4535** qui est incluse dans la mise à jour **KB2768008**. Elle est disponible via **Microsoft Update** ou téléchargeable depuis le 8 octobre courant à l'adresse suivante :

<http://support.microsoft.com/kb/2768008>

Une autre façon d'éviter des crashes serait d'installer la version **15.0.4535** de MSACCESS.EXE qui est incluse dans la mise à jour **KB2752093** qui est téléchargeable depuis le 10 septembre courant à l'adresse suivante :

<http://support.microsoft.com/kb/2752093> (*KB2752093*).

Cette mise à jour fournit les derniers correctifs pour les éditions 32 bits et 64 bits de Microsoft Access 2013.

En outre, cette mise à jour contient des améliorations de stabilité et de performances.

## 15 - Téléchargements/Downloads



### **Vous pouvez télécharger L'assistant Package 2013 32/64Bits sous 3 formes :**

- version complète avec les quatre versions US/FR 32/64Bits des exécutables du Runtime (1027 Mo)
- version complète avec deux versions FR 32/64Bits de l'exécutable du Runtime (512 Mo)
- version complète sans aucun exécutables du Runtime (1,1 Mo)

**Télécharger l'outil d'emballage pour Access 2013**



### **You can download Package Wizard 2013 32/64Bits in 3 forms :**

- full installation with four US/FR 32/64Bits versions of MSAccess Runtime (1027 Mo)
- full installation with the two US/FR 32/64Bits versions MSAccess Runtime (515 Mo)
- full installation without any version of MSAccess Runtime (1,1 Mo)

**Download the deployment tool for MSAccess 2013**