

# TUTORIAL : Créer le setup d'un programme VB 6

NIVEAU : INITIES

Pré requis pour comprendre ce tutorial :

- Connaître les fichiers nécessaires à l'installation d'un programme VB
- Connaître le système de dépendance entre un programme et ses librairies

Auteur : Dark sidious

Date de création : 11/09/2004

Version : 1.0

## SOMMAIRE

- I- Introduction
- II- Ce qu'il vous faut avant de commencer
- III- Etape 1 : La page de choix de l'installation
- IV- Etape 2 : Sélectionner un script
- V- Etape 3 : Type d'emballage
- VI- Etape 4 : Dossier d'emballage
- VII- Etape 5 : Les fichiers de l'installation
- VIII- Etape 6 : Types de fichiers d'installation
- IX- Etape 7 : Titre de l'installation
- X- Etape 8 : Dossier dans le menu Démarrer
- XI- Etape 9 : Destination des fichiers installés
- XII- Etape 10 : Nom du script
- XIII- Conclusion

## INTRODUCTION

Ca y est, vous avez réussi à faire votre premier programme que vous voulez distribuer. Vous l'avez compilé, testé chez vous, et tout marche bien. Vous l'envoyez alors à un ami pour savoir s'il marche aussi sur une autre machine, et là, bizarrement, votre programme ne marche plus ! Que s'est-il passé ? Quel est ce message qu'il m'affiche ? Le fichier untel.dll est introuvable ? C'est impossible, ça marchait très bien chez moi !

Et oui, avant de distribuer un programme, il faut songer à toutes les bibliothèques dont il est dépendant. Il faut savoir qu'un programme écrit en Visual Basic ne peut fonctionner sans les dll de VB, c'est comme ça. Alors au lieu de passer à un autre langage, lisez ce tutorial qui vous montrera comment résoudre ce problème de fichiers dll non reconnus en construisant un setup.

Pour avoir des informations supplémentaires pour la création de setup, si vous voyez des erreurs, si vous avez des problèmes ou que vous ne comprenez pas trop bien ce que j'essaie de vous apprendre dans ce tutorial, ou si vous avez des idées pour un nouveau tutorial, vous pouvez m'envoyer un message privé sur le site [www.ProgOtoP.com](http://www.ProgOtoP.com) à DarK Sidious.

Bonne lecture, et bonne compréhension.

## CE QU'IL VOUS FAUT AVANT DE COMMENCER

Fabriquer un setup d'un programme Visual Basic est très simple, mais encore faut-il avoir les bons outils. Dans ce tutorial, vous découvrirez uniquement l'assistant empaquetage et déploiement qui est donné avec Visual Basic, mais sachez qu'il en existe un bon nombre. Certains sont gratuits (notamment, InnoSetup, Nullsoft Install, Install Maker, et autre), et d'autres sont payants (InstallShield est l'un des plus connus dans ce domaine).

L'assistant de création de setup de VB est assez rudimentaire, mais il a l'avantage d'être livré avec VB, il est gratuit, et rapide à utiliser. Ses inconvénients ? Il est assez laids (les setup créés par son intermédiaire ressemblent à ses bons vieux setup du temps de Windows 3 !), il a des limites (notamment concernant l'enregistrement des ocx dans le registre et avec les fichiers dont votre programme est dépendant) et les setup créés sont assez lourds, et en plusieurs fichiers !

Il vous faut également un projet déjà compilé dont vous avez le code source pour pouvoir créer un setup de ce projet.

Pour lancer l'assistant empaquetage et déploiement, vous avez trois possibilités : par le biais du menu démarrer : il est dans le même dossier que Visual Basic si vous ne l'avez pas déplacé ou supprimé, par l'exécutable lui-même (PDCMDLN.EXE) qui se trouve dans le dossier PDWizard lui-même contenu dans le dossier Wizard du dossier de VB, ou alors directement par VB dans le menu des compléments : Compléments\Gestionnaire des compléments, chargez Assistant empaquetage et déploiement, et vous le trouverez alors dans le menu compléments avec les autres compléments chargés.

## ETAPE 1 : LA PAGE DE CHOIX DE L'INSTALLATION

Une fois l'assistant lancé, voici la page que vous devriez voir :



Comme vous pouvez le voir, vous avez 3 choix possible pour votre installation : l'empaquetage qui vous permet de créer des « paquets » constituant votre installation, le déploiement qui vous permet, comme son nom l'indique, de déployer votre installation vers un dossier ou un site internet, et la gestion des scripts qui vous permet... de gérer les scripts d'installation que vous aurez faits auparavant.

Dans ce tutorial, nous ne nous intéresserons qu'au premier choix : l'empaquetage qui est le principal travail de cet assistant.

Avant de cliquer sur le bouton Empaquetage, il vous faut sélectionner votre fichier de projet. Pour ce faire, cliquez sur le bouton Parcourir, et allez chercher sur votre disque dur le fichier vbp correspondant à votre projet. Ensuite, vous pouvez cliquer sur le bouton Empaquetage pour passer à l'étape suivante.

## ETAPE 2 : SELECTIONNER UN SCRIPT

L'étape de sélection d'un script existant vous permet de partir sur la base d'un script d'installation que vous avez déjà fait au préalable. Cela permet par exemple de partir sur un script déjà fait pour une version antérieure d'un programme sans devoir tout refaire de zéro.

Voici à quoi ressemble cette feuille :



Si vous n'avez jamais fait de scripts auparavant, vous n'aurez guère le choix.

Peut importe la liste des scripts disponibles, dans ce tutorial, nous partirons sur un nouveau script, donc sélectionner « aucun » dans la liste et cliquez sur le bouton Suivant > pour passer à la prochaine étape.

### ETAPE 3 : TYPE D'EMPAQUETAGE

Vous avez deux choix d'empaquetage dans cette étape : soit vous créez un logiciel standard (ce qui est le but de ce tutorial), soit vous créez un simple fichier de dépendance qui énumérera les fichiers dont dépend votre projet pour vous aider par exemple à lister les fichiers indispensables pour le bon fonctionnement de votre programme et les ajouter dans le script d'un autre logiciel de création d'installation.

Dans ce tutorial, nous ne verrons que la création d'un logiciel d'installation standard, mais la création d'un fichier de dépendance est quasi identique, même si elle ne produit pas le même résultat.

Sélectionnez donc « Logiciel d'installation standard » dans la liste comme ci-dessous :



Cliquez ensuite sur le bouton Suivant > pour passer à l'étape suivante.

## ETAPE 4 : DOSSIER D'EMPAQUETAGE

Cette étape vous demande de choisir un dossier qui contiendra tout les fichiers de l'empaquetage, c'est-à-dire, les fichiers de l'installation elle-même, mais aussi un dossier « Support » qui contiendras tout les fichiers dont dépend votre projet (dll, images, etc.). VB, par défaut, place ce dossier dans le dossier de votre projet, ce qui est judicieux pour éviter d'oublier où se trouve ce dossier :



Si un dossier Package est déjà présent, alors VB vous construit un nouveau dossier indexé comme dans l'exemple si dessus.

Vous pouvez très bien modifier le nom du dossier, ou le placer ailleurs. Vous pouvez même le placer sur un disque qui est sur un réseau.

Une fois que vous avez décider où placer votre dossier d'empaquetage, cliquez sur le bouton Suivant >.

## ETAPE 5 : LES FICHIERS DE L'INSTALLATION

Les choses sérieuses commencent avec cette étape. C'est ici que vous définissez les fichiers dont dépend votre projet, et les fichiers à inclure dans votre installation.



Par défaut, VB liste tout les fichiers de dépendances dont votre projet est renseigné, c'est-à-dire les librairies de VB, les ocx dont votre projet fait référence en early-binding, etc. Par contre, VB ne connaissant pas les fichiers dont votre projet dépend indirectement tels que les librairies utilisées en late-binding, les images chargées dynamiquement, les bases de données, etc., il vous faut les inclure dans votre installation « à la main ».

Prenons un cas classique : vous utiliser une base de données avec votre projet. Vous y faites référence par l'intermédiaire du contrôle Data de VB. Vous avez bien appris la leçon qu'un nom de fichier intégré « en dur » dans le contrôle ne vous permettez pas de déployer votre application sans risque (évités de définir la propriété DataBaseName dès la création, et utiliser une liaison dynamique vers votre base plutôt !), mais le problème, c'est que si vous faites une installation de votre projet sans ajouter votre base de données, votre programme, une fois installé, ne pourra pas trouver la base sur un autre ordinateur puisqu'elle n'y sera pas !

Donc, il vous faut ajouter les fichiers dans votre script pour qu'il soit intégré dans l'installation et ainsi être déployés en même temps que votre programme.

Pour cela, cliquez sur le bouton Ajouter, et sélectionnez les fichiers qu'il vous faut.

Vous pouvez également ne pas vouloir déployer certains fichiers qui sont listés par l'assistant. A ce moment là, il vous suffit de les décocher dans la liste, et ils ne seront pas ajoutés à l'installation.

Une fois que vous avez la liste complète des fichiers à installer, cliquez sur le bouton Suivant >.

## ETAPE 6 : TYPE DE FICHIERS D'INSTALLATION

L'assistant d'empaquetage et déploiement possède une option intéressante qui est de pouvoir découper les fichiers d'installation en plusieurs parties. Cela peut-être pratique pour un déploiement sur des disquettes (bien que très peu de personnes utilisent dorénavant ce format !), ou sur un site internet : plus les fichiers sont petits, moins il y a de chance que le téléchargement d'une partie de l'installation échoue. Par contre, les chances de corruption augmentent...

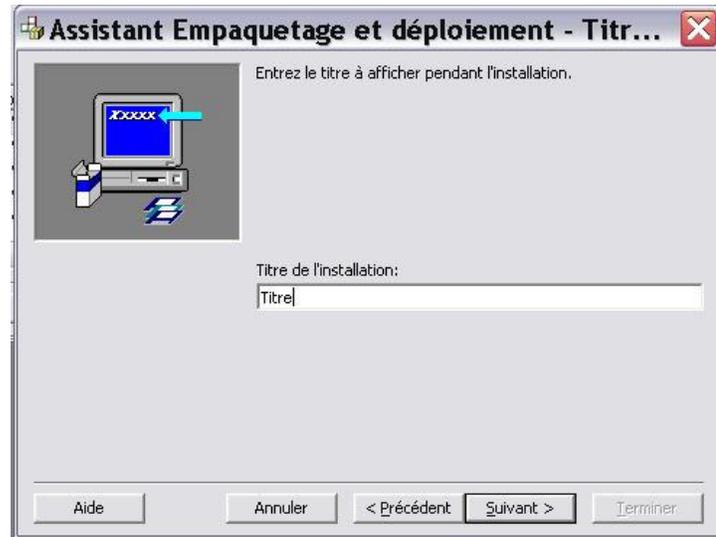


Dans cette étape, vous pouvez sélectionner le type de morcellement à effectuer : soit vous ne faites qu'un seul fichier .cab qui contiendras tout les fichiers compressés (ce n'est pas ce fichier qui permet d'installer votre programme, mais contient tout ce qui est nécessaire pour l'installation), soit vous pouvez faire plusieurs fichiers cab dont vous définissez la taille. L'assistant découpera alors automatiquement le fichier en plusieurs selon la taille choisie.

Sélectionnez le type de morcellement de fichier que vous voulez appliquer, et cliquez sur le bouton Suivant >.

## ETAPE 7 : TITRE DE L'INSTALLATION

Petite étape de personnalisation : elle vous permet de définir le titre que vous verrez lorsque vous installerez votre programme. Ce n'est pas grand-chose, mais c'est toujours mieux que d'avoir un titre aléatoire ou incompréhensible, voir même pas de titre du tout !



Une fois que vous avez saisi le titre de votre application, cliquez sur le bouton Suivant >.

## ETAPE 8 : DOSSIER DANS LE MENU DEMARRER

Une fois votre programme installé, il faut que les utilisateurs puissent y accéder facilement et rapidement. Pour cela, rien de tel qu'un raccourci dans le menu démarrer (un raccourci sur le bureau ou dans la barre de lancement rapide aurait également été le bienvenue !). C'est ce que vous propose cette étape :



Vous pouvez donc créer un nouveau groupe (comprenez par là un nouveau dossier dans l'arborescence du menu) ou un nouvel élément (une nouvelle icône référence un fichier que vous sélectionnez).

Vous êtes assez libre pour choisir où placer votre raccourci.

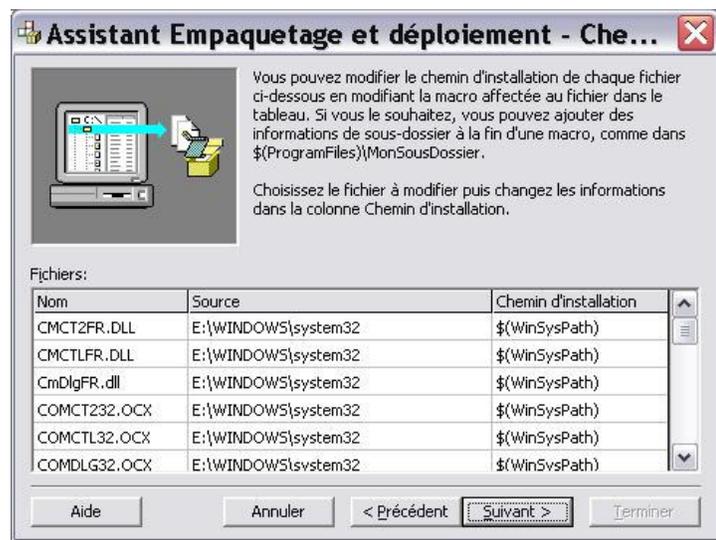
Lorsque vous l'aurez placé, cliquez sur le bouton Suivant >.

## ETAPE 9 : DESTINATION DES FICHIERS INSTALLES

L'une des principales étapes de l'assistant d'empaquetage et de déploiement concerne la destination des fichiers lorsqu'ils seront installés. Dans cette étape, vous pouvez choisir dans quel dossier chacun des fichiers de l'installation seront copiés. Normalement, l'assistant vous facilite la tâche en copiant par défaut les fichiers là où il faut, mais il ne sait pas où est-ce qu'il doit copier les fichiers ajoutés manuellement. Pour chaque fichier, vous avez plusieurs choix de répertoire possible, tout en ayant la possibilité de créer des sous répertoires dans le dossier sélectionné en entrant manuellement : « \NomDuDossier ».

Il s'agit bien souvent de l'étape qui peut poser problème pour le déploiement de votre application : selon le chemin d'accès aux fichiers, votre application peut, ou ne peut pas, accéder aux fichiers qu'il lui faut !

Voici à quoi ressemble la feuille de cette étape :



Les principaux dossiers que vous avez à votre disposition sont :

- WinSysPath : le dossier Système qui doit contenir en théorie tout les fichiers qui nécessite un enregistrement dans la base de registre, tels que les dll, activeX, ou autre librairies.
- WinPath : le dossier de Windows qui peut contenir par exemple des fichiers que vous voulez masquer, ou des fichiers de configuration.
- CommonFiles : le dossier des fichiers en commun. Vous devriez y copier tous les fichiers dont partage votre application avec d'autres programmes, tels que des bases de données ou des images communes.

- ProgramFiles : le dossier des programmes. Vous devriez y copier tout les fichiers exécutables (dans des sous répertoires) dont utilise votre application.
- MSDAOPath : Le dossier contenant les librairies concernant DAO. Pratique si vous voulez modifier la version de DAO déjà installée, mais attention à ne pas écraser une version plus récente !
- AppPath : le dossier qui contiendras votre application. Placez-y tout les fichiers dont vous faites référence dans votre code par des App.Path.
- Font : le dossier des polices de caractères de Windows. Vous devriez y placer tout les fichiers de polices dont votre application se sert.

Tous les fichiers ocx, activeX, dll qui nécessite une inscription dans le registre seront automatiquement inscrits par le programme d'installation, mais il arrive toutefois que la procédure d'enregistrement dans le registre échoue.

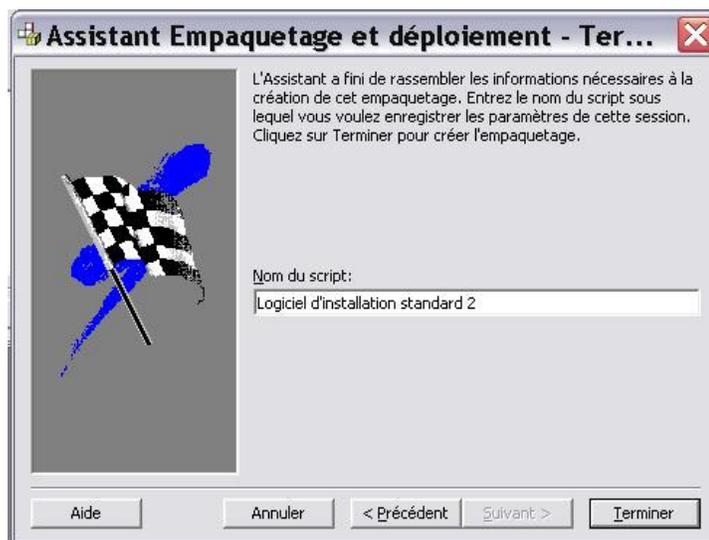
Lorsque vous aurez fini de définir le dossier de destination de chacun de vos fichiers, cliquez sur Suivant >.

## ETAPE 10 : NOM DU SCRIPT

Ca y est, vous êtes arrivé au bout du tunnel ! Il ne vous reste plus qu'à sauvegarder le script que vous venez de faire en suivant les étapes de l'assistant une par une.

Ce script n'est pas très important, mais vous permettra tout de même de le charger si vous voulez refaire une installation, ou si vous faites une mise à jour de votre application, et que vous ne voulez pas répéter la procédure de création du programme d'installation.

Il vous suffit donc de choisir un nom pour votre script, et de cliquer sur le bouton Terminer. L'assistant créera alors votre programme d'installation dans le dossier que vous avez spécifié au début, et vous affichera alors un message vous informant des erreurs éventuelles qui se sont déroulées lors de la compilation du programme d'installation.



## ETAPE 10 : CONCLUSION

Une fois votre programme d'installation construit, il ne vous reste plus qu'à le tester sur une autre machine qui ne possède pas les fichiers dont dépend votre application. Il est possible qu'il y ait des erreurs même après une installation réussie. Cela peut venir de beaucoup de paramètres : des bibliothèques qui n'ont pas été incluses, des fichiers manquants, etc. Un programme tel que l'assistant d'emballage et de déploiement n'est pas sans bug, mais vous permet tout de même de créer des installations qui marchent assez bien, même s'il oublie de temps en temps d'inclure des bibliothèques importantes pour votre projet telles que ADO, ou des contrôles ActiveX. Le programme d'installation n'est certes pas très joli, mais permet de déployer son application et d'installer les bibliothèques de VB sur des ordinateurs qui ne possèdent pas Visual Basic.

Cependant, si vous voulez obtenir un programme d'installation bien plus fonctionnel et esthétique, je vous conseille d'utiliser des logiciels plus puissants tels que InnoSetup ou NullSoft Installer, qui sont gratuits et performants.

J'espère que la lecture de ce tutorial vous a satisfaite, et que vous continuerez à programmer sous Visual Basic. Et si vous avez des remarques, des commentaires à me faire, n'hésitez pas à m'envoyer vos messages sur le site [www.ProgOtoP.com](http://www.ProgOtoP.com)

Bonne programmation à tous.