

Distribuer vos applications VB6 avec InnoSetup

par [ThierryAIM \(home\)](#)

Date de publication : 27/01/2008

Dernière mise à jour :

InnoSetup est un logiciel gratuit de déploiement bien connu des développeurs, pour sa simplicité d'utilisation.
Ses possibilités sont extraordinaires, et nous allons en découvrir les bases, afin de créer des programmes d'installation "professionnels".

I - Introduction.....	3
II - Mon premier "Setup".....	4
II-1 - Générer le fichier Setup.Ist.....	4
II-2 - Importer un fichier Setup.Ist dans InnoSetup/IsTools.....	5
III - Personnaliser un peu notre Setup.....	7
III-1 - La section [Setup].....	7
III-2 - La section [Files].....	7
III-3 - La section [Dirs].....	8
III-4 - La section [Tasks].....	8
III-5 - La section [Icons].....	9
III-6 - La section [INI].....	9
III-7 - La section [Registry].....	10
III-8 - La section [Run].....	10
IV - Installation multilingue.....	11
IV-1 - La section [Languages].....	11
IV-2 - La section [CustomMessages].....	12
V - Conclusions.....	13
V-1 - Liens.....	13
V-2 - Remerciements.....	13

I - Introduction

Créer un programme d'installation constitue l'étape finale du développement d'une application.

Compte tenu de ma spécialité, ce tutoriel est forcément très orienté Visual Basic, et principalement la section II. Mais **InnoSetup** est un installateur universel, et peut tout à fait être utilisé pour la distribution d'applications conçues avec d'autres langages de programmation (Delphi, par exemple, pour ne citer que celui-ci).

Bien sûr, pour Visual Basic, il existe l'assistant d'installation et de déploiement, livré de base avec avec VB6-édition professionnelle ou +

Mais la moindre modification étant tellement fastidieuse, qu'elle en devient vite dissuasive !

InnoSetup constitue une bonne alternative à cet inconvénient, offrant de multiples possibilités de personnalisation, de façon très simple, sans se prendre la tête à modifier le Setup.bas de VB6 !

IsTools est le complément indispensable de **InnoSetup** .

Nécessaire pour importer facilement les fichiers d'installation **Setup.lst**, généré par l'assistant VB, et disponible en français, il vous permet de personnaliser la base de votre **Setup** sans écrire une seule ligne de script ou de code !

Vous trouverez les liens utiles dans la section **Liens**

II - Mon premier "Setup"

La principale difficulté, pour obtenir un setup complet et opérationnel, est de trouver toutes les dépendances de votre projet VB6.

Si vous n'êtes pas un expert en ce domaine, je vous conseille d'utiliser, dans une première phase, l'assistant de déploiement de VB6, afin d'obtenir un fichier *Setup.lst*, qui contiendra ces dépendances.

Si vous ne disposez pas de l'assistant VB6 (distribué uniquement à partir des version professionnelles de VB6), il existe d'autres solutions, mais nous parlerons plus tard !

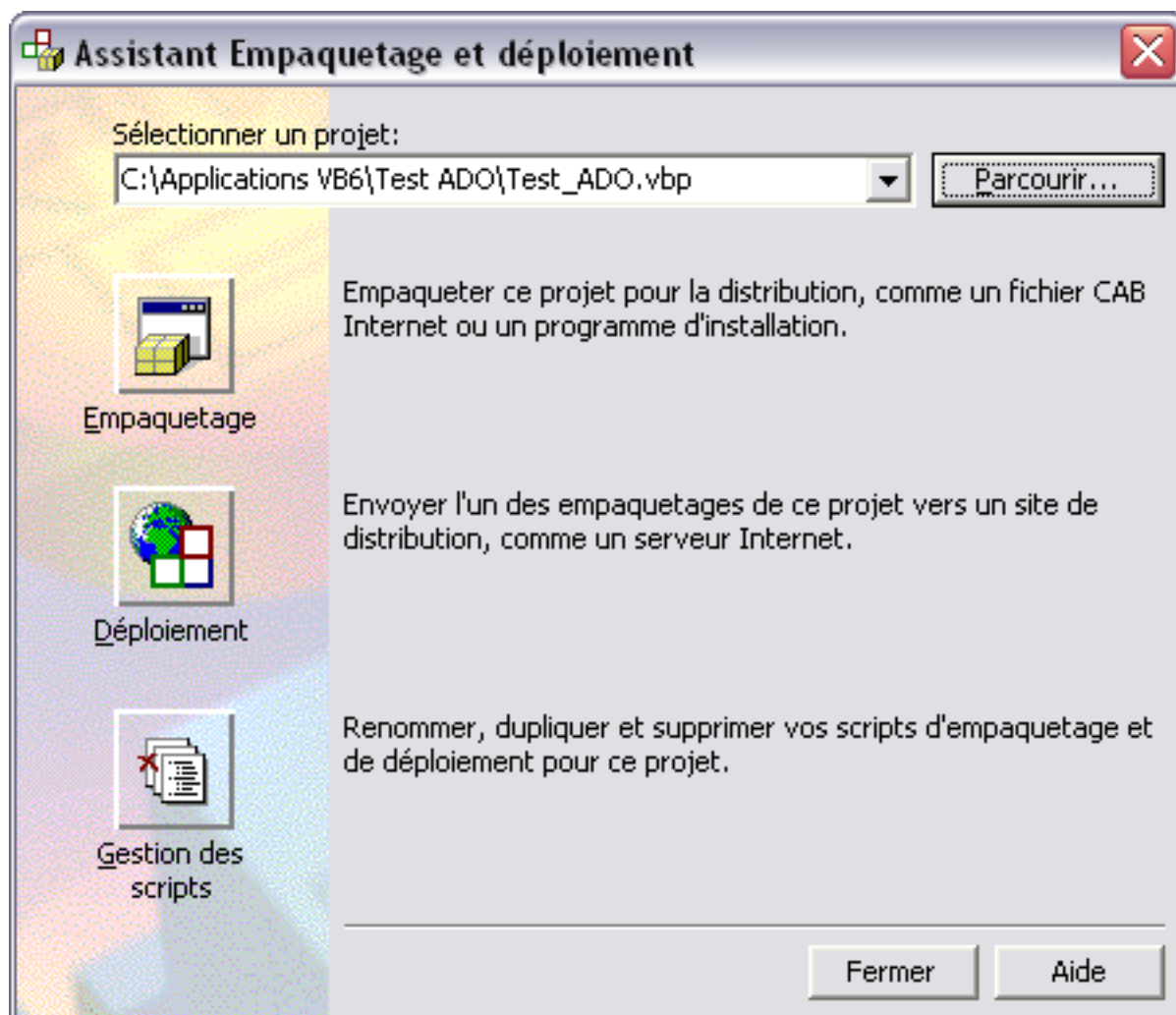
II-1 - Générer le fichier Setup.lst

Je ne ferai aucun commentaire ici, concernant l'assistant de déploiement VB6.

Nous allons l'utiliser comme simple passerelle, afin d'obtenir la liste des fichiers de dépendances de notre projet.

Pour toute question relative à l'utilisation de ce logiciel, je vous convie sur notre [Forum installation et déploiement VB6](#)

- Lancer *l'assistant Installateur et déploiement de VB6* :



Fenêtre d'accueil de l'assistant VB6 Empaquetage et déploiement

- Sélectionner votre fichier projet VB6 en cliquant sur le bouton "**Parcourir...**" (fichier de type .vbp ou .vbg)

- Choisissez l'option "Emballage" et continuez jusqu'à la fin, sans trop vous poser de questions, ce n'est pas le plus important.

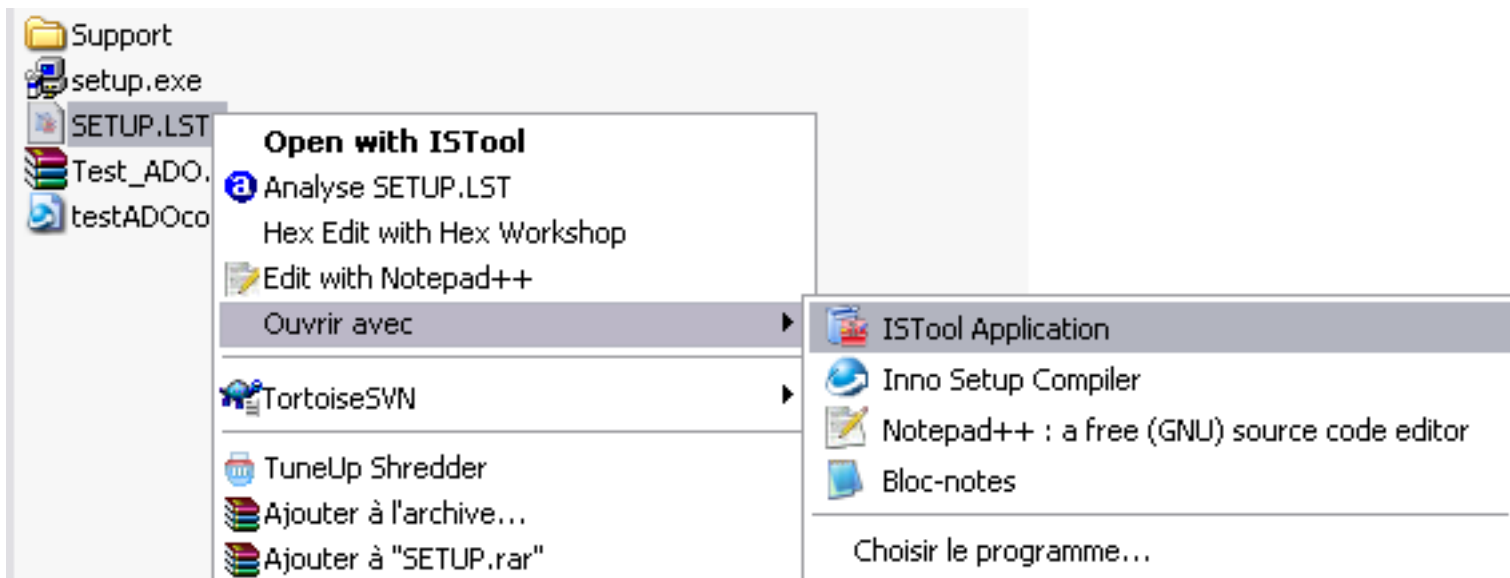
A ce stade, (si vous avez conservé les options de base de l'assistant), vous avez généré dans le répertoire `.MonProjet\Package\` un fichier `Setup.lst`.

Repérez ce fichier, il sera la base de notre installation avec **InnoSetup**.

⚠ Certains composants ocx en freeware peuvent avoir des dépendances cachées, non repérées par l'assistant VB.
 Je n'ai pas de solution miracle et ne peux que vous conseiller de bien tester votre Setup avant diffusion (j'en ai fait l'amère expérience)

II-2 - Importer un fichier Setup.lst dans InnoSetup/IsTools

Vous avez repéré votre fichier `Setup.LST`, alors faites un click droit dessus, dans l'explorateur Windows :



Puis **"Open with IsTool"** ou **"Ouvrir avec" >> IsTools Application**
 - si l'application n'apparaît pas dans la liste, sélectionner **"Choisir le programme..."** et rechercher **IsTools.exe**

Voici ce que nous devons obtenir :

```

Script d'import IsTools

; 'C:\Applications VB6\Test ADO\Package\SETUP.LST' imported by ISTool version 5.2.1

[Setup]
AppName=MyProgram
AppVerName=MyProgram
DefaultDirName={pf}\MyProgram
DefaultGroupName=MyProgram

[Files]
; [Bootstrap Files]
; @COMCAT.DLL,$(WinSysPathSysFile),$(DLLSelfRegister),,6/1/98 1:00:00 AM,22288,4.71.1460.1
Source: COMCAT.DLL; DestDir: {sys}; Flags: restartreplace uninsneveruninstall sharedfile regserver
; @VB6FR.DLL,$(WinSysPath),,$(Shared),10/2/00 1:00:00 AM,119568,6.0.89.88
Source: VB6FR.DLL; DestDir: {sys}; Flags: promptifolder sharedfile
; @STDOLE2.TLB,$(WinSysPathSysFile),$(TLBRegister),,6/3/99 1:00:00 AM,17920,2.40.4275.1
Source: STDOLE2.TLB; DestDir: {sys}; Flags: restartreplace uninsneveruninstall sharedfile regtypelib
; @ASYCFILT.DLL,$(WinSysPathSysFile),,,3/8/99 1:00:00 AM,147728,2.40.4275.1
    
```

Script d'import IsTools

```


Source: ASYCFILT.DLL; DestDir: {sys}; Flags: restartreplace uninsneveruninstall sharedfile
; @OLEPRO32.DLL,$(WinSysPathSysFile),$(DLLSelfRegister),,3/8/99 1:00:00 AM,164112,5.0.4275.1
Source: OLEPRO32.DLL; DestDir: {sys}; Flags: restartreplace uninsneveruninstall sharedfile regserver
; @OLEAUT32.DLL,$(WinSysPathSysFile),$(DLLSelfRegister),,4/12/00 1:00:00 AM,598288,2.40.4275.1
Source: OLEAUT32.DLL; DestDir: {sys}; Flags: restartreplace uninsneveruninstall sharedfile regserver
; @msvbvm60.dll,$(WinSysPathSysFile),$(DLLSelfRegister),,2/23/04 9:42:40 PM,1386496,6.0.97.82
Source: msvbvm60.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall sharedfile regserver

; [Setup1 Files]
; @DBRPRFR.DLL,$(WinSysPath),,$(Shared),7/13/98 1:00:00 AM,33280,6.0.81.63
Source: DBRPRFR.DLL; DestDir: {sys}; Flags: promptifolder sharedfile
; @stdftfr.dll,$(WinSysPath),,$(Shared),11/21/00 4:46:34 AM,6656,6.0.81.63
Source: stdftfr.dll; DestDir: {sys}; Flags: promptifolder sharedfile
; @MSSTDFMT.DLL,$(WinSysPath),$(DLLSelfRegister),$(Shared),2/23/04 1:00:00 AM,119808,6.1.97.82
Source: MSSTDFMT.DLL; DestDir: {sys}; Flags: promptifolder regserver sharedfile
;
; >> J'en ai enlevé un peu au milieu !!!
;
; @msadox.dll,$(WinSysPath),$(DLLSelfRegister),$(Shared),8/5/04 1:00:00 PM,200704,2.81.1117.0
Source: msadox.dll; DestDir: {sys}; Flags: promptifolder regserver sharedfile

; @Test_ADO.exe,$(AppPath),,,10/12/07 9:51:35 PM,53248,1.0.0.0
Source: MyProgram.exe; DestDir: {app}; Flags: promptifolder

[Icons]
Name: {group}\MyProgram; Filename: {app}\MyProgram.exe; WorkingDir: {app}
    
```

Les ; vous montrent les lignes importées du fichier Setup.LST (sous forme de commentaires); ces lignes peuvent être supprimé.

 *A ce niveau, se situe le principal problème de **IsTools**, et vous l'aurez surement vite remarqué : il n'est pas capable d'inclure le chemin du répertoire source pour les fichiers listés !*

La sous-section **[Bootstrap Files]** liste les fichiers du runtime VB6.

Vous pouvez placer ces fichiers dans un répertoire particulier, afin de pouvoir vous en resservir à chaque installation :

Par exemple, ici dans : c:\Redist\VB_Runtime\

```

[Files]
; [VB Runtime files]
Source: c:\Redist\VB_Runtime\stdole2.tlb; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile regtypelib
Source: c:\Redist\VB_Runtime\msvbvm60.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile regserver
Source: c:\Redist\VB_Runtime\oleaut32.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile regserver
Source: c:\Redist\VB_Runtime\olepro32.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile regserver
Source: c:\Redist\VB_Runtime\asycfilt.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile
Source: c:\Redist\VB_Runtime\comcat.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile regserver
Source: c:\Redist\VB_Runtime\Vb6fr.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
promptifolder sharedfile
    
```

Pour les autres fichiers, vous devrez définir le répertoire source.

III - Personnaliser un peu notre Setup

Il me sera impossible de décrire ici, toutes les possibilités de **InnoSetup**
 Nous allons donc nous contenter des plus courantes.

III-1 - La section [Setup]

Il y aurait beaucoup à dire sur cette section, tant ses possibilités et options sont nombreuses.

Un bref aperçu avec ce que nous avons obtenu lors de la conversion du fichier *Setup.lst*, assorti de commentaires d'explication :

```
[Setup]
; Le nom "commercial" de votre application qui sera utilisé dans le titre des fenêtres et les dialogues principaux
AppName=MyProgram
; Le nom et la version de votre programme (visible dans la page d'accueil)
AppVerName=MyProgram v1.0
; Le répertoire d'installation par défaut
DefaultDirName={pf}\MyProgram
; Le groupe de programme d'installation par défaut, dans le menu "Démarrer"
DefaultGroupName=MyProgram
```

Bien d'autres options sont disponibles dans cette section :

Un aperçu des options configurables dans la section [Setup]

```
...
; Définir le nom du programme d'installation
OutputBaseFilename=MyProgram_Setup
; Définir le répertoire d'enregistrement du programme d'installaton compilé
OutputDir=..\Package

; Imposer un compte administrateur pour l'installation (nécessaire pour l'enregistrement de dll et ocx)
PrivilegesRequired=admin

; Fixer les versions minimales de Windows requises pour l'installation
MinVersion=0,5.0.2195

; Définir un répertoire source par défaut dans lequel seront recherché les fichiers de la section [Files],
; si le répertoire source n'est pas précisé
SourceDir=C:\MyProgram

; Afficher un texte d'information avant l'installation
InfoBeforeFile=.\InfosBefore.txt
; Afficher un texte d'information après l'installation
InfoAfterFile=.\InfosAfter.txt
; Afficher une page contenant le texte de licence
LicenseFile=.\Licence.txt

; Définir un Copyright
AppCopyright=Copyright© 2008 thierryaim

; Personnaliser les images des fenêtres de l'installateur
WizardImageFile=C:\Program Files\Inno Setup 5\WizModernImage-IS.bmp
WizardSmallImageFile=C:\Program Files\Inno Setup 5\WizModernSmallImage-IS.bmp
```

 **Toutes les options de la section [Setup] sont aisément configurables dans le menu "Projet" >> "Options de l'installateur" de *IsTools*.**

*Si vous avez la moindre question, rendez-vous sur le **forum installation VB***

III-2 - La section [Files]

Elle a été créée automatiquement lors de l'importation du fichier .lst, à l'étape précédente.

Dans cette section, vous pouvez ajouter tous les dossiers ou fichiers devant être joints à l'application :

Principaux paramètres :

- **Source** (obligatoire) : nom (et répertoire) du fichier source
- **DestDir** (obligatoire) : répertoire de destination
- **DestName** (facultatif) : nom du fichier destination, si différent du nom d'origine

```
[Files]
;[Redist Files]
Source: c:\Redist\DCOM\dcom98.EXE; DestDir: {app}; Flags: promptifolder deleteafterinstall
nocompression; Tasks: DCOM98
; [Application Files]
Source: MyProgram.exe; DestDir: {app}; Flags: promptifolder
Source: MyProgram.ini; DestDir: {app}; Flags: promptifolder
Source: lisezmoi.txt; DestDir: {app}; Flags: promptifolder isreadme

; Ajouter au package un dossier et tout son contenu
Source: images\*.*; DestDir: {app}\images
```

Les principaux Flags de cette section :

- **promptifolder** : Affiche un avertissement si le fichier à installer est plus ancien qu'un fichier identique, déjà présent sur le système cible
- **isreadme** : indique que le fichier est de type lisezmoi. Une boîte de dialogue sera automatiquement proposée en fin d'installation
- **sharedfile** : déclare un fichier à installer comme pouvant être partagé par plusieurs applications
- **restartreplace** : provoquera un redémarrage du système si le fichier doit être remplacé
- **regserver** : après installation, indique que le fichier (de type ocx ou dll) doit être enregistré à l'aide de regsvr32 (nécessite PrivilegesRequired=admin)
- **regtypelib** : après installation, indique que le fichier (de type .tlb) doit être enregistré comme nouvelle librairie de type (nécessite PrivilegesRequired=admin)
- **uninsneveruninstall** : précise que le fichier ne doit jamais être désinstallé
- **deleteafterinstall** : fichier devant être effacé après installation
- **allowunsafefiles** : nécessaire, si vous devez inclure à votre package, des fichiers situés dans les répertoires du système

 *Consulter la liste des fichiers non sûrs à ne pas redistribuer (dépendant des différentes plateformes), dans l'aide en ligne d'[InnoStup](#)*

Pour les autres **Flags**, je vous invite à lire l'aide en ligne de [Inno-Setup](#).

III-3 - La section [Dirs]

permet, lors de l'installation, la création de sous-dossiers du répertoire de l'application

L'exemple ci-dessous crée les dossiers **data** et **bin** comme sous-dossiers du répertoire de l'application **{app}** :

```
[Dirs]
Name: {app}\data
Name: {app}\bin
```

III-4 - La section [Tasks]

Comme son nom l'indique, cette section va nous permettre de définir des tâches à exécuter ou non, suivant les choix de l'utilisateur.

Si cette section est présente dans le script, l'installateur générera automatiquement une page de sélection des tâches à accomplir, en fonction des paramètres spécifiés :

[Tasks]

```
Name: InstalRTVB6; Description: Installer le Runtime VB6
Name: DCOM98; Description: D&COM 1.3 (Windows 98); GroupDescription: Compléments;; Flags:
checkedonce; MinVersion: 4.1,0; OnlyBelowVersion: 4.9,0
Name: DesktopIcon; Description: Créer une icône sur le &bureau; GroupDescription: Icônes; Flags:
checkedonce
Name: QuickLaunchIcon; Description: Créer une icône de &démarrage rapide; GroupDescription:
Icônes; Flags: unchecked
```

L'exécution du code sera conditionné par la préselection de la tâche **[Tasks]** :
 Reprenons l'exemple de l'installation des fichiers du runtime VB6 :

[Files]

```
; [VB Runtime files]
Source: c:\Redist\VB_Runtime\stdole2.tlb; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile regtypelib; Tasks: InstalRTVB6
Source: c:\Redist\VB_Runtime\msvbvm60.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile regserver; Tasks: InstalRTVB6
Source: c:\Redist\VB_Runtime\oleaut32.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile regserver; Tasks: InstalRTVB6
Source: c:\Redist\VB_Runtime\olepro32.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile regserver; Tasks: InstalRTVB6
Source: c:\Redist\VB_Runtime\asycfilt.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile; Tasks: InstalRTVB6
Source: c:\Redist\VB_Runtime\comcat.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
sharedfile regserver; Tasks: InstalRTVB6
Source: c:\Redist\VB_Runtime\Vb6fr.dll; DestDir: {sys}; Flags: restartreplace uninsneveruninstall
promptifolder sharedfile; Tasks: InstalRTVB6
```

L'installation sera effectuée si et seulement si la tâche **InstalRTVB6** a été sélectionnée précédemment.

III-5 - La section [Icons]

Cette section permet de définir les icônes à installer.

L'exemple ci-dessous propose la création des icônes :

- principale du programme
- de désinstallation du programme
- pour le fichier "Lisezmoi"
- sur le bureau, conditionné par la tâche *DesktopIcon*, définie précédemment dans la section **[Tasks]**
- de lancement rapide, conditionné par la tâche *QuickLaunchIcon*, définie précédemment dans la section **[Tasks]**

[Icons]

```
Name: {group}\MyProgram; Filename: {app}\MyProgram.exe; WorkingDir: {app}
Name: {group}\Uninstall MyProgram; Filename: {uninstall.exe}
Name: {group}\Lisezmoi; Filename: {app}\Lisezmoi.txt; WorkingDir: {app}
Name: {userdesktop}\MyProgram; Filename: {app}\MyProgram.exe; WorkingDir: {app}; Tasks: DesktopIcon
Name: {userappdata}\Microsoft\Internet Explorer\Quick Launch
\MyProgram; Filename: {app}\MyProgram.exe; WorkingDir: {app}; Tasks: QuickLaunchIcon
```

III-6 - La section [INI]

permet, lors de l'installation, d'écrire dans un fichier .ini associé à l'applicaton :

Paramètres :

- **Filename** : nom du fichier .ini
- **Section** : nom de la section du fichier .ini
- **Key** : nom de la clé du fichier .ini
- **String** : Valeur de la clé

[INI]

```
Filename: {app}\MyProgram.ini; Section: Setup; Key: Language; String: Français; Languages: fr; Flags: createkeyifdoesntexist
Filename: {app}\MyProgram.ini; Section: Setup; Key: Language; String: English; Languages: en; Flags: createkeyifdoesntexist
```

III-7 - La section [Registry]

permet d'inscrire dans le registre des informations à l'installation :

[Registry]

```
; Création de la clé primaire
Root: HKCU; Subkey: Software\VB and VBA Program Settings\MyProgram; Flags: uninsdeletekey
; Inscription des valeurs de clés secondaires
Root: HKCU; Subkey: Software\VB and VBA Program Settings\MyProgram\Setup; ValueType: string; ValueName: LicenseName; ValueData: {sysuserinfoorg}
Root: HKCU; Subkey: Software\VB and VBA Program Settings\MyProgram\Setup; ValueType: string; ValueName: Language; ValueData: French.lng; Languages: fr
Root: HKCU; Subkey: Software\VB and VBA Program Settings\MyProgram\Setup; ValueType: string; ValueName: Language; ValueData: English.lng; Languages: en
```

III-8 - La section [Run]

permet de définir les tâches à exécuter lorsque l'installation est terminée

[Run]

```
; START DCOM
Filename: {app}\dcom98.exe; Parameters: /r:n /q:u; WorkingDir: {tmp}; Flags: skipifdoesntexist; Tasks: DCOM98; MinVersion: 4.1,0
```

IV - Installation multilingue

Parmi ses nombreuses possibilités, **Inno-Setup** permet, de base, les installation multilingue.

Pour celà, vous devez écrire :

- La section **[Languages]** : permet définir les langues d'installation disponibles pour votre setup.
- La section **[CustomMessages]** : contiendra tous les textes et messages, utilisés dans le setup, dans les différentes langues déclarées

IV-1 - La section [Languages]

La liste des langues pré-formatées est contenue dans le dossier d'installation de Inno-Setup; normalement votre répertoire "C:\Program Files\Inno Setup 5\Languages"

La langue par défaut, vous l'aurez remarqué, est *English*, associée au fichier de langue *default.isl*

Pour une installation en français uniquement, vous n'aurez qu'à définir la langue, en remplacement de la langue par défaut (anglais) :

Pour un installateur qui parle français :

```
[Languages]
Name: fr; MessagesFile: compiler:Languages\French.isl
```

Pour une installation multilingue, vous devez définir un nom (abréviation courte, de préférence) pour chaque langue déclarée.

Nous nous contenterons, dans les exemples ci-après, de deux langues : français et anglais :

Exemple de section [Languages]

```
[Languages]
Name: en; MessagesFile: compiler:Default.isl
Name: fr; MessagesFile: compiler:Languages\French.isl
; Ajouter ci-dessous toutes les langues que vous souhaitez rendre disponibles. Exemples :
Name: gr; MessagesFile: compiler:Languages\German.isl
Name: it; MessagesFile: compiler:Languages\Italian.isl
Name: sp; MessagesFile: compiler:Languages\Spanish.isl
```

Si cette section est présente dans votre programme Setup, et comporte plus d'une langue déclarée, la boite de dialogue de sélection de la langue d'installation sera, par défaut, affichée automatiquement au lancement du programme :



Boite de dialogue : Selection de la langue d'installation

IV-2 - La section [CustomMessages]

Définir ici tous les textes devant apparaître dans les différentes pages de votre Setup.

Chaque message doit être préfixé du nom (Name) défini dans la section **[Languages]**

Exemple de section [CustomMessages]

```
[CustomMessages]
; Français
fr.TaskIconGroup=Icônes :
fr.TaskDesktopIcon=Créer une icône sur le &bureau
fr.TaskQuickLaunchIcon=Créer une icône de &démarrage rapide

; English
en.TaskIconGroup=Icons :
en.TaskDesktopIcon=Create a desktop icon
en.TaskQuickLaunchIcon=Create a quicklaunch icon
```


Ensuite, dans le script, remplacer les messages texte par leur équivalent avec la syntaxe : {cm:monmessage}
Le texte sera affiché en fonction de la langue choisie au démarrage de l'installation.

Exemple avec la section [Tasks]

```
[Tasks]
Name: DesktopIcon; Description: {cm:TaskDesktopIcon}; GroupDescription: {cm:TaskIconGroup}; Flags:
checkedonce
Name:
QuickLaunchIcon; Description: {cm:TaskQuickLaunchIcon}; GroupDescription: {cm:TaskIconGroup}; Flags:
unchecked
```

Section [Files], conditionnée par la langue d'installation :

```
[Files]
Source: C:\licence.txt; DestName: licence.txt; DestDir: {app}; Languages: fr
Source: C:\license.txt; DestName: license.txt; DestDir: {app}; Languages: en
Source: lisezmoi.txt; DestDir: {app}; Flags: promptifolder isreadme; Languages: fr
Source: readme.txt; DestDir: {app}; Flags: promptifolder isreadme; Languages: en
```


Si vous êtes amené à créer des pages personnalisées, pour l'utilisation des {cm:...} dans la section **[Code]**, je vous engage à lire  **La section [Code] de InnoSetup**


V - Conclusions

A chacun ses préférences, mais moi, je suis un inconditionnel de ce logiciel, simple, facile de prise en main, peu gourmand en ressources, et aux possibilités et options immenses.

Pour les adeptes de la langue de Shakespeare, je vous engage à lire l'aide de **InnoSetup**, pour toutes informations complémentaires, principalement sur les **Flags**, qu'il me serait difficile de décrire ici, tant les possibilités sont nombreuses.

V-1 - Liens

Télécharger la dernière version de  **Inno-Setup**

Télécharger la dernière version de  **IsTools** (le site est en anglais, mais le logiciel propose un package langue en Français)

Dans la même série, voir aussi :  **La section [Code] de InnoSetup**

V-2 - Remerciements

Un tout grand **MERCI** à ceux qui ont pris quelques minutes de leur temps précieux pour me relire et me conseiller, et tout particulièrement à **bbil** et **sjrd**.