

# Mémoire Inno Setup

# Sommaire

1	Généralités.....	6
1.1	Avertissement.....	6
1.2	Conventions.....	6
1.3	Présentation.....	6
1.4	Rôles de chacun des trois composants principaux d'Inno Setup.....	7
1.4.1	Inno Setup.....	7
1.4.1.1	Le compilateur en mode « ligne de commande »: ISCC.exe.....	7
1.4.1.2	L'IDE en anglais d'Inno Setup: Compil32.exe.....	8
1.4.2	L'assistant ISTool.....	8
1.4.3	Le préprocesseur (ISPP).....	8
1.5	Installation d'Inno Setup.....	9
1.5.1	Téléchargement d'Inno Setup.....	9
1.5.2	Installation proprement dite d'Inno Setup.....	9
1.5.3	Configuration.....	17
1.5.3.1	Configuration d'ISTool.....	17
1.5.3.2	Configuration d'Inno Setup.....	18
1.5.4	Détail des options d'Inno Setup.....	21
1.5.5	Désinstallation d'Inno Setup.....	22
1.5.6	Inno Setup light, sans installation.....	24
2	Déroulement d'une installation typique.....	26
3	Le script Inno Setup.....	35
3.1	Présentation du script Inno Setup.....	35
3.2	Principales sections.....	37
3.2.1	[Setup].....	37
3.2.2	[Types].....	37
3.2.3	[Components] et [Tasks].....	37
3.2.4	[Dirs].....	37
3.2.5	[Files].....	37
3.2.6	[Icons].....	37
3.2.7	[INI].....	38
3.2.8	[InstallDelete].....	38
3.2.9	[Languages].....	38
3.2.10	[Messages].....	38
3.2.11	[CustomMessages].....	38
3.2.12	[LangOptions].....	38
3.2.13	[Registry].....	38
3.2.14	[Run].....	38
3.2.15	[UninstallDelete].....	38
3.2.16	[UninstallRun].....	38
3.2.17	Autres sections.....	38
3.3	Détail des sections du script.....	39
3.3.1	Section [Setup].....	39
3.3.2	Section [Types].....	46
3.3.3	Section [Components].....	46
3.3.4	Section [Tasks].....	47
3.3.5	Section [Files].....	47
3.3.6	Section [Icons].....	48

3.3.7	Section [Registry].....	48
3.3.8	Section [Run].....	48
3.3.9	Section [Code].....	50
3.4	Identification du setup.....	50
3.4.1	Nom du setup.....	50
3.4.2	Onglet de version.....	51
3.4.2.1	VersionInfoVersion.....	51
3.4.2.2	VersionInfoDescription.....	52
3.4.2.3	VersionInfoCopyright.....	53
3.4.2.4	VersionInfoCompany.....	53
3.4.2.5	VersionInfoTextVersion.....	54
3.4.3	Panneau de contrôle / Ajout-Suppression de programmes.....	54
3.4.3.1	UninstallDisplayIcon.....	54
3.4.3.2	UninstallDisplayName.....	55
3.4.3.3	AppPublisher + AppPublisherURL.....	56
3.4.3.4	AppVersion.....	56
3.4.3.5	AppContact.....	57
3.4.3.6	AppSupportURL.....	57
3.4.3.7	AppSupportPhone.....	57
3.4.3.8	AppReadmeFile.....	58
3.4.3.9	AppUpdatesURL.....	58
3.4.3.10	AppComments.....	58
3.5	Valeurs par défaut cascadées.....	58
3.6	{Constantes}.....	60
4	Le préprocesseur.....	64
4.1	Généralités.....	64
4.2	Commentaires.....	64
4.3	Directive #define.....	64
4.4	Pseudo-fonctions de manipulation des fichiers .ini.....	65
4.5	Ambiguïté dans la section [Code].....	66
4.6	Effet secondaire du préprocesseur.....	66
5	ISTool.....	67
6	Versions de Windows.....	68
7	Pascal.....	69
7.1	Différences C / Pascal.....	69
7.2	Expressions booléennes court-circuitées.....	71
7.3	Création de pages personnalisées.....	73
7.3.1	Page de choix.....	74
7.3.1.1	Création de la page.....	74
7.3.1.2	Paramètres de création.....	75
7.3.1.3	Exploitation des choix faits.....	77
7.3.1.4	Aspects.....	77
7.3.1.4.1	Choix exclusif, sans ListBox.....	78
7.3.1.4.2	Choix exclusif, avec ListBox.....	78
7.3.1.4.3	Choix non-exclusif, sans ListBox.....	78
7.3.1.4.4	Choix non-exclusif, avec ListBox.....	79
7.3.1.4.5	Choix non-exclusif, avec ListBox, sans ascenseur.....	79
7.3.1.5	Hiérarchie de la classe TInputOptionWizardPage.....	79
7.3.2	Page de saisie.....	80
8	Installations non-assistées.....	81

8.1	Installation non-assistée Inno Setup.....	81
9	Fichiers .rtf.....	82
9.1	Généralités.....	82
9.2	Mise en page.....	82
9.3	Limitations.....	83
10	Mode « ligne de commande ».....	84
10.1	Compilation non-assistée du setup.....	84
10.2	Installation non-assistée d'un setup.....	84
10.2.1	Inno Setup.....	85
10.2.2	Microsoft Windows Installer (MSI).....	86
10.2.3	InstallShield.....	87
10.2.4	NSIS.....	88
10.2.5	Wise Installer.....	89
10.2.6	Windows Hotfix.....	89
10.2.7	Setups exclusivement interactifs.....	89
11	Le registre Windows.....	90
11.1	Les moyens d'écriture.....	90
11.1.1	Section [Registry] d'Inno Setup.....	90
11.1.2	Commande externe reg.exe.....	90
11.1.3	Commande externe regedit.exe.....	90
11.1.4	Pascal.....	91
11.2	Quelques clefs intéressantes.....	91
11.2.1	Uninstall.....	91
11.2.2	RunOnce.....	92
12	Trucs et astuces.....	93
13	Annexes.....	95
13.1	Effet de AlwaysShowComponentsList.....	95
13.2	Effet de DisableReadyMemo.....	95
13.3	Setups splittés.....	95
13.4	Compression.....	96
13.5	Principes à respecter.....	96
13.6	Check-lists.....	97
13.7	Références.....	97
13.8	Glossaire.....	101

## Historique

<i>Version</i>	<i>Description</i>	<i>§<sup>1</sup></i>	<i>Page</i>
2008-04-20	Première ébauche du document.	Tous	Toutes
2008-11-24	Corrections mineures (fautes d'orthographe, numéros de pages manquants,...).		
	Ajout de conventions de polices de caractères.	1.2	6
	Précisions.	1.4.2, 1.5.6, 3.2.9, 3.2.17, 3.3.1, 3.3.4, 3.4.2.1, 3.4.2.2, 3.4.3.4, 3.4.3.5, 3.4.3.10, 2, 7, 3.6, 2, 3.3.2, 3.6, 7.3, 3.1, 3.3.8	8, 24, 38, 38, 40, 47, 51, 52, 56, 57, 58, 33, 69, 63, 31, 46, 60, 73, 35, 50
	Ajout de la correspondance Option / Registre.	1.5.3.2	19
	Ajout du paragraphe « Pseudo-fonctions de manipulation des fichiers .ini ».	4.4	65
	Inversion des paragraphes « » et « Création de pages personnalisées ».	7	69
	Création du paragraphe « Expressions booléennes court-circuitées ».	7.2	71
	Découpage du chapitre « Fichiers .rtf » en sous-chapitres. Ajout du paragraphe « Limitations ».	9	82
	Identification de version d'IS.	1.3	7
	Création du paragraphe « Le registre Windows ».	11	90
	Création du paragraphe « Trucs et astuces ».	12	93
	Création du paragraphe « Compression ».	13.4	96
	Création du paragraphe « Setups splittés ».	13.3	95
	Correction: la valeur par défaut de « DisableStartupPrompt » a pour effet de ne <b>pas</b> afficher la boîte de dialogue.	3.3.1	41
Création du paragraphe « Check-lists ».	13.6	97	
Création du paragraphe « Commentaires ».	4.2	64	

<sup>1</sup> Les paragraphes et pages concernés sont ceux de la présente version du document.

# 1 Généralités

## 1.1 Avertissement

Ce document n'a pas la vocation d'être un cours exhaustif sur Inno Setup. C'est simplement un aide-mémoire destiné avant tout à son auteur, reflet de son expérience et de ses besoins (industriels).

Certains aspects *pointus* ne sont volontairement pas abordés. D'autres aspects, tout aussi *pointus*, sont développés car ils correspondaient à un besoin précis de l'auteur.

Ce document est prévu pour être consulté sous sa forme informatique, plutôt que d'être imprimé (prévoir dans ce cas une cartouche *couleurs*): il contient de nombreuses copies d'écran, et est tissé de liens de type hypertexte, soit internes, soit externes vers l'aide en ligne d'Inno Setup<sup>1</sup>.

## 1.2 Conventions

Les entités informatiques sont transcrites avec une police `Courrier`.

Les documents de référence sont notés [DOCREF], en majuscules, à différencier des noms de sections Inno Setup notées [Section], en police `Courrier`.

Les hyperliens *cliquables* sont dans les styles suivants:

- [Vers Internet](#) 
- [Vers un autre paragraphe de ce document](#)

Les numéros de pages et de paragraphes ne sont pas identifiés par un style particulier, mais ils sont tous *cliquables*.

## 1.3 Présentation

Inno Setup (parfois abrégé « IS » dans la suite de ce document) est un outil de création de setups, c'est-à-dire de programmes qui installent d'autres programmes et/ou données.

Il permet de créer des setups utilisables sur toutes plateformes Windows x86, depuis Windows 95 jusqu'à Vista inclus, sauf Windows CE.

Inno Setup étant une application libre, de nombreuses extensions ont été développées, en particulier un préprocesseur similaire à celui du langage « C », et un assistant (*wizard*).

Il possède également la possibilité de développer des scripts en Pascal pour des installations complexes. A l'origine, ce moteur de script était une extension, qui a depuis été totalement intégrée à Inno Setup.

Il existe des concurrents, libres et non-libres, à Inno Setup, principalement:

- Microsoft Windows Installer (MSI)
- InstallShield
- NSIS
- Wise Installer

---

<sup>1</sup> Le fichier d'aide `ISetup.chm` installé avec Inno Setup a été décompilé avec la commande:  
`hh.exe -decompile ISetup_chm ISetup.chm`

☞ Inno Setup est abrégé IS, ce qui est également le cas d'InstallShield. De plus, ces deux programmes utilisent des fichiers scripts avec un suffixe « .iss »; ces deux types de scripts n'ont pas du tout la même structure.

Ce document s'applique à:

<i>Nom</i>	<i>Auteur principal</i>	<i>Setup</i>	<i>Site</i>	<i>Version</i>	<i>Description sommaire</i>
Inno Setup (ISCC)	Jordan Russell	ispack-5.2.2.exe	<a href="http://www.jrsoftware.org/isdl.php">http://www.jrsoftware.org/isdl.php</a>	5.2.2	Compilateur
Inno Setup Preprocessor (ISPP)	Alex Yackimoff	ispack-5.2.2.exe	<a href="http://www.jrsoftware.org/isdl.php">http://www.jrsoftware.org/isdl.php</a>	5.2.1	Préprocesseur
ISTool	Bjørnar Henden	istool-5.2.1.exe	<a href="http://www.istool.org/">http://www.istool.org/</a>	5.2.1	Assistant ( <i>wizard</i> )

Les versions sont identifiables comme suit:

- Inno Setup et Inno Setup Preprocessor:
  - Boîte de dialogue « About » de l'exécutable principal (Compil32.exe)
  - Boîte de dialogue « A propos » d'ISTool
- ISTool:
  - Onglet de version de l'exécutable ISTool.exe
  - Boîte de dialogue « A propos » d'ISTool
- ISCC ( compilateur en mode ligne de commande )
  - non identifiable

Ces trois outils ont été créés par trois auteurs différents, ce qui explique l'hétérogénéité des styles des documents associés.

## **1.4 Rôles de chacun des trois composants principaux d'Inno Setup**

### **1.4.1 Inno Setup**

Inno Setup lui même comprend deux éléments principaux:

#### **1.4.1.1 Le compilateur en mode « ligne de commande »: ISCC.exe.**

Il prend ses directives dans un unique fichier texte doté d'un suffixe « .iss »: le script Inno Setup. Le résultat de la compilation est un fichier `setup.exe`.

Le compilateur peut être utilisé en mode « ligne de commande » (DOS), ou bien plus généralement être appelé depuis l'IHM principale d'Inno Setup ou d'ISTool.

#### 1.4.1.2 L'IDE en anglais d'Inno Setup: *Compi132.exe*.

C'est essentiellement un éditeur syntaxique pour le script « .iss ». Cette IHM permet également de compiler le script, et de dérouler le setup en mode pas-à-pas.

Un assistant natif est également intégré: il permet de créer une trame de script.

L'IDE comporte une **aide en ligne très complète**. Cette aide en ligne est la référence absolue et définitive. Le présent document ne prétend pas être une traduction de cette aide : il ne rentre pas dans le détail et comprend de nombreux liens vers l'aide en ligne d'Inno Setup<sup>1</sup>.

#### 1.4.2 L'assistant ISTool.

Cet assistant ne permet pas la création d'une trame de script, mais est cependant beaucoup plus élaboré que l'assistant natif d'Inno Setup<sup>2</sup>: il permet de renseigner les différentes sections du script, à partir de boîtes de dialogues (case à cocher, listes déroulantes,...) où tous les paramètres sont décrits dans la langue sélectionnée lors de l'installation d'ISTool, le français étant disponible. La langue dans laquelle s'exprime l'assistant est indépendante de la langue du setup créé. Un assistant en français peut parfaitement aider à produire un setup en anglais, ou même multilingue.

ISTool n'apporte aucune aide au développement de code en Pascal.

L'IHM d'ISTool se substitue à l'IHM native d'Inno Setup: elle permet l'édition syntaxique du script, sa compilation, son exécution, mais ne permet pas de l'exécuter en pas-à-pas.

**ISTool ne fonctionne pas sous Windows d'une version antérieure à Windows 2000.**

Autre contre-indication d'ISTool: si pour des raisons d'organisation du projet, le script « .iss » contient volontairement plusieurs sections de mêmes noms, ISTool fond alors ces sections en une seule.

ISTool prend également des initiatives malheureuses avec les lignes qui ont été repliées avec le caractère « \ » (voir le détail au chapitre « [Effet secondaire du préprocesseur](#) », page 66).

Contrairement à l'éditeur d'Inno Setup, l'éditeur intégré à ISTool ne permet pas l'aide contextuelle, c-à-d l'appel de la page décrivant le mot sous le curseur par pression sur la touche F1.

ISTool est particulièrement appréciable lorsqu'on débute avec Inno Setup, et aussi lorsqu'on crée un setup ex-nihilo. Par contre, on ne doit pas l'utiliser pour déverminer le script en l'exécutant pas-à-pas, ni si la mise en page du script est *un peu spéciale*.

Par principe, ISTool n'a pas besoin d'une description détaillée.

#### 1.4.3 Le préprocesseur (ISPP).

Son rôle est assez similaire au préprocesseur du langage « C », mais bien que leurs syntaxes aient des airs de famille (directives `#define`, `#if`,...) il s'agit d'un préprocesseur différent, dont il faut connaître les particularités.

Le **préprocesseur** permet de créer des pseudo-fonctions exécutées lors de la phase de **compilation** (à opposer aux fonctions **Pascal**, qui ne sont utilisées que lors de l'**exécution** du setup).

<sup>1</sup> Sur le Web.

<sup>2</sup> ISTool peut être configuré pour lancer l'assistant natif d'Inno Setup lors de la création d'un nouveau script.

L'aide en ligne du préprocesseur est malheureusement de qualité inférieure à celle d'Inno Setup lui-même.

Le préprocesseur n'est pas un exécutable distinct: il est embarqué dans une version particulière du compilateur `ISCC.exe`.

## 1.5 Installation d'Inno Setup

L'installation d'Inno Setup et ISTool est sans danger pour l'OS hôte: rien n'est installé dans l'arborescence `\WINDOWS`, et ils n'écrivent dans le registre que dans des clefs anodines.

Le setup d'installation **standard** de l'ensemble « Inno Setup + préprocesseur + ISTool » télécharge ISTool au moment de l'exécution du setup, et détruit ce qui a été téléchargé une fois l'installation terminée.

L'installation décrite ci-dessous s'écarte de cette procédure standard en téléchargeant et installant explicitement ISTool, ceci afin de permettre une installation à partir d'une machine n'ayant pas d'accès Internet, et aussi de pouvoir installer une version qui n'est plus disponible sur le Web.

### 1.5.1 Téléchargement d'Inno Setup

Sur la page de téléchargement <http://www.jrsoftware.org/isdl.php>, télécharger le « QuickStart Pack » `ispack-5.x.y.exe`. Cet exécutable comprend Inno Setup et le préprocesseur.

Sur la page <http://www.istool.org/>, cliquer « Downloads », qui pointe vers [http://sourceforge.net/project/showfiles.php?group\\_id=206943](http://sourceforge.net/project/showfiles.php?group_id=206943), et télécharger ISTool, `release-5.x.y1`.

L'installation d'Inno Setup requiert d'avoir les droits d'Administrateur.

### 1.5.2 Installation proprement dite d'Inno Setup

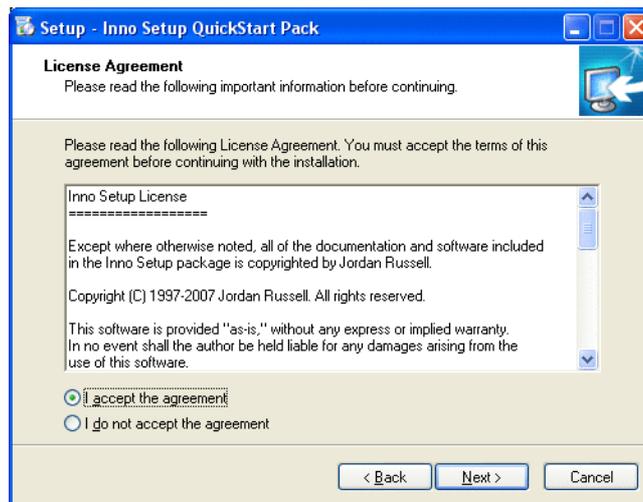
Exécuter `ispack-5.x.y.exe` sous le niveau d'un compte ayant les droits administrateur, et répondre aux questions selon les copies d'écrans ci-dessous:

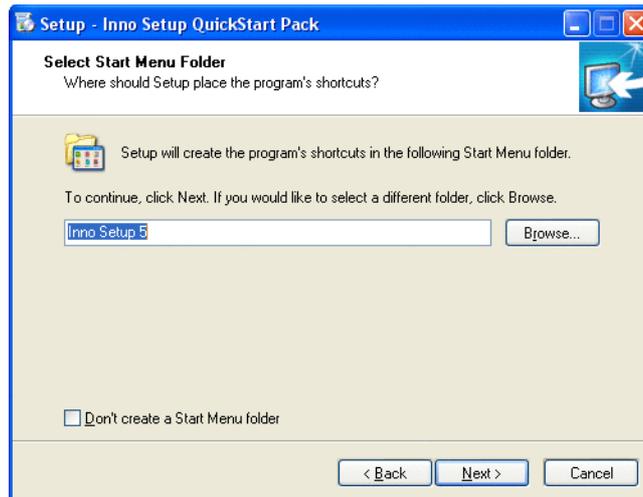


---

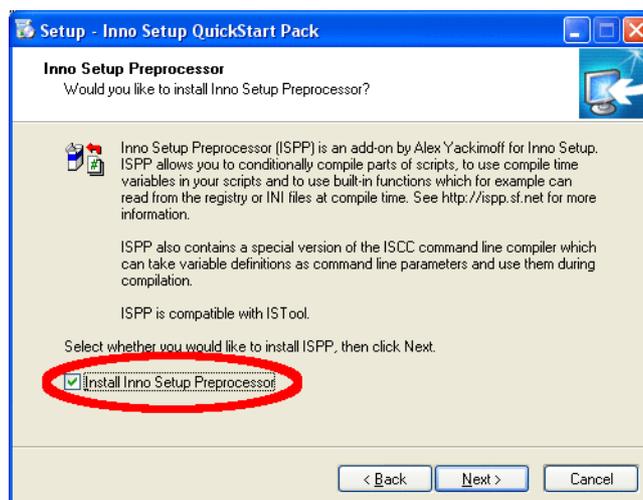
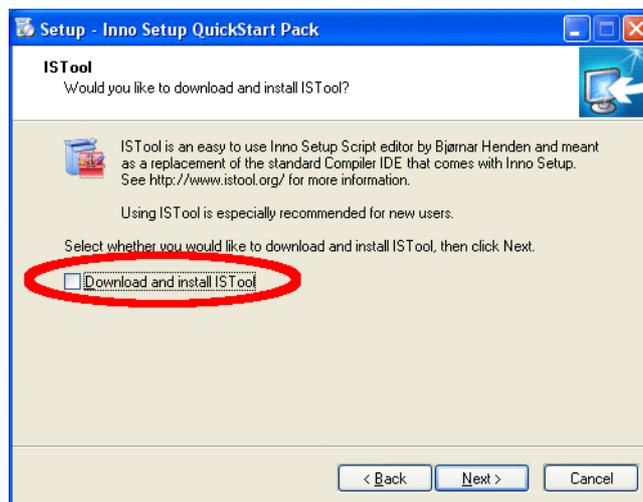
<sup>1</sup> Il peut y avoir un léger retard de version de ISTool sur Inno Setup, sans conséquences dramatiques.

La licence n'est pas très restrictive; en particulier « [...] *Permission is granted to anyone to use this software for any purpose, including commercial applications* [...] ».

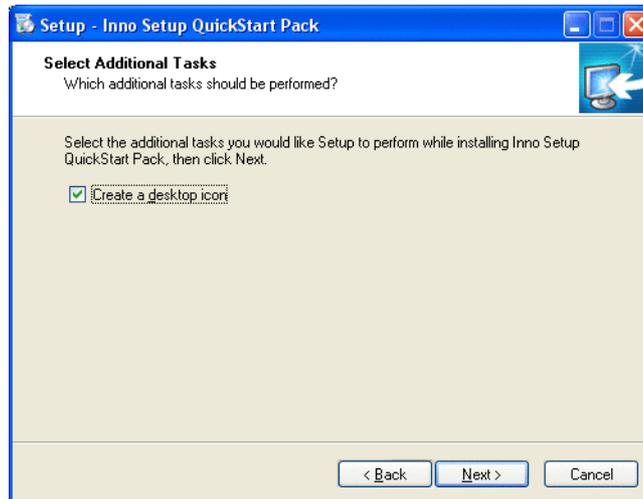
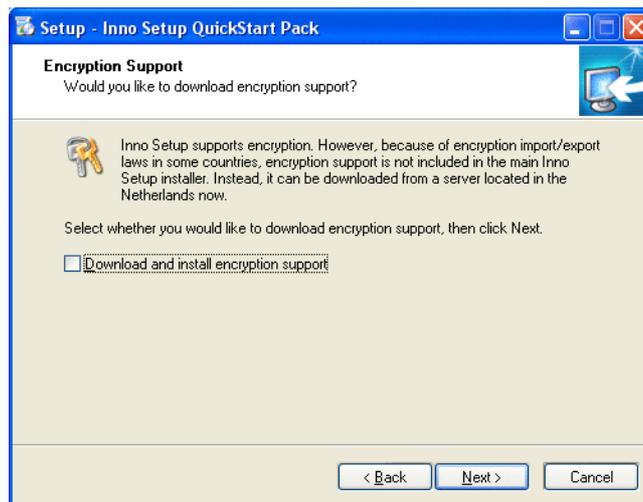




Ne pas sélectionner ISTool. Celui-ci a été téléchargé explicitement, et sera installé séparément.

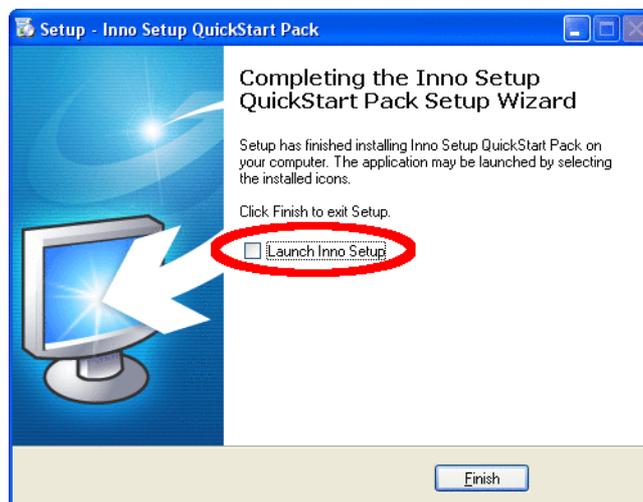


L'extension de chiffrement n'est pas indispensable (le cas échéant, elle est également téléchargée temporairement lors de l'exécution de ce setup).



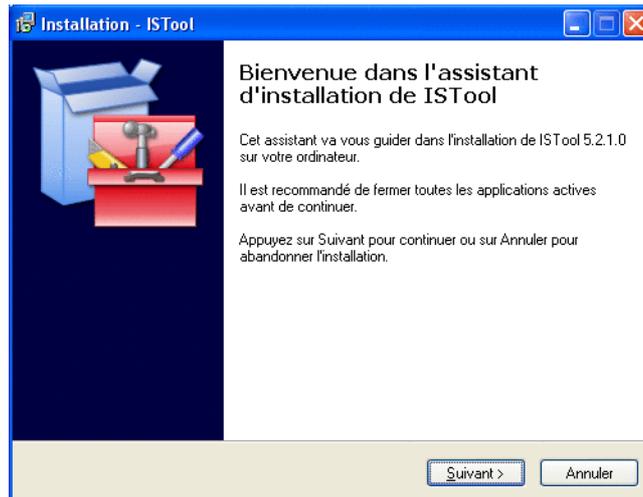


Il est inutile de lancer immédiatement Inno Setup: l'installation n'est pas terminée.



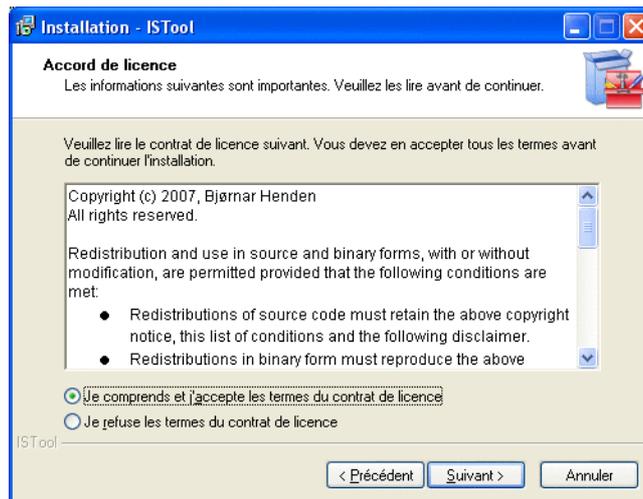
Exécuter `istool-5.x.y.exe`, et sélectionner la langue dans laquelle le setup va se poursuivre. Cette langue est également celle qui sera utilisée par ISTool pour donner ses indications, sous le niveau de l'utilisateur qui procède à l'installation (administrateur).

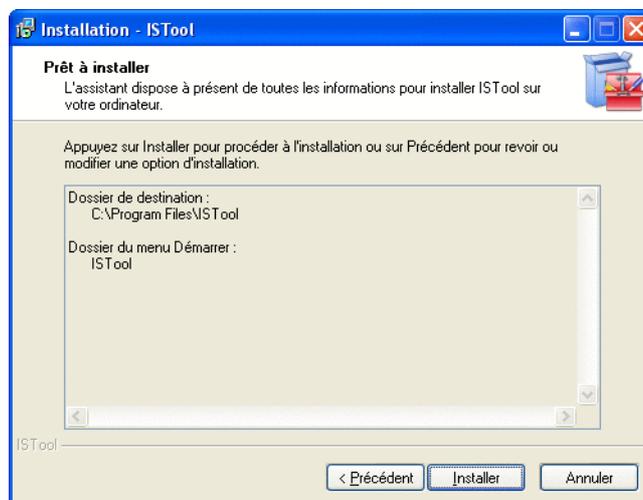
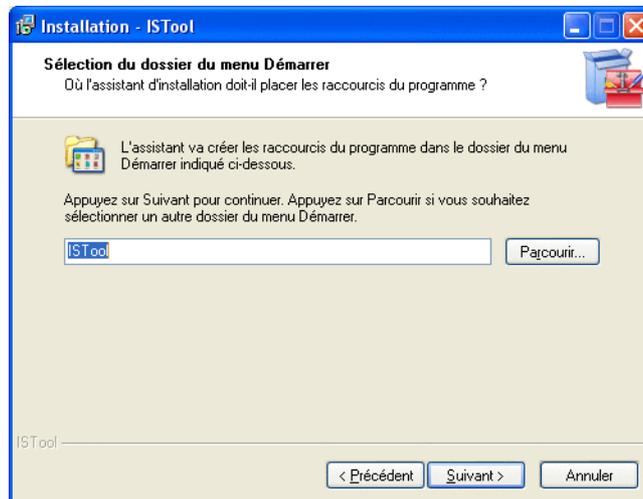
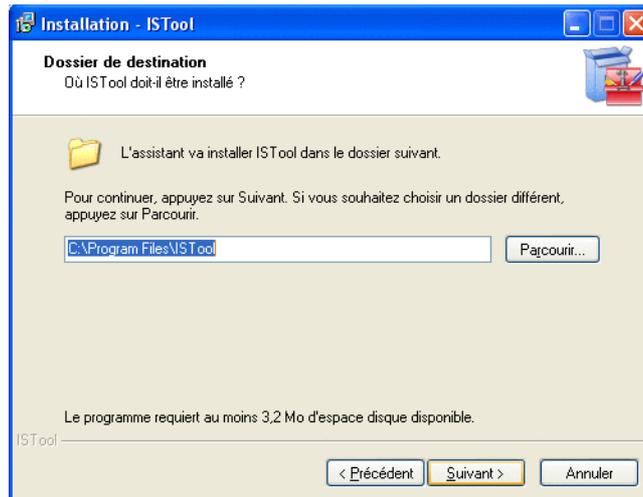


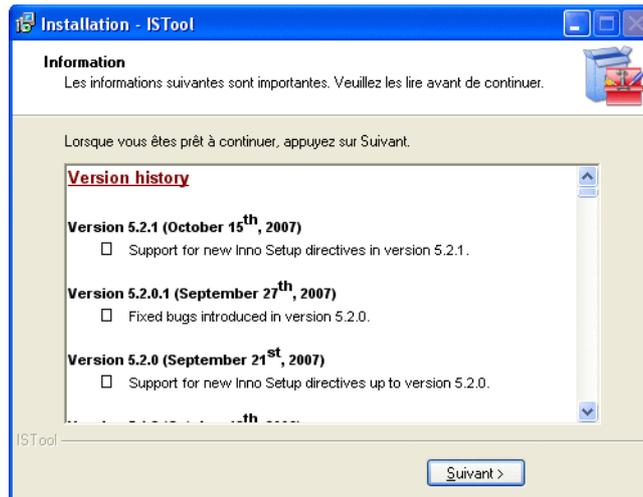


La licence ne pose pas non plus de problème, tant que l'on distribue `istool-5.x.y.exe` sans modification.

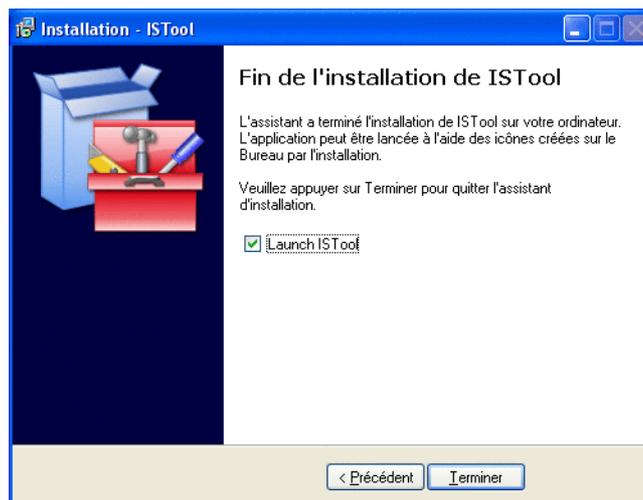
Aucune restriction n'est mentionnée concernant l'utilisation du logiciel.







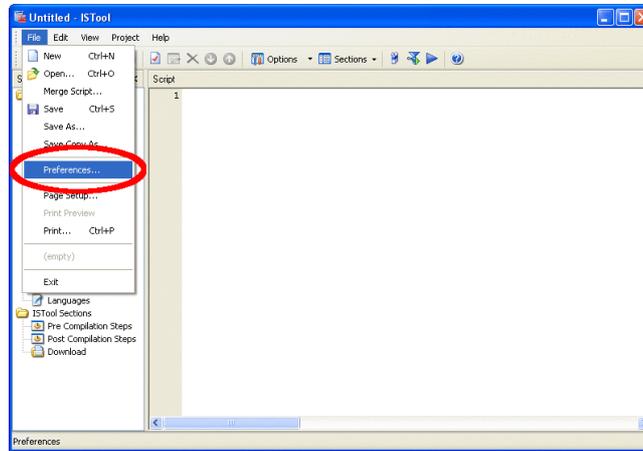
Si Inno Setup est destiné à être utilisé sous le compte qui servi à l'installation (avec les droit de l'administrateur), alors accepter de lancer ISTool, sinon se délogger et se relogger sous le compte utilisateur adéquat et lancer ISTool par le menu « Démarrer » de Windows.



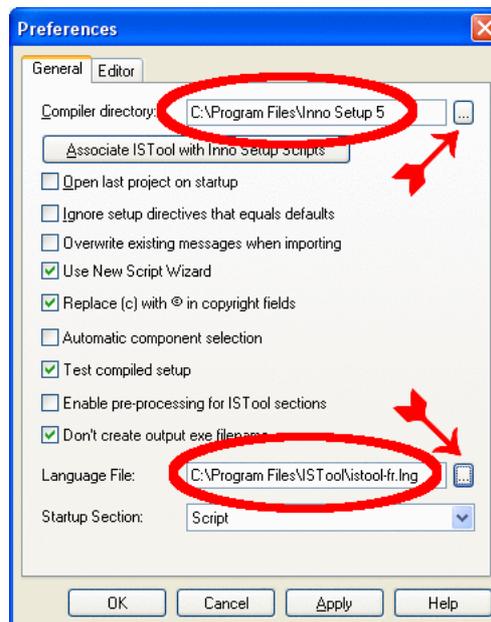
## 1.5.3 Configuration

### 1.5.3.1 Configuration d'ISTool

Ouvrir les préférences d'ISTool (a priori, l'IHM est en anglais).



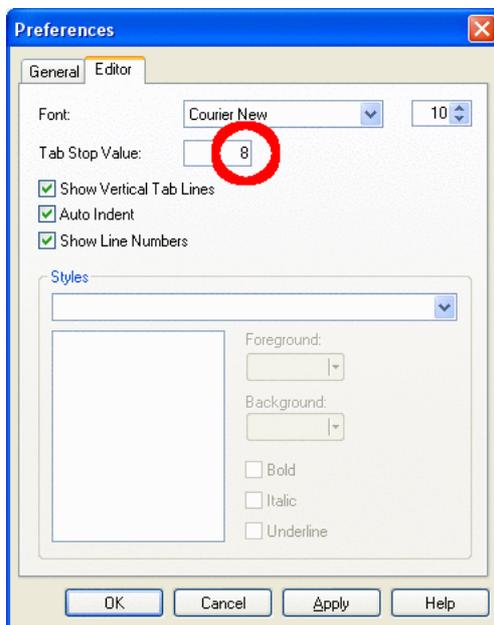
Renseigner le chemin vers Inno Setup (a priori: « C:\Program Files\Inno Setup 5 »).  
Renseigner également le chemin du fichier de personnalisation de la langue de l'IHM d'ISTool.



Configurer la position des tabulations.

L'important est d'avoir une valeur identique pour ISTool (valeur par défaut: 4), et pour Inno Setup (valeur par défaut: 2; voir page 19).

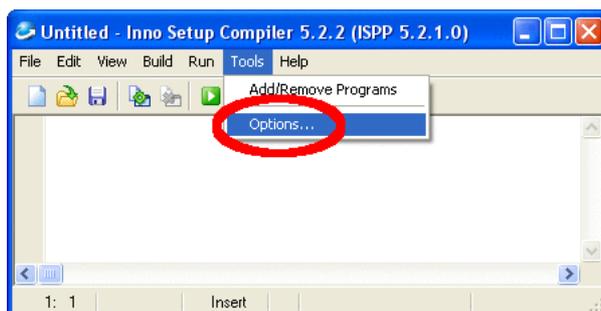
La valeur 8 est très large, mais c'est celle qui pose le moins de problème d'édition et d'impression avec d'autres éditeurs.



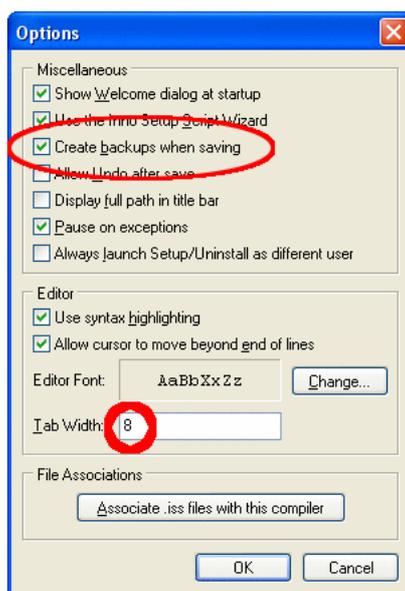
Fermer ISTool, et lancer Inno Setup Compiler.

### 1.5.3.2 Configuration d'Inno Setup

Après le lancement d'Inno Setup Compiler, refuser l'offre de création d'un nouveau script.



Donner la même largeur de tabulation sous Inno Setup, que sous ISTool (voir page 18).  
 Une option également intéressante à prendre, est la création automatique d'une copie de sauvegarde.



Ces options sont mémorisées dans le registre, sous la clef  
 HKEY\_CURRENT\_USER\Software\Jordan Russell\Inno Setup\Options. Toutes les valeurs  
 correspondant aux options sont de type « chaîne de caractères », y compris pour les options de type  
 « tout-ou-rien » (cases à cocher).

Correspondance entre les options, et les valeurs du registre:

<i>Option</i>	<i>Nom de la valeur</i>	<i>Domaine</i>	<i>Par défaut</i>
Show Welcome dialog at startup	ShowStartupForm	{ "0", "1" }	"1"
Use the Inno Setup Script Wizard	UseWizard	{ "0", "1" }	"1"
Create backups when saving	MakeBackups	{ "0", "1" }	"0"
Display full path in title bar	FullPathInTitleBar	{ "0", "1" }	"0"
Allow Undo after save	UndoAfterSave	{ "0", "1" }	"0"
Pause on exceptions	PauseOnDebuggerExceptions	{ "0", "1" }	"1"
Use syntax highlighting	UseSynHigh	{ "0", "1" }	"1"
Allow cursor to move beyond end of lines	EditorCursorPastEOL	{ "0", "1" }	"1"
Tab Width	TabWidth		"2"
Editor Font	EditorFontName		"Courier New"
	EditorFontSize		"10"
	EditorFontCharset		"1"
Always launch Setup/Uninstall as different user	RunAsDifferentUser	{ "0", "1" }	"0"

<i>Option</i>	<i>Nom de la valeur</i>	<i>Domaine</i>	<i>Par défaut</i>
View / Toolbar	ShowToolBar	{ "0", "1" }	"1"
View / Status Bar	ShowStatusBar	{ "0", "1" }	"1"
Build / Low Priority During Compile	LowPriorityDuringCompile	{ "0", "1" }	"0"

Il est ainsi possible de positionner les deux options surlignées dans la copie d'écran en page 19, avec le fichier .reg suivant:

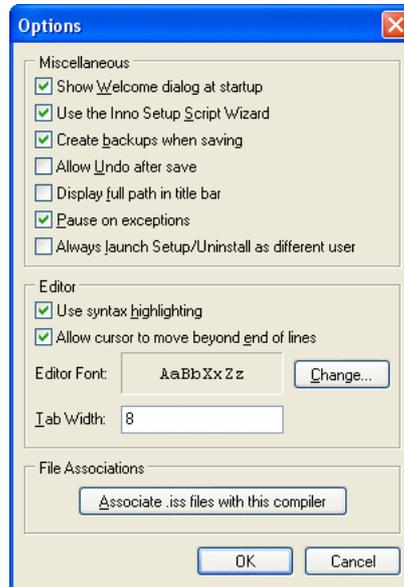
```
REGEDIT4
```

```
[HKEY_CURRENT_USER\Software\Jordan Russell\Inno Setup\Options]
"MakeBackups"="1"
"TabWidth"="8"
```

 La configuration d'Inno Setup et d'ISTool, **y compris le chemin d'Inno Setup**, est propre à un utilisateur.

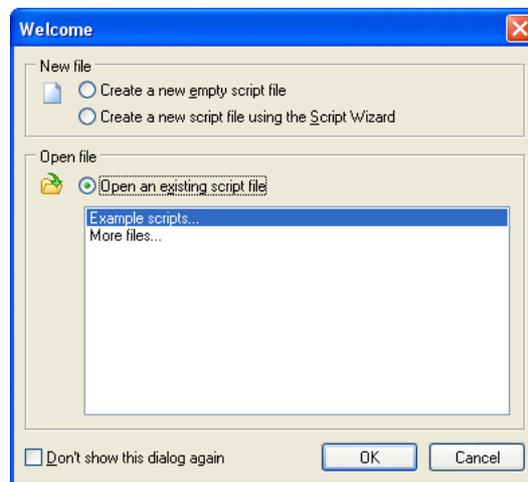
## 1.5.4 Détail des options d'Inno Setup

Rappel des options possibles d'Inno Setup:



*Show Welcome dialog at startup*

Lorsque cette option est présente, Inno Setup propose d'ouvrir ou de créer un fichier:



*Use the Inno Setup Script Wizard*

En cas de création d'un nouveau script, permet de créer une trame avec l'assistant natif d'Inno Setup.

*Create backups when saving*

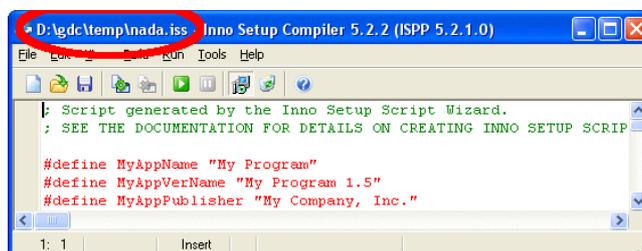
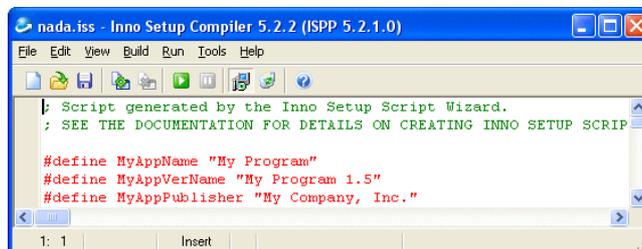
La version précédente du script est sauvegardée avec un suffixe « .~is ».

A noter, que cette option (comme les autres d'ailleurs) est locale à Inno Setup: ISTool ne

sauvegarde pas la version précédente du script même si cette option est cochée.

*Allow Undo after save*

*Display full path in title bar*



*Pause on exceptions*

*Always launch Setup/Uninstall as different user*

## 1.5.5 Désinstallation d'Inno Setup

S'il est nécessaire d'effacer complètement toute trace d'Inno Setup, c'est possible car Inno Setup n'installe rien dans l'arborescence \WINDOWS, et n'utilise que des entrées *anodines* du registre.

Seuls restent à nettoyer après désinstallation classique, les paramètres propres aux utilisateurs ayant lancé Inno Setup et/ou ISTool.



Les éventuels raccourcis créés manuellement vers Inno Setup ou ISTool se trouvent sous « C:\Documents and Settings\ », et ont un suffixe « .lnk ».

Il reste après ces opérations quelques références à « inno » et à « istool » dans le registre, mais celles-ci sont dans les clefs « MRU », « recent file », « cache »,... Ces références sont a priori temporaires, et devraient par principe disparaître d'elles-mêmes. Les exceptions notables sont les logiciels installés avec Inno Setup qui laissent des traces qu'il ne faut **pas toucher**, sous la clef: HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall.

### 1.5.6 Inno Setup *light*, sans installation

Il peut être nécessaire d'avoir à compiler un script Inno Setup sur une machine sur laquelle Inno Setup n'est pas installé, par exemple dans le cadre d'une chaîne de compilation automatisée.

Heureusement, Inno Setup est auto-suffisant.

La version minimum est non-interactive: elle ne comprend donc pas ISTool. Son exécution ne laisse pas de traces indésirables dans le registre, ni dans les fichiers.

Jordan liste les fichiers à distribuer ([REDIS], page 100): cette liste, établie en 2005, ne prend pas en compte le préprocesseur.

Depuis une machine sur laquelle Inno Setup a été installé de manière conventionnelle, il suffit de récupérer les fichiers suivants:

Fichier	Description
ISCC.exe	Interface en mode « ligne de commande » pour le compilateur.
ISCMplr.dll	Le compilateur proprement dit.
ISCMplr.dls	Semble être une DLL complémentaire du compilateur, avec préprocesseur.
Builtins.iss	Prédéfinitions pour le préprocesseur.
Default.isl	Messages du setup, en anglais.
SetupLdr.e32	Images binaires embarquées dans le setup.
Setup.e32	
islzma.dll	Compresseur/Décompresseur LZMA. Indispensable dans tous les cas, car utilisé en interne.
WizModernImage.bmp	Images par défaut.
WizModernSmallImage.bmp	

A ces fichiers indispensables, il faut éventuellement ajouter:

Fichier	Description
Compil32.exe	Interface graphique pour le compilateur, alternative à ISCC.exe.
isbzip.dll	Compresseur + décompresseur bzip <sup>1</sup>
isbunzip.dll	
iszlib.dll	Compresseur + décompresseur zip
isunzlib.dll	
Languages/	Fichiers de langues <sup>2</sup> , autres que l'anglais.
WizModernImage-IS.bmp	Autres images que celles par défaut.
WizModernSmallImage-IS.bmp	

1 Nécessaire, si la compression adéquate est utilisée (paramètre Compression de la section [Setup], page 40). Selon Jordan, la compression lzma est généralement le meilleur choix.

2 A choisir selon les langues supportées par le setup cible.

Les fichiers doivent respecter la même hiérarchie que dans le répertoire d'installation original, c'est-à-dire que tous les fichiers doivent être dans le même répertoire, sauf les fichiers de langues autres que `Default.isl`, qui eux doivent être placés dans le sous-répertoire « `Languages/` ».

La version minimale occupe à peu près 1,5 MiB, et seulement 700 KiB après compression.

## 2 Déroulement d'une installation typique

L'exécution d'un setup compilé avec Inno Setup peut prendre 2 formes:

1. Setup interactif (cas habituel).
2. Setup non-assisté, en mode ligne de commande (options `/SILENT` et `/VERYSILENT`).

Cette section décrit les boîtes de dialogues affichées pendant le déroulement d'un setup interactif.

L'installation d'ISTool (pages [13](#) à [16](#)) est un exemple assez typique d'une installation par Inno Setup.

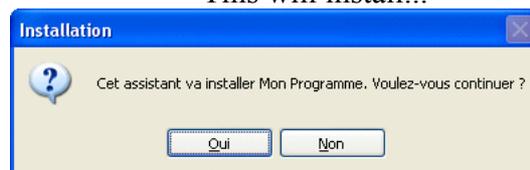
La séquence d'installation détaillée ci-dessous est plus étoffée: elle offre un échantillonnage assez complet de ce qu'il est possible de faire, sans avoir à recourir à des boîtes de dialogues spécialisées, et sans aucune ligne de code Pascal.

Référence dans l'aide Inno Setup: [Wizard Pages](#)

Dans la description de chaque boîte de dialogue, on mentionne les éléments du script concernés. La présentation du script est faite au chapitre [3.1](#), page [35](#).



Cet assistant va installer...  
This will install...



Conditionnée par `:DisableStartupPrompt`

Option ISTool:

Paramètre ligne de commande: `/SP-`, page [85](#).

Identificateur Pascal de la page: N/A

Commentaire:



Langue de l'assistant d'installation  
Select Language



Conditionnée par `:ShowLanguageDialog`

Section `[Languages]` avec aux moins deux langues définies.

Option ISTool:

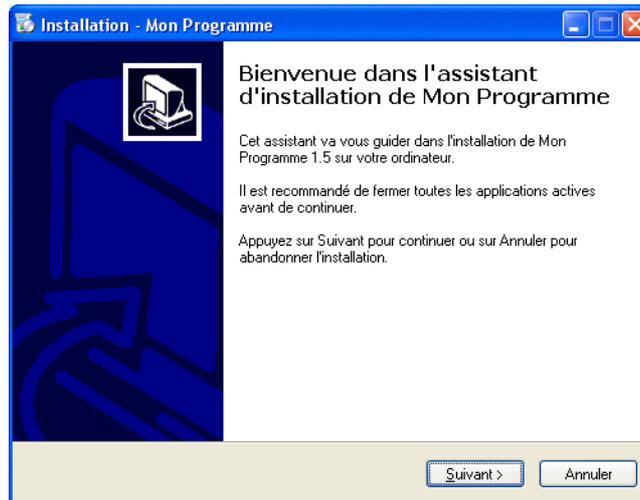
Paramètre ligne de commande:

Identificateur Pascal de la page:

Commentaire:



## Bienvenue Welcome



Conditionnée par :Néant: en mode interactif, cette boîte de dialogue est toujours affichée.

Option IStool:N/A

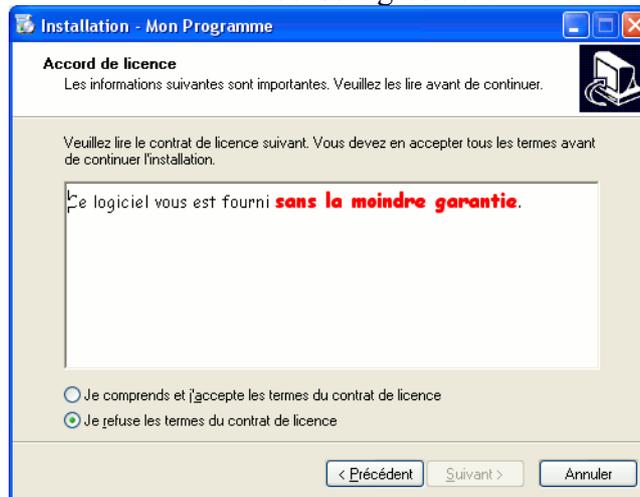
Paramètre ligne de commande:

Identificateur Pascal de la page:wpWelcome

Commentaire:



## Accord de licence License Agreement



Conditionnée par :LicenseFile

Option IStool:

Paramètre ligne de commande:

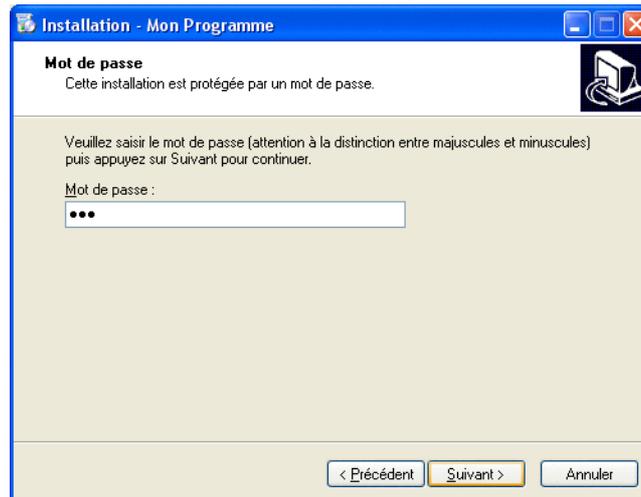
Identificateur Pascal de la page:wpLicense

Commentaire:Le bouton « Suivant » est validé quand l'utilisateur accepte la licence.

Rien n'oblige l'utilisateur à faire défiler la licence jusqu'à la fin. Ne pas oublier d'installer explicitement le fichier de licence dans la section [Files], sinon l'utilisateur n'aura pas d'autre moyen que de procéder à une nouvelle installation afin de consulter ses droits.



## Mot de passe Password



Conditionnée par :Password

Option ISTool:

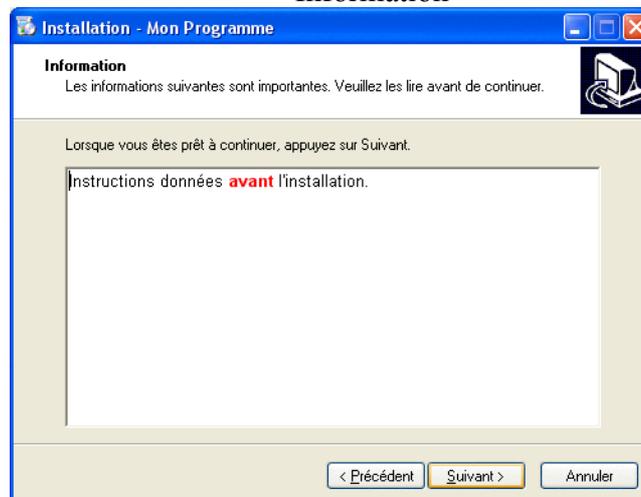
Paramètre ligne de commande:

Identificateur Pascal de la page:wpPassword

Commentaire:Pour une gestion un peu élaborée du mot de passe, il faut en passer par le cryptage et des fonctions Pascal.



## Information Information



Conditionnée par :InfoBeforeFile

Option ISTool:

Paramètre ligne de commande:

Identificateur Pascal de la page:wpInfoBefore

Commentaire:La page n'est affichée que si le champ InfoBeforeFile du script est renseigné avec le nom du fichier à afficher.



## Informations sur l'Utilisateur User Information

Installation - Mon Programme

**Informations sur l'Utilisateur**  
Veuillez saisir les informations qui vous concernent.

Nom d'utilisateur :  
Hugo Z. Hackenbush

Organisation :

< Précédent   Suivant >   Annuler

Conditionnée par :UserInfoPage

Option IStool:

Paramètre ligne de commande:

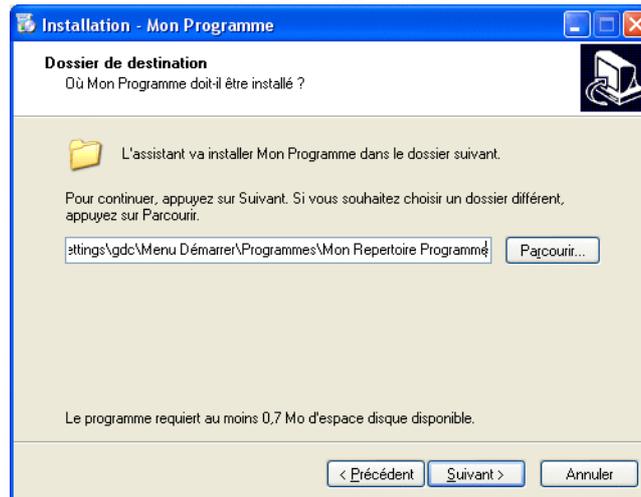
Identificateur Pascal de la page:wpUserInfo

Commentaire:Ces informations ne sont pas utilisées par le setup. Elles n'ont d'intérêt que si une routine Pascal les exploite.

Si une fonction Pascal nommée CheckSerial est définie, alors cette page affiche également un champ « Numéro de série », prérenseigné avec la valeur de DefaultUserInfoSerial.



## Dossier de destination Select Destination Location



Conditionnée par :DisableDirPage<sup>1</sup>

Option IStool:

Paramètre ligne de commande:

Identificateur Pascal de la page:wpSelectDir

Commentaire:Si cette boîte de dialogue n'est pas affichée, l'installation se fait dans le répertoire par défaut.

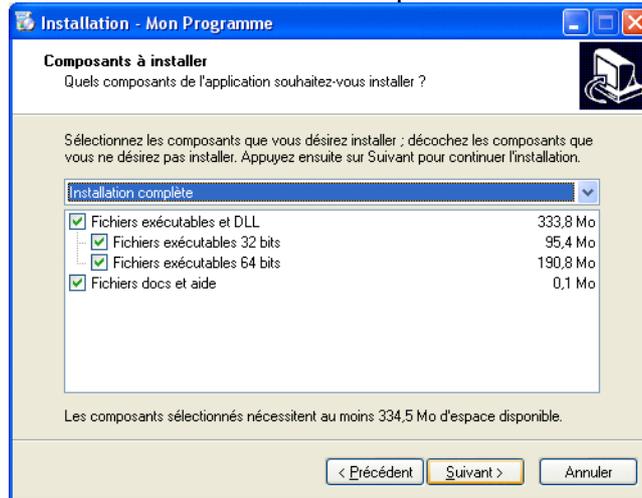
Le répertoire par défaut peut être donné relativement à des chemins système (par exemple: « Program Files »). Il peut aussi être donné par une entrée du registre, un paramètre dans la ligne de commande ou être déterminé par une routine Pascal.

---

<sup>1</sup> CreateAppDir conditionne également l'affichage de cette boîte de dialogue, mais le but est alors d'installer des fichiers directement dans le répertoire \WINDOWS.



## Composants à installer Select Components



Conditionnée par :Découpage du setup en composants.  
AlwaysShowComponentsList

Option IStool:

Paramètre ligne de commande:

Identificateur Pascal de la page:wpSelectComponents

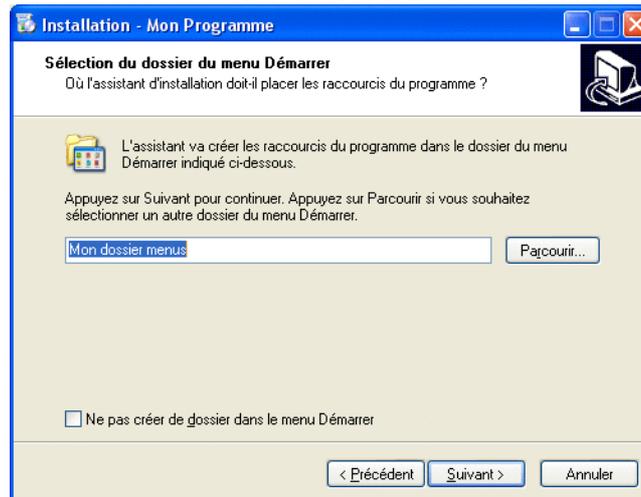
Commentaire:La liste des composants, avec les cases à cocher associées, n'est affichée que si un des types d'installation possède le flag « iscustom ».

Les composants peuvent être hiérarchisés.

Voir en annexe 13.1 page 95, l'effet de AlwaysShowComponentsList.



## Sélection du dossier du menu Démarrer Select Start Menu Folder



Conditionnée par : `DisableProgramGroupPage`

`AllowNoIcons` conditionne la présence de la case à cocher «Ne pas créer de dossier dans le menu Démarrer».

Option IStool:

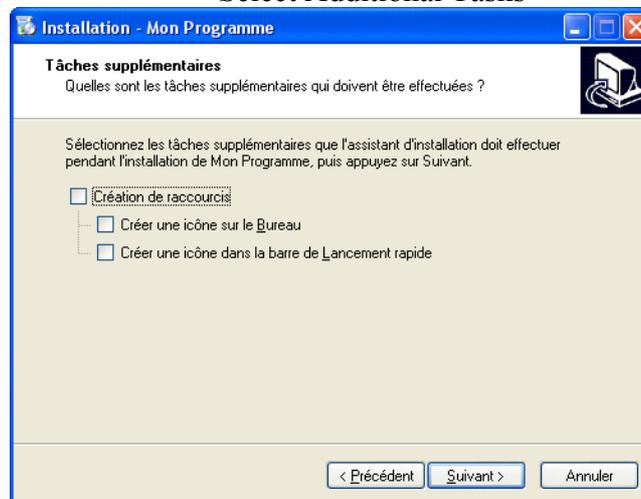
Paramètre ligne de commande:

Identificateur Pascal de la page: `wpSelectProgramGroup`

Commentaire: Si l'affichage de cette boîte de dialogue est inhibé, l'entrée du menu est quand même créée, avec sa valeur par défaut.



## Tâches supplémentaires Select Additional Tasks



Conditionnée par : Présence d'une section [Tasks].

Option IStool:

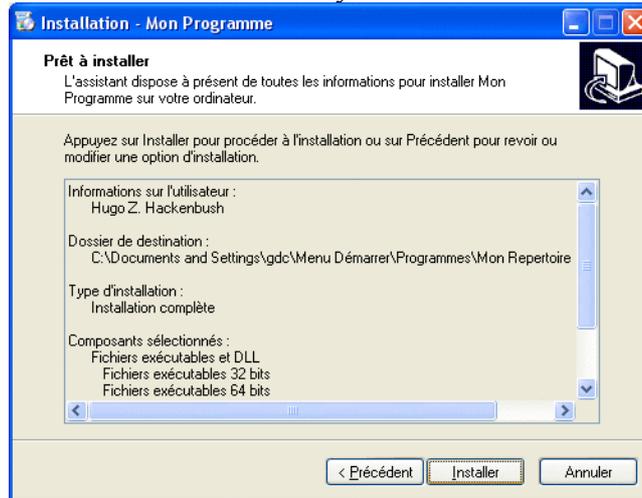
Paramètre ligne de commande:

Identificateur Pascal de la page: `wpSelectTasks`

Commentaire: De la même façon que pour les composants, les tâches peuvent être hiérarchisées.



## Prêt à installer Ready to Install



Conditionnée par :DisableReadyPage  
DisableReadyMemo

Option IStool:

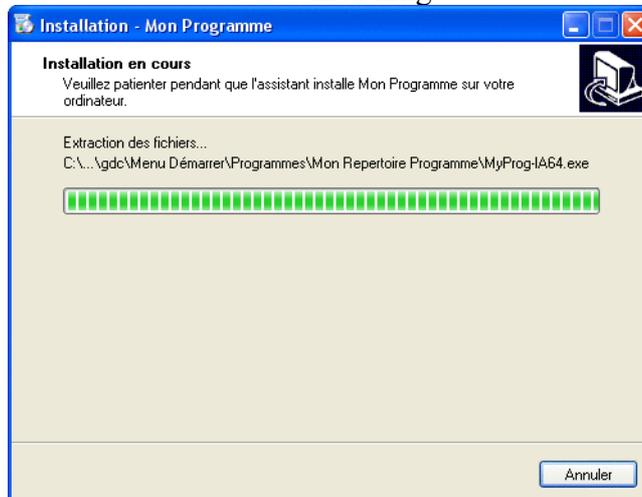
Paramètre ligne de commande:

Identificateur Pascal de la page:wpReady

Commentaire:DisableReadyMemo permet d'afficher la boîte de dialogue, mais sans le résumé des choix qui ont été faits. Voir en Annexe 13.2, page 95.



## Installation en cours Installing



Conditionnée par :N/A

Option IStool:

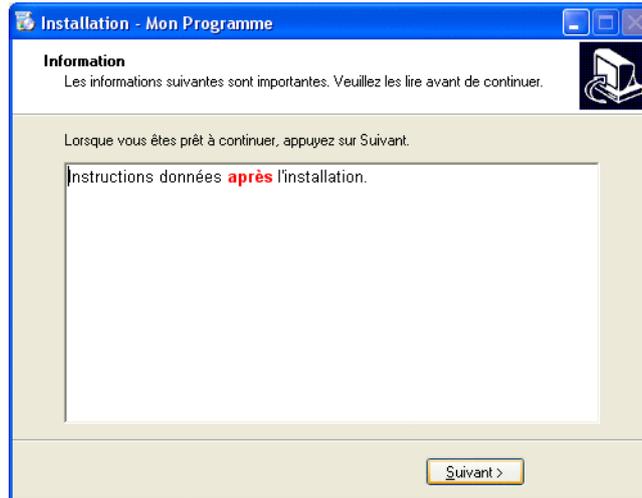
Paramètre ligne de commande:N/A

Identificateur Pascal de la page:wpInstalling

Commentaire:Il n'y a pas d'interaction possible avec cette boîte: elle sert juste à montrer le déroulement du setup.



## Information Information



Conditionnée par :InfoAfterFile

Option ISTool:

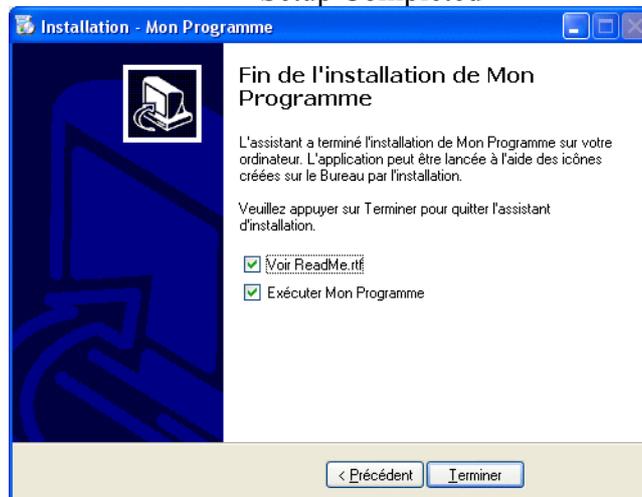
Paramètre ligne de commande:N/A

Identificateur Pascal de la page:wpInfoAfter

Commentaire:La page n'est affichée que si le champ InfoAfterFile du script est renseigné avec le nom du fichier à afficher.



## Fin de l'installation Setup Completed



Conditionnée par :DisableFinishedPage

Option ISTool:

Paramètre ligne de commande:

Identificateur Pascal de la page:wpFinished

Commentaire:Les cases à cocher sont les actions déclarées dans la section [Run] avec le flag postinstall.

## 3 Le script Inno Setup

### 3.1 Présentation du script Inno Setup

Le script est un fichier texte décrivant le setup. Il est composé de sections prédéfinies. Chaque section décrit une phase du setup, soit lors de la compilation, soit lors de l'installation, soit lors de la désinstallation.

L'installation d'ISTool (page [13](#) à page [16](#)) est un exemple assez typique d'une installation par Inno Setup.

Les délimiteurs de sections sont de la forme `[Section]`, mais cela n'en fait pas pour autant un fichier de structure « `.ini` ».

Les commentaires standard d'IS sont introduits par un caractère « `;` » en début de ligne, sauf dans la section `[Code]`, où ce type de commentaire est interdit et doit être remplacé par un commentaire dans le style Delphi; voir le détail au paragraphe 3.3.9, page 50.

Si le préprocesseur est installé, les commentaires « `//` » qui lui sont spécifiques sont reconnus dans la totalité du script, y compris dans la section `[Code]`. Voir la discussion sur une éventuelle contre-indication, au paragraphe 4.2, page 64.

Exemple de script:

; Script generated by the Inno Setup Script Wizard.

; SEE THE DOCUMENTATION FOR DETAILS ON CREATING INNO SETUP SCRIPT FILES!

[Setup]

← Délimiteur de section

AppName=My Program

AppVerName=My Program 1.5

AppPublisher=My Company, Inc.

AppPublisherURL=http://www.example.com/

AppSupportURL=http://www.example.com/

AppUpdatesURL=http://www.example.com/

DefaultDirName={pf}\My Program

DefaultGroupName=My Program

OutputBaseFilename=setup

Compression=lzma

SolidCompression=yes

[Languages]

Name: english; MessagesFile: compiler:Default.isl

[Tasks]

Name: desktopicon; Description: {cm:CreateDesktopIcon}; GroupDescription: {cm:AdditionalIcons}; Flags: unchecked

[Files]

Source: C:\Program Files\Inno Setup 5\Examples\MyProg.exe; DestDir: {app}; Flags: ignoreversion

; NOTE: Don't use "Flags: ignoreversion" on any shared system files

[Icons]

Name: {group}\My Program; Filename: {app}\MyProg.exe

Name: {commondesktop}\My Program; Filename: {app}\MyProg.exe; Tasks: desktopicon

[Run]

Filename: {app}\MyProg.exe; Description: {cm:LaunchProgram,My Program}; Flags: nowait postinstall skipifsilent

L'ordre des sections dans le fichier script n'est pas important. Plusieurs sections de même nom peuvent exister.

- ☞ ISTool peut regrouper en une seule section toutes les sections de même nom.  
Compil32 préserve le découpage en sections original du script.

## **3.2 Principales sections**

### **3.2.1 [Setup]**

Cette section contient les directives générales du setup, comme la version, le nom du setup, les boîtes de dialogues affichées ou non pendant l'exécution, le répertoire principal d'installation, les différents fichiers du genre licence ou LisezMoi qui seront montrés à l'utilisateur,...

La plupart des paramètres sous cette section possèdent des valeurs par défaut, mais quelques uns doivent être explicitement définis, ce qui rend cette section obligatoire.

### **3.2.2 [Types]**

Cette section facultative définit les types d'installation autorisés, typiquement « Installation complète », « Installation compacte » et « Installation personnalisée ».

Lors de l'exécution du setup, l'utilisateur choisit le type d'installation avec la boîte de dialogue « [Composants à installer](#) » montrée page [31](#). Le type d'installation conditionne les composants installés par défaut.

### **3.2.3 [Components] et [Tasks]**

Ces deux sections décrivent les composants qui peuvent être installés, et les tâches qui peuvent être effectuées lors de l'installation.

Ces sections sont facultatives, auquel cas le setup est monolithique.

### **3.2.4 [Dirs]**

### **3.2.5 [Files]**

Cette section est théoriquement facultative, mais comme c'est elle qui liste les fichiers à livrer, elle est pratiquement toujours nécessaire.

### **3.2.6 [Icons]**

Ici, on liste le cas échéant les raccourcis à créer, soit sur le bureau, soit dans des menus de démarrage.

### 3.2.7 [INI]

### 3.2.8 [InstallDelete]

### 3.2.9 [Languages]

On liste ici les langues parmi lesquelles l'utilisateur peut choisir pour procéder à l'installation.

La boîte de dialogue « [Langue de l'assistant d'installation](#) » (page 26) n'est affichée que si au moins deux langues ont été configurées dans cette section.

La langue choisie peut aussi être prise en compte dans d'autres sections pour installer sélectivement des fichiers, ou dans toute autre phase de l'exécution du setup.

Cette section est facultative: si elle n'existe pas, l'installation aura lieu en anglais.

Une installation multilingue n'est pas une affaire simple: Inno Setup peut afficher son IHM en de nombreuses langues, mais il faut aussi localiser les messages spécifiques ainsi que les descriptions de types d'installation, de composants, de tâches...

### 3.2.10 [Messages]

### 3.2.11 [CustomMessages]

### 3.2.12 [LangOptions]

### 3.2.13 [Registry]

Cette section est le moyen le plus simple pour écrire dans le registre Windows.

Les autres possibilités sont d'utiliser `regedit.exe` (ou `reg.exe`), ou bien d'utiliser un bout de code écrit en Pascal.

### 3.2.14 [Run]

Permet d'exécuter des programmes, **après** avoir installé les fichiers.

### 3.2.15 [UninstallDelete]

### 3.2.16 [UninstallRun]

### 3.2.17 Autres sections

D'autres sections peuvent exister, pour les besoins des extensions d'Inno Setup, mais ignorées par Inno Setup lui-même. Par exemple, ISTool utilise une section `[_ISTool]` pour ses besoins propres. En conséquence, Inno Setup ignore silencieusement les sections qu'il ne reconnaît pas; une section titrée `[File]` n'aura aucun effet, puisque Inno Setup attend une section nommée `[Files]`!

A l'inverse, dans une section reconnue, et structurée comme un `.ini` (`[Setup]`, `[LangOptions]`), il est interdit d'insérer des clefs inconnues d'Inno Setup.

### 3.3 Détail des sections du script

#### 3.3.1 Section [Setup]

C'est la section la plus riche, mais la plupart des paramètres sont facultatifs et possèdent une valeur par défaut raisonnable.

Certains paramètres ont des valeurs par défaut chaînées, c-à-d, si le paramètre `Xxx` n'est pas renseigné dans le script, il prendra comme valeur par défaut la valeur courante du paramètre `Yyy`, lui-même ayant une valeur par défaut prédéfinie.

Ne sont discutés ici que les paramètres obligatoires, leur nom est en caractères gras, ou ceux fonctionnellement très intéressants.

<i>Paramètre</i>	<i>Discussion</i>	<i>Par défaut</i>
<b>AppName</b>	Cette entrée est obligatoire. Elle donne le nom en clair de l'application, <b>sans mention de la version</b> . Elle donne aussi le nom de la clef de désinstallation, si AppId n'est pas définie.	
<b>AppVerName</b>	Obligatoire. Très similaire à AppName, mais cette entrée <b>doit indiquer la version</b> du programme.	
AppId	C'est essentiellement le nom de la clef de désinstallation dans le registre. Si cette entrée n'est pas définie, c'est AppName qui est utilisée. Pour éviter tout conflit avec une application qui aurait le même nom que celle que l'on cherche à installer, on peut donner un UUID comme AppId. Par exemple, la clef de désinstallation d'ISTool est « {A9E12684-DD23-4D11-ACAF-6041954BCA00}_is1 ».  Le caractère « { » (accolade ouvrante) est un caractère spécial pour Inno Setup: il doit être doublé. AppId={{A9E12684-DD23-4D11-ACAF-6041954BCA00}}.	

<i>Paramètre</i>	<i>Discussion</i>	<i>Par défaut</i>
PrivilegesRequired	Niveau de privilèges requis pour procéder à l'installation: soit « admin (Administrateur) », soit « none (utilisateur <i>lambda</i> ) ». Sous XP, seul l'administrateur a le droit d'écrire dans certains répertoires, comme \WINDOWS, \Program Files,...	Administrateuradmin.
MinVersion	Version minimale de Windows nécessaire à l'installation. Voir le détail au chapitre « <a href="#">Versions de Windows</a> », page 68.	« 4.0,4.0 », c-à-d Windows NT4, et <b>Windows 95</b> .
OnlyBelowVersion	Version maximale (non-incluse) de Windows nécessaire à l'installation. Voir le détail au chapitre « <a href="#">Versions de Windows</a> », page 68.	« 0,0 », c-à-d pas de contrôle de la version maximale.
DiskSpanning	Permet de fractionner le setup en fichiers de taille prédéfinie (de la taille d'une disquette par exemple).	Le setup n'est pas fractionné. Voir en annexe 13.3, page 95, une discussion sur l'intérêt de cette option.
DiskSliceSize	Si l'option de fractionnement du setup est retenue, ce paramètre donne la taille des fractions.	Taille d'une disquette standard.
OutputDir	Répertoire dans lequel le setup va être construit. Ce répertoire est relatif au script « .iss ».	« ./Output ».
OutputBaseFilename	Nom du setup, sans le répertoire ni le suffixe.	« setup ».
Compression	Algorithme et niveau de compression. En général, il est judicieux de choisir « lzma/ultra ». Il peut être rentable de choisir « zip/9 », si le setup embarque des fichiers précompressés, *.zip, images, fichiers OOo,.... Voir la discussion en annexe 13.4, page 96.	« lzma <sup>1</sup> /max ».

<sup>1</sup> lzma est l'algorithme utilisé par « 7-zip ».

<i>Paramètre</i>	<i>Discussion</i>	<i>Par défaut</i>
VersionInfoCompany	Ces paramètres servent essentiellement à renseigner l'onglet de version du setup. Voir le détail au chapitre « <a href="#">Onglet de version</a> », page 51.	Certains de ces paramètres prennent comme valeur par défaut une autre entrée de la section [Setup]. Voir le chapitre « <a href="#">Valeurs par défaut</a> », page 58.
VersionInfoCopyright		
VersionInfoDescription		
VersionInfoTextVersion		
VersionInfoVersion		
AppPublisher	Ces entrées renseignent la boîte de dialogue « Informations sur le support technique » dans le panneau de contrôle.	
AppPublisherURL		
AppVersion		
AppContact		
AppSupportURL		
AppSupportPhone		
AppReadmeFile		
AppUpdatesURL		
AppComments		
AppCopyright		Notice de copyright. Dans la pratique, fait double emploi avec VersionInfoCopyright. Utiliser de préférence AppCopyright.
DisableStartupPrompt	Permet de <b>ne pas</b> afficher la boîte de dialogue « <a href="#">Cet assistant va installer...</a> », page 26.	La boîte de dialogue n'est pas affichée.
LicenseFile	Fichier de licence que l'utilisateur doit explicitement accepter avant de pouvoir continuer.	Néant.
InfoBeforeFile	Fichier montré à l'utilisateur <b>avant</b> l'installation.	
InfoAfterFile	Fichier montré à l'utilisateur <b>après</b> l'installation.	
Password	Mot de passe nécessaire pour poursuivre l'installation.	
UserInfoPage	Permet d'afficher la boîte de dialogue « <a href="#">Informations sur l'utilisateur</a> » (page 29), dans laquelle l'utilisateur peut entrer son nom, sa compagnie, et un éventuel numéro de série.	Cette boîte de dialogue n'est pas affichée.

<i>Paramètre</i>	<i>Discussion</i>	<i>Par défaut</i>
DefaultUserInfoName	Informations proposées par défaut dans la boîte de dialogue « <a href="#">Informations sur l'utilisateur</a> », page 29.	Utilisateur et organisation déclarés sous Windows.
DefaultUserInfoOrg		
DefaultUserInfoSerial	Ces informations sont mémorisées en clair dans le registre.	
<b>DefaultDirName</b>	C'est le répertoire d'installation par défaut proposé dans la boîte de dialogue « <a href="#">Dossier de destination</a> » page 30.	Paramètre obligatoire. En général, c'est « {pf}\nom du programme », avec « {pf} = \Program Files ».
DisableDirPage	Permet de ne pas afficher la boîte de dialogue « <a href="#">Dossier de destination</a> », page 30.	La boîte de dialogue est affichée.
CreateAppDir	Permet d'installer les fichiers dans \WINDOWS\system32 plutôt que dans un répertoire dédié.	
AlwaysShowComponentsList	Dans la boîte de dialogue « <a href="#">Composants à installer</a> » page 31, cette option permet de ne montrer les composants que si le type d'installation choisi est marqué « personnalisé » ( <i>custom</i> ). Voir en page 95, l'annexe « <a href="#">Effet de AlwaysShowComponentsList</a> » illustrant l'influence de ce paramètre.	
ExtraDiskSpaceRequired	Espace disque nécessaire à l'installation, <b>en plus</b> de celui des fichiers à installer. Cette espace peut être requis par des fichiers créés lors de l'installation ou du lancement de l'application, et qu'Inno Setup ne peut donc pas estimer.	
DefaultGroupName	Valeur par défaut proposée dans la boîte de dialogue « <a href="#">Sélection du dossier du menu Démarrer</a> » page 32.	
DisableProgramGroupPage	Permet de ne pas afficher la boîte de dialogue « <a href="#">Sélection du dossier du menu Démarrer</a> », page 32.	La boîte de dialogue est affichée.

<i>Paramètre</i>	<i>Discussion</i>	<i>Par défaut</i>
AllowNoIcons	Ce paramètre conditionne la présence de la case à cocher «Ne pas créer de dossier dans le menu Démarrer» Voir page <a href="#">32</a> .	
ShowTasksTreeLines	Permet de relier les tâches et les sous-tâches par un arbre, similaire à l'arbre affiché dans la boîte de dialogue de sélection des composants.	L'arbre n'est pas tracé.
DisableReadyPage	Permet de ne pas afficher la boîte de dialogue « <a href="#">Prêt à installer</a> », page <a href="#">33</a> .	La boîte de dialogue est affichée.
DisableReadyMemo	Dans la boîte de dialogue « <a href="#">Prêt à installer</a> », page <a href="#">33</a> , permet de ne pas afficher le résumé des choix qui ont été faits.	Le résumé est affiché.
AlwaysShowDirOnReadyPage	Permet de ne pas afficher certaines informations dans la page de résumé.	Les informations sont affichées.
AlwaysShowGroupOnReadyPage		
DisableFinishedPage	Permet de ne pas afficher la boîte de dialogue « <a href="#">Fin de l'installation</a> », page <a href="#">34</a> .	La boîte de dialogue est affichée.
UsePreviousAppDir	Permet d'utiliser ou non les paramètres sélectionnés lors d'une précédente installation.	Utilisation des informations mémorisées dans le registre.
UsePreviousGroup		
UsePreviousSetupType		
UsePreviousTasks		
UsePreviousUserInfo		
Uninstallable	Permet de supprimer toute possibilité de désinstallation.	La désinstallation est validée.
CreateUninstallRegistry	Permet de ne pas pouvoir désinstaller le programme.	Une clef de désinstallation est créée.
UninstallDisplayIcon	Désigne une icône pour l'entrée de désinstallation dans le « Panneau de contrôle ».	L'icône par défaut est choisie par Windows selon des règles peu déterministes.
UninstallDisplayName	Nom de l'entrée de désinstallation dans le « Panneau de contrôle ».	AppVerName
UninstallFilesDir	Répertoire dans lequel le programme de désinstallation <code>unins0000.exe</code> sera rangé.	Répertoire principal de l'installation.

<i>Paramètre</i>	<i>Discussion</i>	<i>Par défaut</i>
UninstallLogMode		
UninstallRestartComputer		
UpdateUninstallLogAppName		
AlwaysRestart	<p>Permet de forcer un redémarrage du PC, après l'installation.</p> <p>Le redémarrage du PC est aussi conditionné par le flag « restart » de la section [Components] ou de la section [Tasks], ou le flag restartreplace de la section [Files], ainsi que par le paramètre RestartIfNeededByRun.</p> <p>Une fonction Pascal nommée NeedRestart () définie dans le setup peut aussi contrôler la nécessité de rebooter la machine.</p>	Le redémarrage n'est pas forcé.
RestartIfNeededByRun	Inno Setup peut détecter si les programmes de la section [Run] ont fait quelque chose demandant le redémarrage du PC.	La détection de la nécessité de redémarrer le PC est effectuée.
ChangesAssociations	Permet de signaler à Windows pour prise en compte immédiate que des suffixes ont été associés à de nouveaux programmes, et donc à de nouvelles icônes.	
ChangesEnvironment	<p>Permet de signaler à Windows pour prise en compte <b>immédiate</b> que des variables d'environnement ont été modifiées.</p> <p>L'effet <b>n'est pas rétroactif</b>.</p>	

<i>Paramètre</i>	<i>Discussion</i>	<i>Par défaut</i>
SetupLogging	<p>Création d'un fichier journal de l'installation.</p> <p>La création du fichier journal peut toujours être forcée en passant le paramètre « /LOG » dans la ligne de commande.</p> <p>Le cas échéant, le journal est créé dans le répertoire temporaire %TEMP% de l'utilisateur procédant à l'installation; on peut spécifier un autre journal uniquement en donnant un paramètre « /LOG=chemin\fichier.log » dans la ligne de commande.</p>	Pas de création de fichier journal.
WizardImageFile	<p>Image qui apparaît dans la partie gauche des boîtes de dialogue « Bienvenue », page 27, et « Fin de l'installation », page 34.</p> <p>L'image est au format BMP (indexée?), d'un maximum de 164 pixels en largeur et 314 pixels en hauteur.</p>	<p>WizModernImage.bmp:</p> 
WizardSmallImageFile	<p>Image qui apparaît en haut à droite des boîtes de dialogue de la page 27, à la page 34.</p> <p>L'image est au format BMP (indexée?), d'un maximum de 55 pixels en largeur et 58 pixels en hauteur.</p>	<p>WizModernSmallImage.bmp</p> 

### 3.3.2 Section [Types]

Référence dans l'aide en ligne, page « [\[Types\] section](#) ».

Le type d'installation détermine quels composants vont être installés, et quelles tâches vont être effectuées.

La section a l'aspect suivant:

[Types]

**Name:** Typique; **Description:** Installation typique

**Name:** Complete; **Description:** Installation complète

**Name:** Personnalisee; **Description:** Installation personnalisée; **Flags:** **iscustom**

La description est la chaîne qui est montrée à l'utilisateur dans la boîte de dialogue « [Composants à installer](#) », page 31.

Le nom est référencé en interne dans la section [Components] du script, mais il peut aussi être donné en argument dans la ligne de commande pour une exécution en mode non-assisté. Enfin, ce nom est aussi mémorisé dans le registre sous la clef de désinstallation.

Les types sont affichés dans la boîte de dialogue dans l'ordre de leur déclaration dans le script, le premier type déclaré étant celui proposé par défaut à l'utilisateur. Un seul peut être déclaré « **iscustom** »: c'est le seul dont on peut modifier la liste des composants à installer<sup>1</sup>.

Les types d'installation peuvent être conditionnés par la version de Windows de la machine cible, ainsi que par la langue choisie pour l'installation.

### 3.3.3 Section [Components]

Référence dans l'aide en ligne, page « [\[Components\] section](#) ».

Un composant est une unité installable. Dans le cas dégénéré, un composant peut être réduit à un fichier, mais dans le cas général, un composant comptera plusieurs fichiers, comme par exemple un composant « documentation ».

Les composants peuvent être hiérarchisés, en utilisant le caractère « \ » ou « / » pour séparer les niveaux de hiérarchie dans le nom du composant.

La section a l'aspect suivant:

[Components]

**Name:** Executables; \

**Description:** Fichiers exécutables et DLL; \

**ExtraDiskSpaceRequired:** 50000000; \

**Types:** Typique Complete

**Name:** Executables\32bits; \

**Description:** Fichiers exécutables 32 bits; \

**ExtraDiskSpaceRequired:** 100000000; \

**Types:** Typique Complete

**Name:** Executables\64bits; \

**Description:** Fichiers exécutables 64 bits; \

**ExtraDiskSpaceRequired:** 200000000; \

**Types:** Complete

**Name:** Documentation; \

**Description:** Fichiers docs et aide; \

**Types:** Complete

*Composant parent*

*Sous-composants*

Un composant peut faire partie de plusieurs types d'installation.

<sup>1</sup> En fait, on peut modifier la liste des composants avec n'importe quel type d'installation, mais dans ce cas Inno Setup le change en type d'installation personnalisée.

Il est inutile de spécifier le type personnalisé (le cas échéant) parmi les types d'installation auxquels le composant est associé: lorsque l'utilisateur sélectionne le type « personnalisé », les composants sélectionnés par défaut sont ceux du précédent type d'installation sélectionné.

### 3.3.4 Section [Tasks]

Référence dans l'aide en ligne, page « [\[Tasks\] section](#) ».

Cette section est assez similaire à la section [Components], la différence principale étant qu'elle décrit des actions à effectuer au lieu de fichiers à installer.

La sélection des tâches peut être conditionnée par la sélection de composants.

Les tâches peuvent être hiérarchisées de la même façon que les composants, avec le caractère « \ » ou « / ».

La section a l'aspect suivant:

[Tasks]

```
Name: CreerRaccourcis; \  
  Description: Création de raccourcis  
Name: CreerRaccourcis\desktopicon; \  
  Description: Création de raccourcis sur le bureau; \  
  GroupDescription: Icônes supplémentaires; \  
  Flags: unchecked  
Name: CreerRaccourcis\quicklaunchicon; \  
  Description: Création de raccourcis de lancement rapide; \  
  GroupDescription: Icônes supplémentaires; \  
  Flags: unchecked
```

### 3.3.5 Section [Files]

Référence dans l'aide en ligne, page « [\[Files\] section](#) ».

Cette section décrit les fichiers à installer. Pour chaque entrée, on précise le chemin du fichier source, le répertoire d'installation sur la machine cible.

Dans le cas général, le chemin du fichier source est celui du fichier au moment de la compilation, mais il est aussi possible de référencer des fichiers au moment de l'exécution du setup avec le flag **external**; ce flag permet de paramétrer le setup sans toucher au setup.exe, ou bien il permet de donner accès aux fichiers embarqués, ce qui peut être utile pour un setup embarquant d'autres setups. Attention dans ce cas à prendre en compte l'absence éventuel du fichier.

Eventuellement, le fichier peut être installé avec un nom différent de celui du fichier source.

Les fichiers peuvent être désignés par des caractères génériques.

Un exemple assez simple de section [Files]:

```

[Files]
Source: MyProg.exe; \
DestDir: {app}; \
Flags: ignoreversion; \
Components: Executables\32bits
Source: MyProg.chm; \
DestDir: {app}; \
Components: Documentation
Source: lisez_moi.rtf; \
DestDir: {app}; \
DestName: ReadMe.rtf
Source: MyProg-x64.exe; \
DestDir: {app}; \
Components: Executables\64bits

```

Fichier installé avec un nom différent

Pas de composant associé: toujours installé.

De nombreux flags permettent de contrôler la mise à jour de fichiers: mise à jour inconditionnelle, par date de modification, par version, remplacement du fichier au prochain redémarrage du PC...

A noter, que l'aide en ligne signale l'usage d'un flag **isreadme**, mais ce flag est déclaré obsolète dans la page « [\[Run\] & \[UninstallRun\] sections](#) », dans la description du flag **postinstall**.

En général, les fichiers sont copiés dans le répertoire principal de l'installation désigné par la {constante} « {app} » ou un de ses sous-répertoires, ou bien dans le répertoire « \WINDOWS\system32 » désigné par « {sys} », ou enfin dans le sous-répertoire temporaire créé par Inno Setup dans le répertoire %TEMP% de l'utilisateur courant, désigné par « {tmp} ». Si les fichiers sont placés dans « {tmp} », il vont être détruits lorsque le setup va se terminer, il faut donc les utiliser avant: la section [Run] est idéale pour ça.

### 3.3.6 Section [Icons]

Référence dans l'aide en ligne, page « [\[Icons\] section](#) ».

Cette section, devrait plutôt s'appeler « raccourcis ».

Les raccourcis peuvent être conditionnés par des composants ou des tâches, ainsi que par la langue d'installation et la version de Windows de la machine cible.

```

[Icons]
Name: {group}\Mon Programme; \
Filename: {app}\MyProg.exe; \
IconIndex: 0; \
Components: Executables\32bits
Name: {commondesktop}\Mon Programme; \
Filename: {app}\MyProg.exe; \
Tasks: CreerRaccourcis\desktopicon; \
IconIndex: 0; \
Components: Executables\32bits
Name: {userappdata}\Microsoft\Internet Explorer\Quick Launch\Mon Programme; \
Filename: {app}\MyProg.exe; \
Tasks: CreerRaccourcis\quicklaunchicon; \
IconIndex: 0; \
Components: Executables\32bits

```

### 3.3.7 Section [Registry]

### 3.3.8 Section [Run]

Référence dans l'aide en ligne, page « [\[Run\] & \[UninstallRun\] sections](#) ».

Cette section est très intéressante car elle permet la création d'un setup assez évolué, à peu de frais: elle autorise l'exécution d'un programme externe, après que les fichiers décrits dans la section [Files] soient copiés dans leur emplacements désignés.

#### [Run]

```
Filename: {app}\MyProg.exe; \  
Description: Exécuter Mon Programme; \  
Flags: nowait postinstall skipifsilent
```

La commande passée à « **Filename** » peut être de type « .exe », « .bat » ou « .cmd », mais il peut aussi s'agir d'un fichier avec un suffixe quelconque si une application est associée pour lancer un fichier avec ce suffixe. Dans ce dernier cas, le suffixe doit explicitement être précisé.

Par exemple, pour afficher un fichier « .pdf » en fin d'installation (à condition qu'Acrobat Reader soit installé, et que le suffixe « .pdf » lui soit associé):

#### [Run]

```
Filename: {app}\MyProg.pdf; \  
Description: Afficher MyProg.pdf; \  
Flags: nowait postinstall skipifsilent shellexec
```

Si « **Filename** » n'est pas exécutable (cas du « .pdf » en exemple ci-dessus), il faut impérativement placer le flag **shellexec**.

**Filename** ne doit désigner que le fichier que l'on veut exécuter; d'éventuels arguments doivent être donnés avec le paramètre **Parameters**.

#### Cas particulier de l'exécution des commandes internes:

Certaines commandes héritées de DOS ne sont pas des fichiers « .exe » ni même « .com »: ce sont des commandes reconnues directement par l'interpréteur de commandes.

Entre autres, ces commandes comprennent:

DPATH	COLOR	TITLE	CHDIR	CLS	<b>COPY</b> <sup>1</sup>	PATH	DATE
PROMPT	PUSHD	POPD	ASSOC	FTYPE	<b>DEL</b>	<b>DIR</b>	<b>ECHO</b>
ERASE	KEYS	MKDIR	PAUSE	MOVE	RENAME	REN	<b>RMDIR</b>
SET	<b>START</b>	TIME	TYPE	VER	VOL	MD	<b>MKDIR</b>

Les commandes en caractères gras sont celles qui sont d'un usage relativement courant.

Ces commandes étant internes à l'interpréteur de commandes, elle ne peuvent pas être directement utilisées dans la section [Run]. Dans ce cas, c'est l'interpréteur de commandes lui-même qu'il faut appeler, et lui donner la commande interne en argument.

```
Filename: cmd.exe; \  
Parameters: "/c del /f ""{commondesktop}\Ethereal.lnk"" "; \  
Flags: runhidden
```

Le flag **runhidden** évite que des fenêtres DOS ne s'ouvrent pour une fraction de seconde, ce qui aurait un effet perturbateur sur l'utilisateur.

---

<sup>1</sup> A noter, que la commande « xcopy » n'est pas une commande interne: elle correspond à l'exécutable « xcopy.exe ».

Pour la mise au point du script, il peut être intéressant de supprimer momentanément ce flag `runhidden`, et remplacer l'option « /c » par l'option « /k » qui a l'effet inverse: la boîte DOS persiste tant que l'on n'exécute pas manuellement la commande « `exit` ».

Pour les commandes susceptibles de demander plus de quelques secondes pour s'exécuter, il peut être utile d'utiliser le paramètre « `StatusMsg` »; ce paramètre permet de remplacer le message par défaut « Finalisation de l'installation... » par quelque chose comme « Je suis en train de formater le disque dur. Veuillez patienter... ».

### 3.3.9 Section [Code]

Cette section n'a pas du tout le même aspect que les autres sections. Inno Setup confie intégralement le contenu de cette section au compilateur Pascal, **après action du préprocesseur**, ce qui explique que les commentaires de type « ; » ne soient pas acceptés dans cette section, alors que les commentaires Pascal / Delphi « (\* commentaire \*) », « { commentaire } » et « // commentaire » sont reconnus alors qu'ils sont interdits dans le reste du script<sup>1</sup>.

Selon le même principe, les commentaires de types « { commentaire } » peuvent être employés sans précautions particulières bien que le caractère « { » soit spécial pour le compilateur Inno Setup.

## 3.4 Identification du setup

Un setup, comme tout exécutable sous Windows, peut posséder un onglet de version.

Dans le cas d'un setup, il est particulièrement important de renseigner cet onglet de version.

Il est amusant de constater que le setup d'installation d'Inno Setup lui-même possède un onglet de version dans lequel le numéro de version est 0,0,0,0...

D'autres informations d'identification ne sont pas directement visibles depuis le setup: elles apparaissent seulement à partir du « Panneau de contrôle » (« Panneau de configuration »), dans « Ajout-Suppression de programmes ».

### 3.4.1 Nom du setup

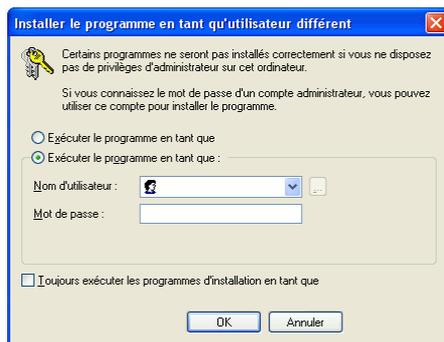
Il existe plusieurs philosophies de nommage de setup. On peut appeler un setup:

- `setup.exe` (c'est le nom par défaut),
- `setup-nom_du_programme_installé.exe`,
- `setup-nom_du_programme_installé-version.exe`
- `nom_du_programme_installé-version-setup.exe`
- ...

---

<sup>1</sup> Les commentaires de type « // » ne sont pas reconnus par Inno Setup, mais le préprocesseur les accepte et les élimine avant de passer à la phase de compilation..

Les noms « setup.exe » et « install.exe » sont privilégiés par Windows: il propose d'exécuter le setup en tant qu'administrateur (si ce n'est pas le cas):



Les chapitres suivants détaillent les paramètres de la section [Setup] relatifs à l'identification de version.

### 3.4.2 Onglet de version

Les entrées `VersionInfoXxx` sont affichées dans l'onglet de version.

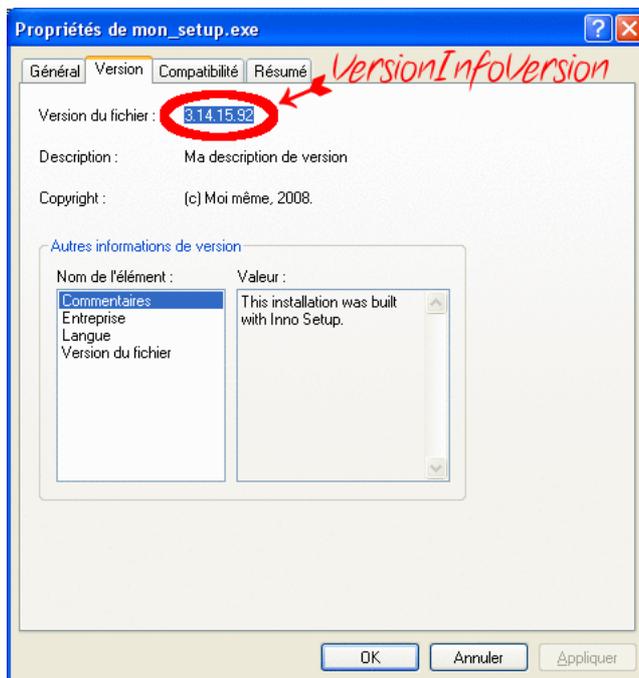
Une possibilité est de récupérer automatiquement tout ou partie de ces informations dans un des exécutables installés.

#### 3.4.2.1 *VersionInfoVersion*

Identificateur de version strictement numérique. La version est composée de 0 à 4 champs, dont la valeur est comprise entre 0 et 65535.

Les champs non-spécifiés prennent la valeur « 0 » (à droite): le cas extrême, est de ne pas du tout renseigner `VersionInfoVersion`, dans ce cas la version dans l'onglet est affichée comme « 0.0.0.0 ».

`VersionInfoVersion=3.14.15.92`

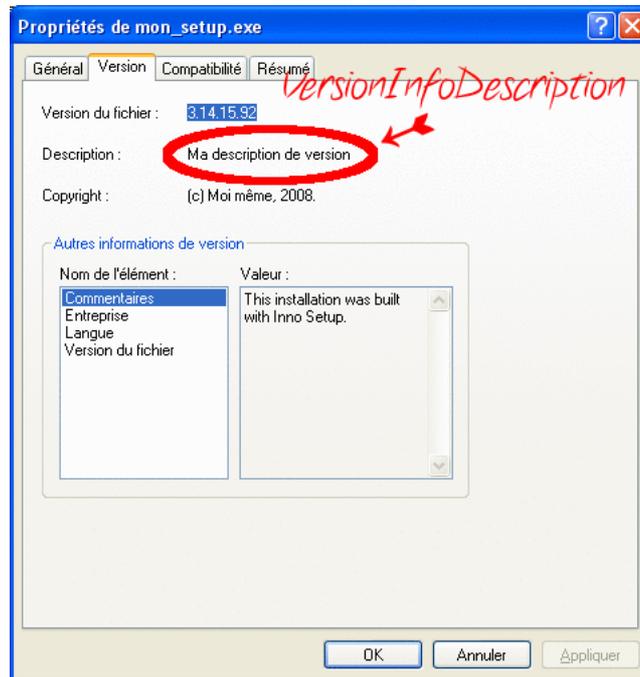


### 3.4.2.2 VersionInfoDescription

[Setup]

...

`VersionInfoDescription=Ma description de version`



Ce paramètre ne peut pas contenir de {constantes}, car celles-ci sont évaluées lors de l'exécution du setup.

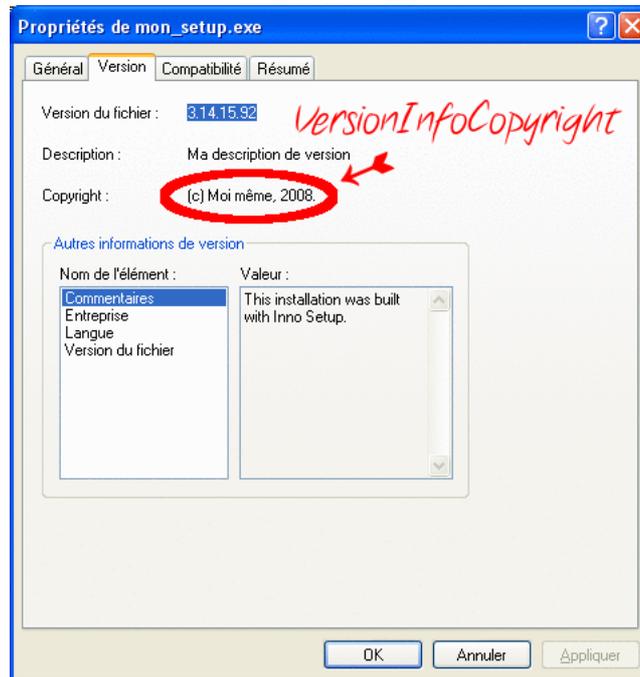
On constate empiriquement que les caractères au-delà du 60<sup>ème</sup> sont perdus, et qu'il n'est pas possible d'insérer de retour-chariot dans la description.

### 3.4.2.3 VersionInfoCopyright

[Setup]

...

**VersionInfoCopyright**=(c) Moi même, 2008.

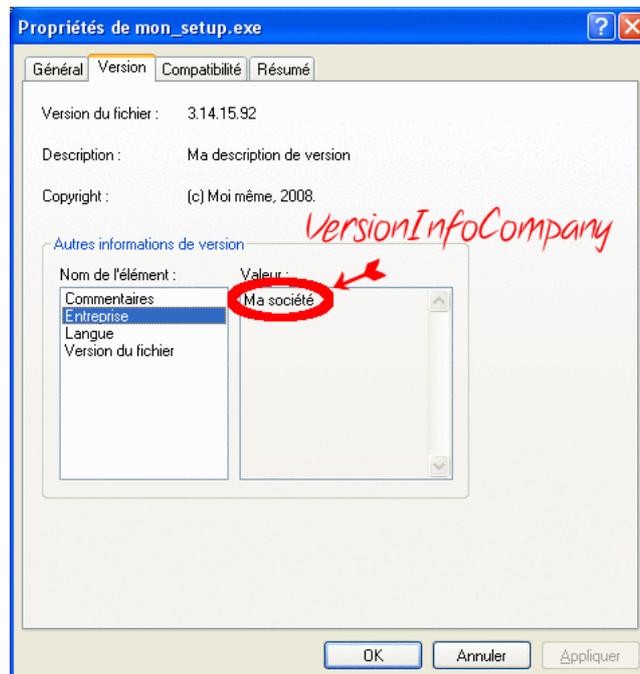


### 3.4.2.4 VersionInfoCompany

[Setup]

...

**VersionInfoCompany**=Ma société

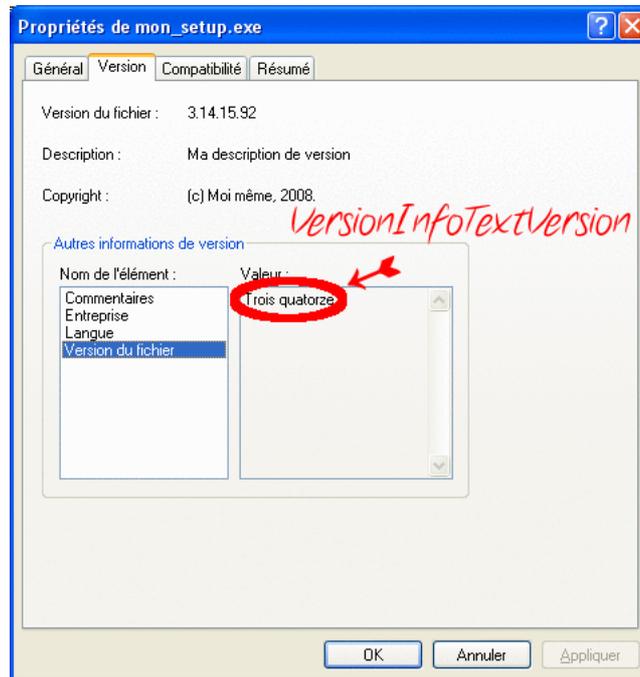


### 3.4.2.5 VersionInfoTextVersion

[Setup]

...

VersionInfoTextVersion=Trois quatorze



### 3.4.3 Panneau de contrôle / Ajout-Suppression de programmes

#### 3.4.3.1 UninstallDisplayIcon

Il peut être utile d'avoir à préciser explicitement une icône, car Windows en affiche une un peu au hasard. On désigne soit un fichier icône « .ico », soit un fichier exécutable (.exe, .dll, .ocx,...) contenant une ressource icône, et un numéro d'ordre de l'icône dans le fichier s'il en contient plusieurs.



Après clic sur « Cliquer ici pour obtenir des informations sur le support technique » :



### 3.4.3.3 AppPublisher + AppPublisherURL

Une entrée donne le texte affiché, l'autre entrée donne l'URL sous le texte.

[Setup]

...

**AppPublisher**=Ma Compagnie, SA

**AppPublisherURL**=<http://www.example.com/publisher>



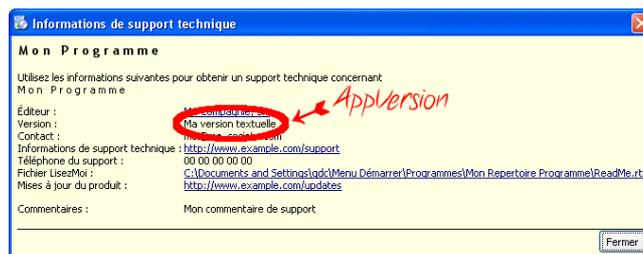
### 3.4.3.4 AppVersion

N'est significatif que pour une installation sous un OS >= XP.

[Setup]

...

**AppVersion**=Ma version textuelle



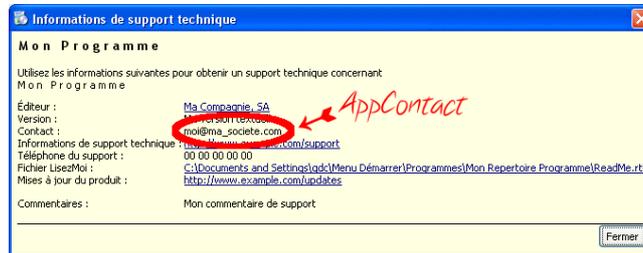
### 3.4.3.5 AppContact

N'est significatif que pour une installation sous un OS >= XP.

[Setup]

...

**AppContact**=moi@ma\_societe.com

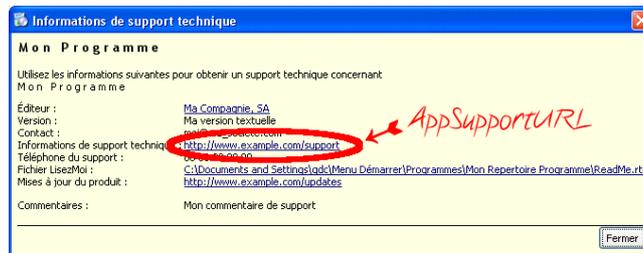


### 3.4.3.6 AppSupportURL

[Setup]

...

**AppSupportURL**=http://www.example.com/support

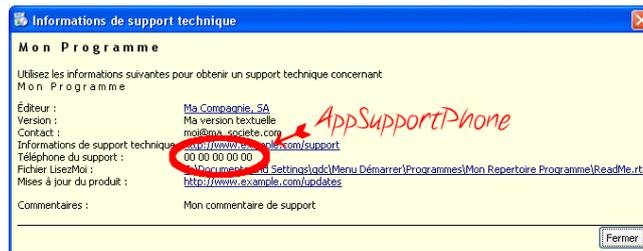


### 3.4.3.7 AppSupportPhone

[Setup]

...

**AppSupportPhone**=00 00 00 00 00



### 3.4.3.8 AppReadmeFile

[Setup]

...

`AppReadmeFile={app}\ReadMe.rtf`



### 3.4.3.9 AppUpdatesURL

[Setup]

...

`AppUpdatesURL=http://www.example.com/updates`



### 3.4.3.10 AppComments

N'est significatif que pour une installation sous un OS >= XP.

[Setup]

...

`AppComments=Mon commentaire de support`



## 3.5 Valeurs par défaut cascadées

Certaines entrées de la section [Setup] prennent des valeurs par défaut qui sont d'autres entrées de la même section [Setup], ou fournies par Inno Setup.

A moins d'avoir un besoin très particulier, généralement il ne sera nécessaire de n'utiliser que les entrées de la colonne « Valeur par défaut »; quelques cas peuvent justifier de définir les deux entrées:

- Pour contourner une limitation de la valeur par défaut (pas de constante possible par exemple, ou pas le même format comme pour `VersionInfoVersion`, numérique, et

VersionInfoTextVersion, texte),

- Lorsque les deux entrées n'ont pas la même fonction, comme AppName et VersionInfoDescription par exemple.

<i>Entrée</i>	<i>Valeur par défaut</i>
VersionInfoCompany	AppPublisher
VersionInfoCopyright	AppCopyright
VersionInfoDescription	AppName
VersionInfoTextVersion	VersionInfoVersion <sup>1</sup>
AppId	AppName
DefaultUserInfoName	{sysuserinfoname}
DefaultUserInfoOrg	{sysuserinfoorg}
UninstallDisplayName	AppVerName
UninstallFilesDir	{app}

---

<sup>1</sup> Version strictement numérique

### 3.6 {Constantes}

Ce que Inno Setup qualifie de constantes (notées dans ce document « {constante} ») ne sont en fait pas des constantes: ces *constantes* sont **évaluées dynamiquement au moment de l'exécution du setup**, et non pas lors de la compilation comme le seraient de véritables constantes; ce sont plutôt des paramètres d'exécution qui ne sont pas modifiables lors de leur utilisation.

La référence dans l'aide en ligne: chapitre « [Constants](#) ».

Certaines constantes existent par paires: l'une a son nom préfixé par « common », elle concerne tous les utilisateurs, le nom de l'autre commence par « user », elle désigne un répertoire propre à l'utilisateur courant, c'est-à-dire celui qui procède à l'installation.

En général, l'installation ayant lieu sous un compte « administrateur », l'utilisation de constantes relatives à cet utilisateur est d'intérêt limité.

<i>{constante}</i>	<i>Utilisation</i>
{app}	Répertoire principal d'installation. C'est le répertoire choisi avec la boîte de dialogue « <a href="#">Dossier de destination</a> », page <a href="#">30</a> .
{cmd}	Chemin de l'interpréteur de commandes DOS (shell) sous Windows. Peut servir dans la section [Run], pour exécuter des commandes internes au DOS, comme « copy <sup>1</sup> ». L'intérêt de ce paramètre semble limité car il faut en général passer l'option « /C » à ce shell, et donc faire l'hypothèse qu'il s'agit de « cmd.exe <sup>2</sup> » (sous NT, 2000 et XP).
{commonappdata}, {userappdata}	Chemin du répertoire « Application Data ».
{commondesktop}, {userdesktop}	Chemin du répertoire correspondant au Bureau. Sert en général pour créer des raccourcis.

<sup>1</sup> On préférera utiliser xcopy.exe, beaucoup plus souple d'emploi, et qui est une commande externe et donc ne nécessite pas « {cmd} ».

<sup>2</sup> « cmd.exe » est dans le PATH.

<i>{constante}</i>	<i>Utilisation</i>
{commonprograms}, {userprograms}	Chemins de divers répertoires correspondant au menu « Démarrer » de Windows, et ses sous-menus. Un raccourci placé dans le répertoire « Démarrage / Startup » lancera l'application concernée lorsque l'utilisateur se loggiera.
{commonstartmenu}, {userstartmenu}	
{commonstartup}, {userstartup}	
{computername}	Nom du PC sur lequel s'effectue l'installation.
{groupname}	
{group}	
{language}	
{log}	Chemin complet du fichier de traces du setup, si les traces ont été autorisées avec le paramètre SetupLogging de la section [Setup], ou bien avec l'option /LOG donnée dans la ligne de commande.
{pf}	Emplacement du répertoire « Program files », qui n'est pas obligatoirement dans « C : \ », et qui ne s'appelle pas forcément de ce nom: en espagnol, c'est « Archivos de programa ».
{srcexe}	Chemin complet du setup en cours d'exécution. Le setup a peut-être été renommé par l'utilisateur...
{src}	Répertoire dans lequel se trouve le setup en cours d'exécution. Peut servir pour référencer un fichier relativement à l'emplacement du setup.
{sysuserinfo}	Nom et organisation de l'utilisateur, tels qu'ils ont été saisis lors de la création du compte sous Windows.
{sysuserinfoorg}	
{sys}	Chemin du répertoire System32. Pour la mise à jour de <b>DLLs système</b> sous <b>XP</b> , il n'est pas suffisant de placer ces DLLs dans ce répertoire: le mot-clef est « WinSxS ».

<i>{constante}</i>	<i>Utilisation</i>
{tmp}	Chemin du sous-répertoire temporaire créé par le setup, et détruit à la fin. Ce sous-répertoire est créé dans le répertoire %TEMP% de l'utilisateur, c-à-d en général « C:\Documents and Settings\utilisateur\Local Settings\Temp ». Le sous-répertoire propre à une installation Inno Setup est de la forme IS-xxxxx.tmp. Il peut parfois être nécessaire de détruire manuellement ces sous-répertoires, suite à un crash du setup.
{uninstallexe}	Chemin complet de l'exécutable de désinstallation du setup courant. En général, le fichier s'appelle unins000.exe.
{userinfo}	Informations fournies par la boîte de dialogue « <a href="#">Informations sur l'utilisateur</a> » (voir page 29).
{userinfoorg}	
{userinfoserial}	
{username}	Nom de l'utilisateur (son <i>login</i> ) procédant à l'installation.
{win}	Nom du répertoire principal de Windows (« C:\WINDOWS »). En général, c'est plutôt « C:\WINDOWS\System32 » qu'il faut utiliser (donné par « {sys} »).

La syntaxe des {constantes} permet également de consulter l'environnement:

<i>{constante}</i>	<i>Utilisation</i>
{%NAME DefaultValue}	Récupère la valeur de la variable d'environnement NAME, et retourne la valeur DefaultValue si la variable d'environnement n'est pas définie.
{ini:Filename,Section,Key DefaultValue}	Retourne l'entrée Key, section Section, du fichier Filename.ini, avec la valeur par défaut DefaultValue.
{reg:HKxx\SubkeyName,ValueName DefaultValue}	Idem lecture de fichier .ini, mais ici on consulte le registre Windows.

<i>{constante}</i>	<i>Utilisation</i>
{param:ParamName   DefaultValue}	<p>Récupération de l'option arbitraire /ParamName=value passée dans la ligne de commande, avec la valeur par défaut DefaultValue.</p> <p>Le nom du paramètre est indépendant de la casse. Il doit impérativement être préfixé par un slash, par opposition à l'autre convention qui préfixe les options par des tirets.</p> <p>Cette syntaxe permet de récupérer les options standard d'Inno Setup passées dans la ligne de commande, mais aussi des options personnalisées.</p>
{code:FunctionName   Param}	<p>Donne la chaîne de caractères retournée par la fonction Pascal FunctionName, à laquelle on a donné le paramètre Param.</p> <p>La syntaxe est décrite dans l'aide en ligne à la section « <a href="#">Pascal Scripting: Scripted Constants</a> ».</p>
{cm:MessageName }	<p>Donne le message personnalisé (cm=Custom Message) identifié MessageName, en fonction de la langue d'installation sélectionnée.</p> <p>Les messages personnalisés doivent être définis dans les langues supportées par le setup, dans la section [CustomMessages] non-décrite dans le présent document.</p> <p>Il est possible d'utiliser les messages prédéfinis d'Inno Setup, décrits dans les fichiers .isl du répertoire d'installation d'Inno Setup.</p>

## 4 Le préprocesseur

### 4.1 Généralités

Le préprocesseur ISPP est similaire au préprocesseur « C »: il opère des **substitutions de texte avant la phase de compilation**. De par ce principe, on peut utiliser des directives du préprocesseur dans la section [Code] en Pascal.

Par rapport au préprocesseur « C », ISPP a la possibilité de définir des pseudo-fonctions, avec des variables locales. Il possède une instruction « for ».

Il propose de nombreuses fonctions prédéfinies, permettant (entre autres) de manipuler des chaînes de caractères, de lire et d'écrire des fichiers .ini, et de consulter la version d'un fichier désigné.

La directive « #include » est une directive du préprocesseur, mais cette directive est également directement reconnue (au moins dans sa forme la plus simple) par le compilateur Inno Setup: c'est la seule directive de la forme « #xxx » dans ce cas.

### 4.2 Commentaires

Le préprocesseur apporte les commentaires préfixés par « // ». Contrairement aux autres styles de commentaires utilisables dans le script Inno Setup, ceux du préprocesseur ont l'avantage d'être utilisable dans la totalité du script. Ils ont cependant l'inconvénient de fausser les numéros de lignes donnés dans les diagnostics de compilation: le préprocesseur élimine en totalité les lignes de commentaires préfixées par « // », et donne le résultat au compilateur, qui donne ses diagnostics en numérotant les lignes données par le préprocesseur. Les numéros de lignes donnés par le compilateur ne sont plus ceux du fichier source original.

### 4.3 Directive #define

C'est la directive la plus couramment employée.

Tout comme la même directive du « C », elle n'est pas limitée à la définition de constantes: elle définit des macros, la constante n'étant que le cas dégénéré de la macro.

Une macro peut bien sûr être définie à partir d'une autre macro, soit prédéfinie, soit définie explicitement par un #define préalable.

Par exemple:

```
#define NOM_COURT "xxx"
#define VERSION_NUM ReadIni( ".\SetupXxx .ini", NOM_COURT, "Version", "?" )
#define VERSION_CHAINS VERSION_NUM
#define NOM_LONG ReadIni( ".\SetupXxx.ini", NOM_COURT, "Name", "?" )
```

*Macro prédéfinie*

Pour son utilisation dans le corps du script, ou dans une fonction Pascal, le nom de la macro utilisée doit être encadrée par « {# » et « } ». Cette règle ne s'applique pas si la macro est utilisée dans la définition d'une autre macro, comme dans l'exemple ci-dessus.

Exemples d'utilisation dans le corps du script et en Pascal:

```

[Setup]
AppName={#NOM_LONG}
OutputBaseFilename=setup-#{#NOM_PGM_COURT}
[...]
[Code]

procedure CurStepChanged( CurStep: TSetupStep );
//      =====
var
    on_continue:    Boolean;
begin
    on_continue := true;    // a priori
    // ssInstall: just before the actual installation starts
    if CurStep = ssInstall then
    begin
        on_continue := PeutDesinstaller( '#{#NOM_LONG}' );
        if on_continue then
            on_continue := on_continue and VerifieXxxEstInstalle;
        if on_continue then
            on_continue := on_continue and VerifieUtilisateurYyy;
        if not on_continue then
        begin
            Abort;
        end;
        if IsComponentSelected( 'ComposantZzz' ) then
        begin
            Desinstaller( '#{#NOM_LONG}' );
        end;
    end;
end;
end;

```

#### 4.4 Pseudo-fonctions de manipulation des fichiers .ini

La fonction ReadIni() permet de lire depuis un fichier au format .ini. Cette fonction peut être mise à profit pour récupérer des paramètres du projet à compiler.

Il existe également la fonction symétrique WriteIni() qui permet d'écrire dans un fichier.

Comme toutes les fonctions prédéfinies du préprocesseur, elles ne sont utilisables que lors de la compilation du setup par opposition à l'utilisation lors de l'exécution du setup; dans ce dernier, on utilisera une *{constante}* de la forme « {ini:Filename,Section,Key|DefaultValue} », (voir page 62).

Le piège de ces fonctions est que le nom du fichier .ini doit être absolu, ou bien doit être préfixé par « \ » si le fichier est relatif au répertoire courant d'exécution **du compilateur**.

Exemple de macro, permettant de lire des paramètres tout-ou-rien, depuis un fichier .ini:

```

;
; Macro dérivée de YesNo(), définie dans Builtins.iss.
;
; Macro générique, retourne vrai, si le paramètre existe et a une valeur
; possible pour "vrai".
;
#define ParametreEstVrai( str fic_ini, str section, str clef )      \
    Local[0] = LowerCase( ReadIni( ".\"+fic_ini, section, clef, "0" ) ), \
    Local[0] == "oui"           ||                                     \
    Local[0] == "yes"           ||                                     \
    Local[0] == "vrai"          ||                                     \
    Local[0] == "true"          ||                                     \
    Local[0] == "1"

```

## 4.5 Ambiguïté dans la section [Code]

Les directives du préprocesseur commencent avec le caractère « # ». Ce même caractère est utilisé en Pascal/Delphi pour représenter un caractère par son code décimal; c'est le moyen d'enfourer des retours chariot dans une chaîne de caractères en insérant « #13#10 » (CR/LF, "\r\n", 0x0d/0x0a), comme dans la procédure InitializeWizard en exemple page 75.

La forme suivante n'est pas ambiguë:

```
CreateInputOptionPage( Page3ChoixNonExclusif.ID,
    'Choix non-exclusif',
    'Faites encore un choix',
    'Choisissez une option.'#13#10
    'Plusieurs choix sont possibles.',
    False,
    True );
```

Le même appel de procédure, avec une mise en page différente devient ambigu:

```
CreateInputOptionPage( Page3ChoixNonExclusif.ID,
    'Choix non-exclusif',
    'Faites encore un choix',
    'Choisissez une option.'
    #13#10'Plusieurs choix sont possibles.',
    False,
    True );
```

Dans ce cas, le caractère « # » est le premier caractère non-blanc de la ligne: il est interprété par le préprocesseur, qui ne reconnaît pas une directive correctement formée.

Ici, la chaîne de caractères a été constituée par juxtaposition de trois chaînes élémentaires, dont l'une est définie par le code décimal de ses caractères.

On peut aussi opter pour l'opérateur de concaténation de chaînes « + », avec la forme:

```
CreateInputOptionPage( Page3ChoixNonExclusif.ID,
    'Choix non-exclusif',
    'Faites encore un choix',
    'Choisissez une option.'
    + #13#10' + Plusieurs choix sont possibles.',
    False,
    True );
```

Cette forme est sûrement moins optimisée: la concaténation a probablement lieu lors de l'exécution du code, alors que dans le premier cas, la concaténation est a priori faisable lors de la compilation.

## 4.6 Effet secondaire du préprocesseur

Les lignes de certaines sections du script peuvent être assez longues, en particulier dans la section [Files]. Dans le monde Unix, il existe une convention pour replier une ligne: il faut insérer à la coupure de la ligne le caractère « \ » (backslash) **immédiatement** suivi d'un retour chariot.

Cette convention est reconnue par le préprocesseur d'Inno Setup<sup>1</sup>, tout comme le préprocesseur du langage « C ».

Cependant, **ISTool ne connaît pas cette convention**: en présence d'une ligne repliée par un backslash, il ajoute intempestivement des caractères « ; » (point-virgule) et « : » (deux points).

---

<sup>1</sup> « line spanning » dans la terminologie ISPP.

## 5 ISTool

Par principe, ISTool ne demande pas une description très poussée, puisqu'il s'agit d'un assistant permettant de faciliter l'utilisation d'Inno Setup.

L'emploi d'ISTool est impossible dans les cas suivants:

1. le script comporte volontairement plusieurs sections de mêmes noms,
2. les lignes trop longues ont été repliées avec le caractère « \ ».

Dans ces deux cas, non seulement le script ne compile pas, mais ISTool reformate assez radicalement le script original. Le palliatif est d'empêcher la compilation sous ISTool, en incluant une section:

```
[_ISToolPreCompile]
Name: cmd /c echo; Parameters: Ce script ne doit pas être compilé ni SAUVEGARDÉ
sous ISTool.
Name: cmd /c echo; Parameters: Utilisez "Inno Setup" à la place.
Name: cmd /c exit; Parameters: 1; Flags: abortonerror
```

Cette section aura avantage à être placée en début de fichier, de façon à être visible dès l'ouverture du script.

En dehors de ce cas précis, la section [\_ISToolPreCompile] semble faire double emploi avec le préprocesseur.

Deux autres sections paraissent plus intéressantes: [\_ISToolPostCompile] qui est un postprocesseur, et [\_ISToolDownload] qui permet de télécharger un fichier depuis Internet.

Ces sections ne sont pas suffisamment documentées pour juger de leur intérêt.

La section [\_ISTool] est utilisée par ISTool pour ses besoins propres: elle permet de mémoriser certains paramètres.

## 6 Versions de Windows

Inno Setup permet de discriminer les versions de Windows sur lesquelles l'installation doit se faire.

Ces versions sont sans relation avec les versions de Windows sur lesquelles Inno Setup lui-même peut être installé et exécuté.

Dans la terminologie Inno Setup, dans le contexte des versions, « **Windows** » désigne la famille Windows 95 / 98 / Me, tandis que « **Windows NT** » est l'appellation générique de la famille Windows NT / 2000 / XP ainsi que Vista.

Les familles Windows 3.1 et Windows NT 3.51 ainsi que Windows CE ne sont pas des cibles possibles.

L'identification quasi-complète des versions de Windows est donnée à la page « [Windows Versions](#) » de l'aide contextuelle d'Inno Setup.

Avec la famille « Windows NT », le *build number* est redondant: il peut ne pas être précisé dans les champs `MinVersion` et `OnlyBelowVersion`.

La seule chose manquante est l'identification des Services Packs (ceux-ci sont discutés dans la page « [Using Build Number and/or Service Pack Levels](#) » de l'aide en ligne): on peut suffixer la version avec « `sp1`, `sp2`,... ». Le plus simple, est de renseigner les paramètres de version avec ISTool (celui-ci ne connaît pas XP SP2 ni SP3, mais on peut changer manuellement SP1 en SP2/SP3 dans le script: le SP2/SP3 est correctement pris en compte par Inno Setup).

A noter, que la version de XP est « **5.01** », et non pas « ~~5.1~~ ».

Les valeurs par défaut de `MinVersion` et `OnlyBelowVersion` autorisent l'installation sur une version quelconque de Windows.

Si l'on veut se limiter à des cibles de NT4 SP4 à XP, il convient d'inclure dans le script:

```
MinVersion=0,4.0.1381sp4
```

```
OnlyBelowVersion=0,6.0
```

La plage de versions de Windows autorisée est spécifiée globalement dans la section `[Setup]`. Elle est contrôlée dès le lancement du setup.

Il est possible d'installer des fichiers ou d'exécuter des tâches différentes selon la version de Windows sur la machine cible, avec les paramètres `MinVersion` et `OnlyBelowVersion` des sections `[Files]`, `[Tasks]`,... qui sont similaires aux entrées de même nom de la section `[Setup]`, mais qui sont néanmoins distincts.

## 7 Pascal

Il est possible de créer des setups parfaitement opérationnels sans avoir à recourir à Pascal, mais quelques lignes de codes bien placées peuvent apporter un grand gain de fonctionnalités.

Il peut être nécessaire de recourir à Pascal, essentiellement dans les deux cas suivants:

1. Pour implémenter une fonctionnalité pointue,
2. Pour greffer un traitement à certaines étapes-clefs du déroulement du setup.

Pour intercepter un événement, il suffit de définir une fonction Pascal avec le prototype spécifié dans la page « Pascal Scripting: Event Functions » de l'aide en ligne.

L'éditeur syntaxique d'ISTool ne distingue pas spécialement ces routines des autres routines, contrairement à l'éditeur d'Inno Setup, qui affiche leur nom en caractères gras.

Les principales routines prédéfinies de traitement d'événements sont les suivantes:

<i>Routine</i>	<i>Description</i>
<code>InitializeSetup</code>	Appelée à l'initialisation du setup: en retournant un compte-rendu adéquat, elle permet d'interrompre le setup. Etant appelée à l'initialisation, de nombreux paramètres ne sont pas encore significatifs à ce stade.
<code>DeinitializeSetup</code>	Cette procédure est appelée avant que le setup ne se termine, que ce soit après une installation arrivée à terme, ou bien que celle-ci ait avorté.
<code>CurStepChanged</code>	Cette fonction est appelée en 3 points du déroulement du setup: <ul style="list-style-type: none"><li>● <code>ssInstall</code>: immédiatement avant l'installation proprement dite (juste après la boîte de dialogue « Fin de l'installation », page 34),</li><li>● <code>ssPostInstall</code>: immédiatement après l'installation proprement dite,</li><li>● <code>ssDone</code>: juste avant la sortie du setup, si celui-ci s'est bien passé.</li></ul> <code>CurStepChanged(ssInstall)</code> est appelée avant que les fichiers ne soient extraits. Si dans cette procédure on veut utiliser des fichiers (temporaires), il faut les extraire explicitement avec la fonction Pascal <code>ExtractTemporaryFile()</code> .
<code>CurPageChanged</code>	Appelée après l'affichage de chacune des boîtes de dialogue des pages 27 à 34.
<code>CheckPassword</code>	Permet de vérifier un mot de passe. Le simple fait de définir cette fonction provoquera l'ouverture de la boîte de dialogue « Mot de passe », page 28.

### 7.1 Différences C / Pascal

Ce chapitre est un rappel pour ceux dont l'expérience en « C » est plus importante que celle en Pascal

Ici, l'appellation « Pascal » est considérée comme générique pour Pascal et Delphi<sup>1</sup>.

Autre convention: on appelle routine, une fonction ou une procédure.

Le tableau qui suit est tout au plus un pense-bête, et n'a en aucun cas la prétention de constituer une référence.

Un très bon condensé de Pascal pour Inno Setup est disponible à l'URL

<http://www.mirality.co.nz/inno/tips.php>.

<i>Sujet</i>	<i>Pascal</i>
Ordre de définition des fonctions.	La compilation se fait en une passe. Les fonctions et procédures doivent en principe être définies avant d'être utilisées. L'utilisation de directives « forward » (l'équivalent de « extern » du « C ») doit rester exceptionnelle.
Fonction / Procédure	En Pascal, une fonction retourne toujours une valeur. Une fonction qui ne retourne pas de valeur s'appelle une procédure. Dans ce document, le terme « routine » est une appellation commune aux procédures et aux fonctions.
Valeur de retour des fonctions	En « C », une fonction retourne une valeur avec l'instruction « return ». En Pascal, la valeur de retour est affectée à une pseudo-variable « Result ».
Casse des identificateurs	La casse des identificateurs n'est pas significative. MaFonction, MAFUNCTION, ma fonction et MaFoNcTiOn désignent tous les quatre la même fonction. Il est de coutume d'utiliser des identificateurs en minuscules, avec des majuscules pour les premiers mots composant l'identificateur. Ex: MaFonction.
Bloc	En « C », un bloc est délimité par les accolades « { » et « } ». En Pascal, un bloc est délimité par les mots-clefs « begin » et « end ».
Définition de fonctions	En Pascal, une routine peut être définie dans une autre routine.
	Lors de la déclaration de variable, le type et le nom de la variable sont inversés en Pascal par rapport au « C ». « C »: <code>int i;</code> Pascal: <code>I: Integer;</code>
Affectation	L'opérateur de comparaison est « = » (simple caractère « égale »). L'opérateur d'affectation est « := » 'deux-points, égale).
Parenthèses vides	Les parenthèses vides ne sont pas nécessaires à l'appel d'une routine ne prenant pas de paramètres, mais elles sont acceptées par le compilateur.
Chaînes de caractères	Les chaînes de caractères sont délimitées par des apostrophes (des guillemets en « C »).
	L'indice du premier caractère d'une chaîne est 1.
	Les chaînes de caractères ne sont pas des tableaux.
	Les chaînes de caractères ne sont pas terminées par des caractères nuls.

<sup>1</sup> Delphi ne s'appelle pas Pascal pour des raisons de marketing.

<i>Sujet</i>	<i>Pascal</i>
Backslash	Le caractère « \ » (backslash) n'a pas de signification particulière dans les chaînes. Il n'a pas besoin d'être doublé.
	<p>Dans l'alternative:</p> <pre>if condition then     bloc1 else     bloc2;</pre> <p>il ne faut <b>pas de point-virgule</b> après bloc1<sup>1</sup>.</p>
Expressions booléennes court-circuitées	Voir plus en détail, la discussion au paragraphe 7.2, page 71.
	Les constantes peuvent ne pas être explicitement typées.
Priorité des opérateurs	La priorité des opérateurs est différente du « C ». L'usage systématique de parenthèses est vivement recommandé. Par exemple: « if a = b and c then ... » n'est pas évaluée comme on pourrait s'y attendre; il convient d'écrire « if ( a = b ) and c then ... ».
	Un point-virgule est nécessaire après la déclaration du nom de la routine, avant le corps de la routine.

## 7.2 Expressions booléennes court-circuitées

Une caractéristique officielle importante du langage « C » et des langages dérivés est qu'une expression **logique** à deux opérandes est évaluée de la gauche vers la droite, et que le deuxième opérande n'est évalué que si c'est nécessaire.

Exemple:

```
int oui()
{
    return 1;
}
```

```
int non()
{
    return 0;
}
```

<sup>1</sup> Il existe d'autres cas où le point-virgule n'est pas requis mais est toléré par le compilateur en tant qu'instruction nulle.

```
int main()
{
    return oui() || non();
}
```

Dans cet exemple, la fonction `non()` n'est pas appelée, car l'appel à la fonction `oui()` a suffi à évaluer l'expression finale comme vraie.

Extrait du chapitre « C Language Reference » de MSDN:

*The operands of logical-AND and logical-OR expressions are evaluated from left to right. If the value of the first operand is sufficient to determine the result of the operation, the second operand is not evaluated. This is called "short-circuit evaluation."*

En Pascal, la situation est moins nette: le Pascal d'origine semble ne pas permettre de court-circuit. Delphi laisse le choix au développeur, par le biais d'une directive de compilation.

Extrait du paragraphe « Boolean operators » de l'aide de Delphi v.3.0:

*Object Pascal supports two different models of code generation for the **and** and **or** operators: complete evaluation and short-circuit (partial) evaluation.*

*Complete evaluation means that every operand of a Boolean expression built from the **and** and **or** operators is guaranteed to be evaluated, even when the result of the entire expression is already known. This model is convenient when one or more operands of an expression are functions with side effects that alter the meaning of the program.*

*Short-circuit evaluation guarantees strict left-to-right evaluation and that evaluation stops as soon as the result of the entire expression becomes evident. This model is convenient in most cases because it guarantees minimum execution time, and usually minimum code size. Short-circuit evaluation also makes possible the evaluation of constructs that would not otherwise be legal. For example,*

```
while (I <= Length(S)) and (S[I] <> ' ') do
    Inc(I);
while (P <> nil) and (P^.Value <> 5) do
    P := P^.Next;
```

*In both cases, the second test isn't evaluated if the first test is False.*

*The evaluation model is controlled through the **\$B** compiler directive. The default state is **{\$B-}**, and in this state, the compiler generates short-circuit evaluation code. In the **{\$B+}** state, the compiler generates complete evaluation.*

Cependant, Inno Setup n'utilise pas le moteur de compilation de Delphi, mais celui de RemObjects, qui semble ne pas reconnaître la directive de compilation `$B`. Empiriquement, on constate que par défaut le court-circuit d'expressions logiques est pratiqué sous Inno Setup.

Conclusion: à moins que le comportement du compilateur Pascal d'Inno Setup ne soit précisé, il est préférable d'éviter de dépendre du comportement par défaut actuel. En conséquence, on utilisera plutôt des constructions du genre:

```

function OuiOuNon() : Boolean;
begin
    Result := Oui();
    if not Result then
    begin
        Result := Non();
    end;
end;

```

### **7.3 Création de pages personnalisées**

La sélection des composants à installer (page 31) et des tâches à effectuer (page 32), autorise une certaine souplesse dans la construction du setup, en utilisant uniquement les fonctionnalités de base d'Inno Setup.

L'étape suivante est de créer une boîte de dialogue totalement personnalisée.

Cependant, chaque bouton, chaque case à cocher doit être dessiné par programme (Pascal) et géré individuellement.

Un moyen terme est d'utiliser une boîte de dialogue personnalisable, permettant de faire des saisies simples. La mise en oeuvre est relativement simple: en phase d'initialisation, on crée un objet de portée globale du type adéquat, en indiquant où insérer la page dans la séquence, et on renseigne ensuite dans cet objet le choix proposé par défaut.

Cette page est plus souple que la sélection du type d'installation, car les types d'installation sont prédéfinis, alors que cette page est construite dynamiquement.

Un exemple est fourni avec l'installation de Inno Setup:

C:\Program Files\Inno Setup 5\Examples\CodeDlg.iss.

Il existe quatre types principaux de pages personnalisables:

1. Page de choix, avec des boutons radio ou des cases à cocher,
2. Page de saisie,
3. Page de sélection de répertoire,
4. Page de sélection de fichier.

La page totalement personnalisée constituant une cinquième possibilité.

Ne sont détaillées ici que le deux premiers types.

Il est préférable de respecter les principes d'Inno Setup lorsqu'on insère une page personnalisée dans le: aucun traitement ne doit être fait avant que l'utilisateur n'ait accepté globalement tous les choix qui lui ont été proposés, avec la boîte « Prêt à installer », page 33. Cette boîte peut ne pas être affichée, si l'option `DisableReadyPage` est prise, mais cette étape existe malgré tout.

Lorsqu'on clique sur le bouton « Suivant » de la page personnalisée, il est possible de greffer un contrôle de la saisie en définissant une fonction `NextButtonClick()`, qui autorise le passage à la page suivante en retournant « True ».

Le traitement proprement dit consécutif aux choix faits dans la boîte de dialogue personnalisée peut être fait dans la fonction `CurStepChanged()`, appelée immédiatement avant l'installation proprement

dite, ou immédiatement après. S'il est nécessaire de greffer un traitement à un instant bien précis **pendant** l'installation, les paramètres BeforeInstall et AfterInstall des sections [Files] et [Run] sont prévus pour ça.

Exemple de contrôle:

```
function NextButtonClick( CurPageID: Integer ): Boolean;
begin
    Result := True; (* par défaut, le passage à la page suivante *)
                  (* est autorisé *)

    (* Il n'est pas possible d'utiliser "case CurPageID of", *)
    (*car l'identificateur de page personnalisée n'est pas une constante *)
    if CurPageID = PagePersonnalisee.ID then
    begin
        Result := parametres_saisis_sont_corrects;
    end;
end;
```

### 7.3.1 Page de choix

Classe: TInputOptionWizardPage

Fonction de création: CreateInputOptionPage

Cette page propose à l'utilisateur de faire un choix, soit à partir de boutons radio, une possibilité, et une seule, soit à partir de cases à cocher auquel cas toutes les combinaisons sont permises.

#### 7.3.1.1 Création de la page

La création de la page comprend quatre temps:

1. déclaration d'une variable globale de type TInputOptionWizardPage,
2. création de la page vierge (avec seulement les textes),
3. création des cases à cocher (ou boutons radio),
4. préselection des options par défaut.

```
var
    (* Variables globales *)
[...]
    Page4ChoixNonExclusif: TInputOptionWizardPage;
[...]
```

```

procedure InitializeWizard;
begin
[...]
```

*Fonction d'initialisation standard.*

*Création de la page, vierge.*

```

    Page4ChoixNonExclusif :=
        CreateInputOptionPage( Page3ChoixNonExclusif.ID,
            'Choix non-exclusif',
            'Faites encore un choix',
            'Choisissez une option.'#13#10
                'Plusieurs choix sont possibles'#13#10
                'ListBox := True;',
            False,
            True );
```

```
[...]
```

```

    Page4ChoixNonExclusif.Add( 'Autre choix #0' );
    Page4ChoixNonExclusif.Add( 'Autre choix #1' );

    Page4ChoixNonExclusif.Add( 'Autre choix #15' );
    Page4ChoixNonExclusif.Add( 'Autre choix #16' );
```

*Création des cases à cocher (ou boutons radio).*

```

    (* Les indices commencent à zéro *)
    Page4ChoixNonExclusif.Values[ 2 ] := True;
    Page4ChoixNonExclusif.Values[ 7 ] := True;
```

*Renseignement des cases préalablement cochées.*

```
[...]
end;
```

### 7.3.1.2 Paramètres de création

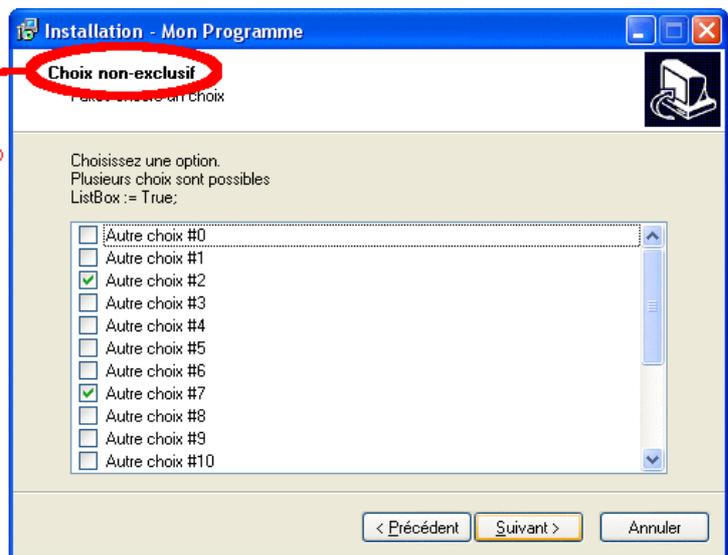
Le prototype de la fonction de la création est:

```

function CreateInputOptionPage(
    const AfterID : Integer;
    const ACaption : String;
    const ADescription : String;
    const ASubCaption : String;
    Exclusive : Boolean;
    ListBox : Boolean )
    : TInputOptionWizardPage;
```

```

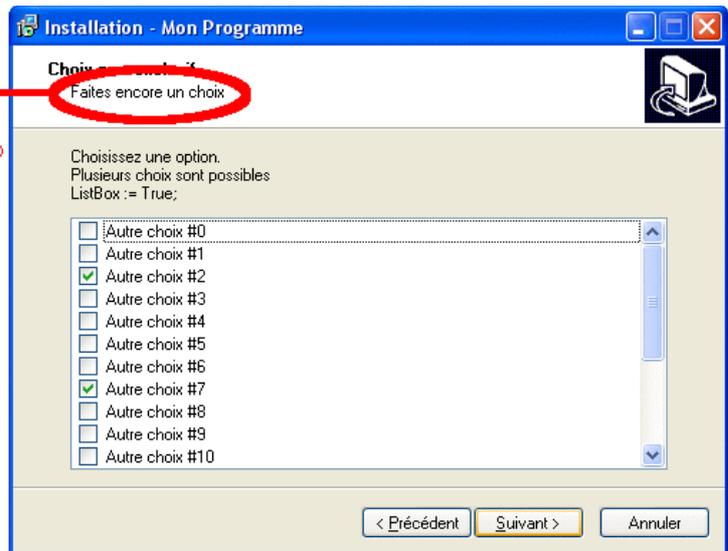
CreateInputOptionPage( const AfterID : Integer;
    const ACaption : String;
    const ADescription : String;
    const ASubCaption : String;
    Exclusive : Boolean;
    ListBox : Boolean )
```



```

CreateInputOptionPage( const AfterID : Integer;
const ACaption : String;
const ADescription : String;
const ASubCaption : String;
Exclusive : Boolean;
ListBox : Boolean )

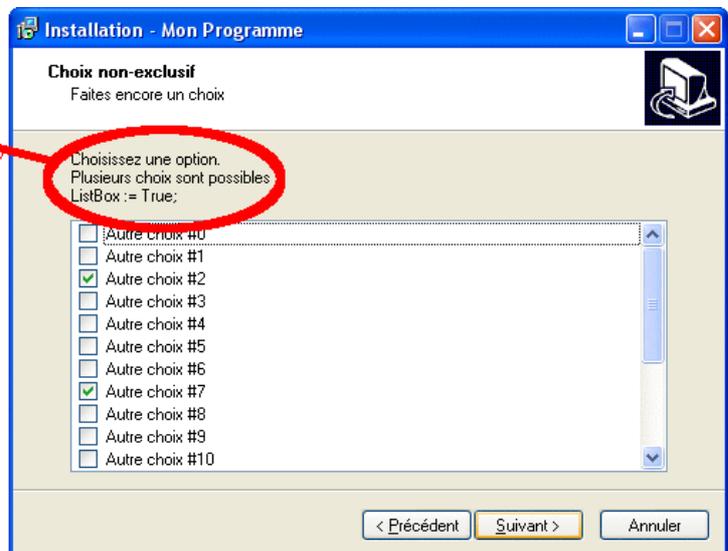
```



```

CreateInputOptionPage( const AfterID : Integer;
const ACaption : String;
const ADescription : String;
const ASubCaption : String;
Exclusive : Boolean;
ListBox : Boolean )

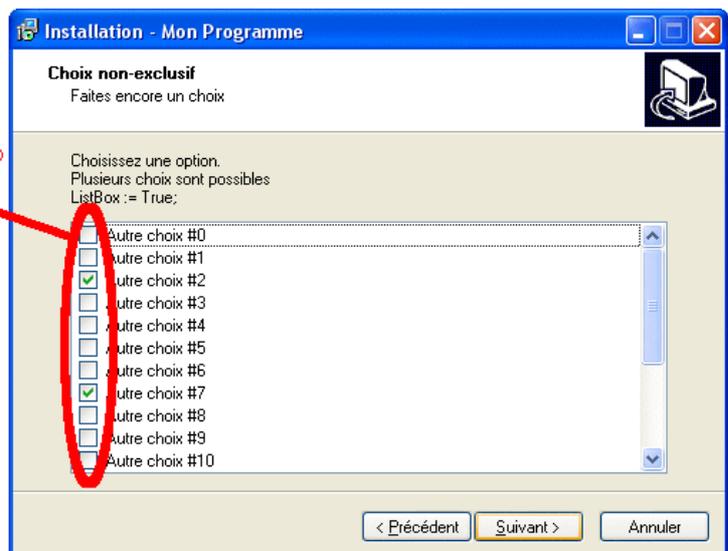
```



```

CreateInputOptionPage( const AfterID : Integer;
const ACaption : String;
const ADescription : String;
const ASubCaption : String;
Exclusive : Boolean;
ListBox : Boolean )

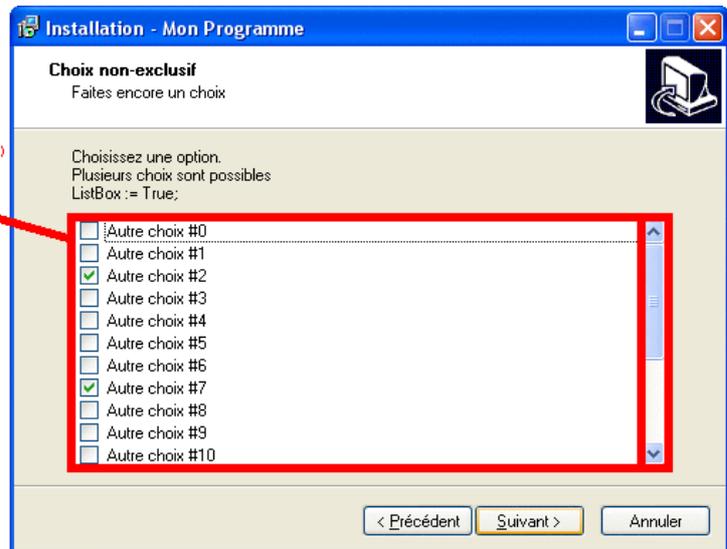
```



```

CreateInputOptionPage( const AfterID : Integer;
                      const ACaption : String;
                      const ADescription : String;
                      const ASubCaption : String;
                      Exclusive : Boolean;
                      ListBox : Boolean )

```



### 7.3.1.3 Exploitation des choix faits

Toutes les actions décrites dans le script et qui sont exécutées au moment de l'installation peuvent être conditionnées par des fonctions Pascal, avec le paramètre « Check »: l'action est effectuée si la fonction Pascal donnée à « Check » retourne « True ».

[Files]

```

Source: MonExemple.iss; \
DestDir: {app}; \
DestName: MonExemple13.iss; \
Check: CheckChoixExclusif( 3 ); \

```

*Le fichier est installé si le 4ème bouton (d'indice 3) est coché.*

[...]

[Code]

[...]

*(\* Retourne True, si le paramètre donné correspond à une case cochée \*)*

```

function CheckChoixExclusif( choix: Integer ) : Boolean;
begin
    Result := Page1ChoixExclusif.Values[ choix ];
end;

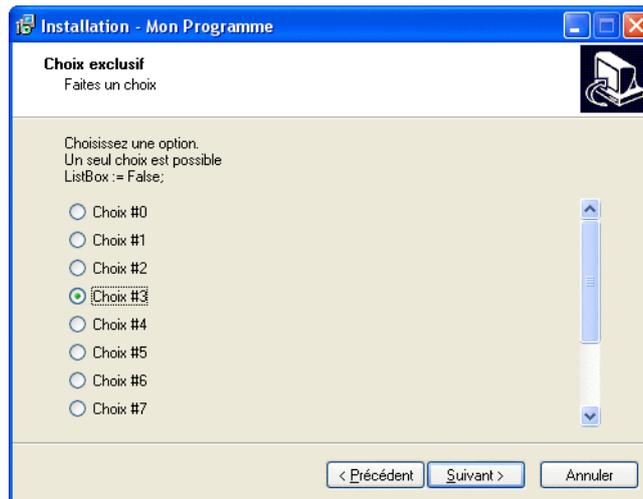
```

### 7.3.1.4 Aspects

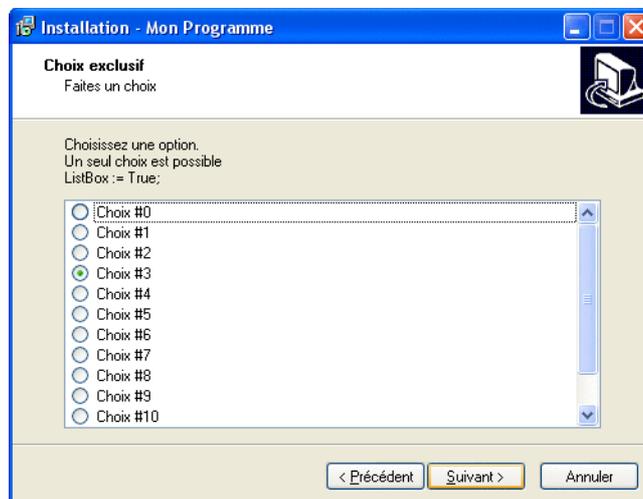
Si les possibilités de choix sont trop nombreuses pour être affichées sur sur seule page, on peut demander à ce que les choix soit présentés dans une *ListBox*, munie d'un ascenseur.

En fait, la création de l'ascenseur est automatique, indépendamment du paramètre « ListBox »: le paramètre influe seulement sur l'aspect de la boîte:

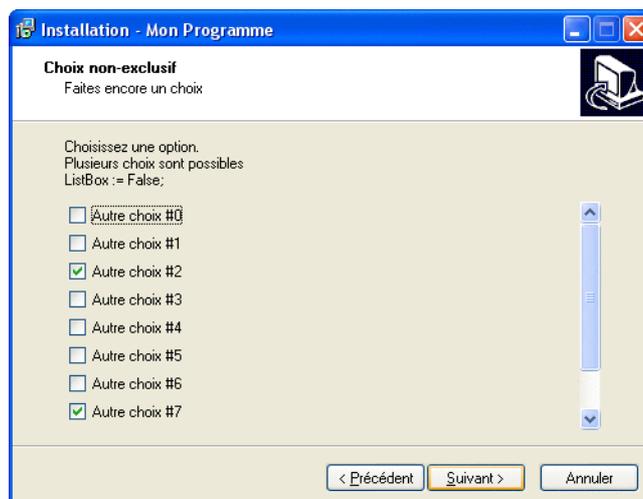
### 7.3.1.4.1 Choix exclusif, sans ListBox



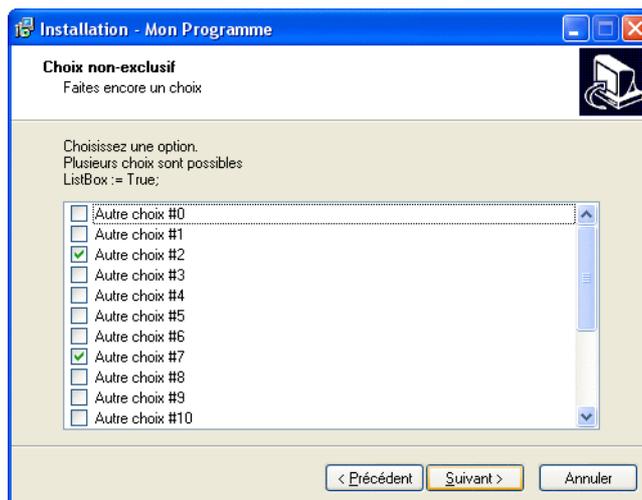
### 7.3.1.4.2 Choix exclusif, avec ListBox



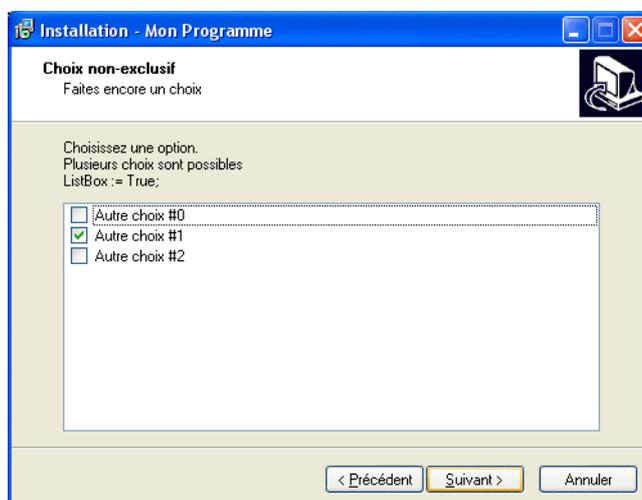
### 7.3.1.4.3 Choix non-exclusif, sans ListBox



#### 7.3.1.4.4 Choix non-exclusif, avec ListBox



#### 7.3.1.4.5 Choix non-exclusif, avec ListBox, sans ascenseur



#### 7.3.1.5 Hiérarchie de la classe TInputOptionWizardPage

Cette hiérarchie (inversée) permet de connaître la totalité des méthodes et propriétés de la page personnalisée, pour des besoins *pointus*.

```
TInputOptionWizardPage = class( TWizardPage )
    function Add( const ACaption: String ): Integer;
    function AddEx( const ACaption: String; const ALevel: Byte; const
AExclusive: Boolean ): Integer;
    property CheckListBox: TNewCheckListBox; read;
    property SelectedValueIndex: Integer; read write;
    property SubCaptionLabel: TNewStaticText; read;
    property Values[Index: Integer]: Boolean; read write;
end;
```

```

TWizardPage = class( TComponent )
    property ID: Integer; read;
    property Caption: String; read write;
    property Description: String; read write;
    property Surface: TNewNotebookPage; read write;
    property SurfaceHeight: Integer; read write;
    property SurfaceWidth: Integer; read write;
    property OnActivate: TWizardPageNotifyEvent; read write;
    property OnBackButtonClick: TWizardPageButtonEvent; read write;
    property OnCancelButtonClick: TWizardPageCancelEvent; read write;
    property OnNextButtonClick: TWizardPageButtonEvent; read write;
    property OnShouldSkipPage: TWizardPageShouldSkipEvent; read write;
end;

TWizardPageNotifyEvent = procedure( Sender: TWizardPage );
TWizardPageButtonEvent = function( Sender: TWizardPage ): Boolean;
TWizardPageCancelEvent = procedure( Sender: TWizardPage; var ACancel, AConfirm: Boolean );
TWizardPageShouldSkipEvent = function( Sender: TWizardPage ): Boolean;

TComponent = class( TPersistent )
    function FindComponent( AName: String ): TComponent;
    constructor Create( AOwner: TComponent );
    property Owner: TComponent; read write;
    procedure DestroyComponents;
    procedure Destroying;
    procedure FreeNotification( AComponent: TComponent );
    procedure InsertComponent( AComponent: TComponent );
    procedure RemoveComponent( AComponent: TComponent );
    property Components[Index: Integer]: TComponent; read;
    property ComponentCount: Integer; read;
    property ComponentIndex: Integer; read write;
    property ComponentState: Byte; read;
    property DesignInfo: Longint; read write;
    property Name: String; read write;
    property Tag: Longint; read write;
end;

```

### 7.3.2 Page de saisie

Classe: TInputQueryWizardPage

Fonction de création: CreateInputQueryPage

## 8 Installations non-assistées

Une installation non-assistée consiste à lancer le setup en lui fournissant les réponses nécessaires, de façon à ne plus demander d'intervention de la part de l'utilisateur.

L'intérêt est d'enchaîner des setups, ou aussi par exemple de fournir à l'utilisateur une possibilité d'installation ne demandant pas de connaissances particulières.

### 8.1 Installation non-assistée Inno Setup

Il est important, lorsqu'on pose une question à l'utilisateur par l'intermédiaire d'une boîte de dialogue dans une fonction Pascal, de retourner la réponse par défaut en mode non-assisté, sans ouvrir la boîte de dialogue. La fonction `WizardSilent` peut être mise à profit pour cela, comme dans l'exemple suivant:

```
function VerifieEstDejaInstalle( AppId: String ) : Boolean;
(*
=====*)
var
    message: String;
begin
    Result := true; (* a priori, on continue l'installation *)

    (* options /SILENT ou /VERYSILENT ?*)
    if not WizardSilent then
    begin
        Result := EstDejaInstalle( AppId );

        if not Result then
        begin
            message := 'It seems that program ''';
            message := message + AppId;
            message := message + '' is not installed.'
                + #13#10'Continue anyway?';

            Result := MsgBox( message,
                mbConfirmation,
                MB_YESNO ) = IDYES;
        end;
    end;
end;
```

## 9 Fichiers .rtf

### 9.1 Généralités

Dans un setup standard, on a la possibilité d'afficher 4 fichiers:

1. Un fichier de licence,
2. Un fichier d'instructions à afficher **avant** l'installation proprement dite,
3. Un fichier d'instructions à afficher **après** l'installation proprement dite,
4. Un fichier *ReadMe*, juste avant de quitter le setup.

Le fichier *ReadMe* n'est en fait pas privilégié: ce n'est qu'une possibilité générique de lancement d'une application ou d'ouverture d'un document. En général, le fichier *ReadMe* sera ouvert avec Notepad ou WordPad selon l'association faite avec le type du fichier sur le PC cible, mais ça peut aussi être un fichier HTML qui sera ouvert avec le butineur par défaut..

Les 3 premiers fichiers cités sont affichés dans les boîtes de dialogue standard du setup.

Ces fichiers peuvent être de deux formats possibles:

1. Format texte (suffixe .txt), créé avec Notepad ou WordPad, sauvegardé au format ANSI (mais **pas au format Unicode**),
2. Format RTF (suffixe .rtf), enregistré **exclusivement avec WordPad**.

Le format texte est simple; il ne permet aucune fantaisie typographique. Comme il sera affiché avec une police à espacement proportionnel, la mise en page est difficilement prévisible.

Le format RTF est un format de traitement de texte quasi-universel; il permet donc de créer des textes mieux présentés et plus lisibles. Son inconvénient est d'être beaucoup plus gros qu'un fichier texte pur, et même d'être largement plus volumineux que le même fichier au format natif des traitements de texte (Word, OpenOffice.org,...). Toutefois, pour les trois fichiers qui nous intéressent, ce ne devrait pas être un problème étant donné que ce sont (ou devraient être) des fichiers de tailles modérées.

[Jordan Russell](#) recommande de produire les fichiers .rtf avec WordPad et non pas avec MS Word, car MS Word inclut de nombreuses balises inutilisables. De plus, un simple fichier .rtf de quelques lignes fait 2 [KiB](#) enregistré avec WordPad, mais passe à 30 KiB enregistré avec MS Word.

Il peut être nécessaire d'utiliser un traitement de texte *lourd* pour créer le fichier .rtf, par exemple WordPad ne sait pas créer des caractères en indice ou en exposant, mais il sait relire et réenregistrer un fichier comportant de tels caractères.

Le fichier ne doit pas être en édition avec OOo ou MS Word pendant qu'Inno Setup le compile. Par contre, cela ne pose pas de problème s'il est ouvert avec WordPad.

### 9.2 Mise en page

Pour créer un fichier .rtf, et avoir une idée assez précise de ce que cela donnera une fois affiché par Inno Setup, le plus simple est d'utiliser OOo Writer<sup>1</sup> pour créer un document vierge dans lequel on

---

<sup>1</sup> Il est sans doute possible de faire la même chose avec MS Word, mais les dimensions des cadres sont peut-être à revoir.

insérera un cadre de **11,23 cm** de large et de **5,27 cm** de haut, et de saisir le texte dans ce cadre.

Si le texte n'entre pas en hauteur, alors redimensionner le cadre à **10,68 cm** de large pour tenir compte de l'apparition d'un ascenseur dans la partie droite du texte, et à la hauteur nécessaire (la hauteur n'est pas critique).

Enregistrer ensuite le document au format natif d'OOo, c-à-d au format .odt pour OOo version 2.

Toujours sous OOo, réenregistrer le document au format .rtf, et le fermer.

Ouvrir ce fichier .rtf avec WordPad, et le réenregistrer afin de ne conserver que les attributs de texte les plus simples, interprétables par Inno Setup.

### **9.3 Limitations**

- Inno Setup **n'affiche pas les images incrustées** dans un fichier .rtf.
- Il est possible d'insérer des URL cliquables, mais le texte normalement affiché **par dessus l'URL** est affiché **avant** celle-ci (si le texte est différent de l'URL; sinon il n'est affiché qu'une fois), perturbant ainsi la mise en page.
- Il ne semble pas possible d'insérer un lien hypertexte vers un fichier local en utilisant OOo + WordPad, ni à l'intérieur du fichier.

## 10 Mode « ligne de commande »

Même au 21<sup>ème</sup> siècle il est encore utile de pouvoir exécuter un programme sans avoir à cliquer une seule fois, principalement pour automatiser un processus.

### 10.1 *Compilation non-assistée du setup*

Par principe, ISTool est 100% interactif. Seul le compilateur Inno Setup nous intéresse.

Les options sont décrites en détails dans l'aide en ligne au chapitre « Inno Setup Preprocessor / Extended Command Line Compiler ».

La ligne de commande ne demande que peu d'arguments et d'options:

```
iscc script.iss /Q /dCONSTANTE=valeur
```

L'option « /Q » (**Q**uiet) n'est pas indispensable: elle demande une compilation silencieuse; dans ce mode, seuls les messages d'erreurs sont affichés.

L'option /d est similaire à l'option -D ou /D des compilateurs « C »: elle permet de définir une constante dans la ligne de commande de la même façon qu'avec #define.

A noter, que l'aide en ligne « [Other Information / Command Line Compiler Execution](#) » donne comme syntaxe:

```
iscc /Q script.iss
```

Cette syntaxe est celle de la version d'ISCC qui n'inclut pas le préprocesseur.

Rappel: ISCC n'est pas dans le PATH. Il faut donc:

- soit ajouter, au moins temporairement, le répertoire Inno Setup dans le PATH

```
set PATH=%PATH%;"C:\Program Files\Inno Setup 5"
```

- soit spécifier le chemin complet d'ISCC, comme par exemple:

```
"C:\Program Files\Inno Setup 5\ISCC.exe" monexemple.iss /Q
```

Les guillemets sont nécessaires, à moins de spécifier le chemin au format 8+3 hérité de MS-DOS:

```
C:\PROGRA~1\INNOSE~1\ISCC monexemple.iss /Q
```

Ce chemin court est susceptible de changer d'une installation à l'autre d'IS.

### 10.2 *Installation non-assistée d'un setup*

Un setup peut avoir à lancer lui-même d'autres setups, ceux-ci étant dans le cas général non-maîtrisés: mise à jour du système, installation d'un pilote de périphérique,...

Ce chapitre tente de rassembler le plus d'éléments sur les options que l'on peut donner aux setups natures diverses, avec le but d'exécuter ces setups de manière non assistée.

Il va sans dire que les paramètres décrits ici ne sont qu'indicatifs, car ils peuvent varier d'une version à l'autre du programme installateur. De plus, il faut que les concepteurs du setup aient prévu l'installation non-assistée: il ne sert à rien de passer l'option /SILENT à un setup Inno Setup, si le développeur qui a conçu le setup ouvre inconditionnellement une boîte de dialogue dans sa section [Code]...

En général, les options d'installation non-assistée sont de la forme /SILENT, /S, -s, /quiet, /Q, /q ou -q.

Le site du projet SourceForge « Unattended, A Windows deployment system » est une mine d'informations concernant les installations non-assistées.

## 10.2.1 Inno Setup

Ce chapitre n'a pour but que de donner une idée des options disponibles. Pour le détail, on se reportera avec profit à la page « [Setup Command Line Parameters](#) » de l'aide en ligne.

<i>Option</i>	<i>Paramètre [Setup]</i>	<i>Description succincte</i>
/SP-	DisableStartupPrompt, page 41.	Supprime la boîte « Cet assistant va installer... » (voir page 26). Assez bizarrement, cette boîte est quand même affichée avec les options /SILENT et /VERYSILENT.
/SILENT	Néant.	Ces options permettent une installation ne demandant pas d'intervention, hormis l'acquiescement de la boîte de dialogue « Cet assistant va installer... » qui doit être supprimée avec l'option /SP- dans la ligne de commande, ou DisableStartupPrompt dans le script. Avec l'option/SILENT, la page d'avancement du setup Installation en cours, page 33 est quand même affichée, mais ne demande pas d'intervention. Avec l'option /VERYSILENT, même cette page n'est pas affichée.
/VERYSILENT		
/SUPPRESSMSGBOXES		Associée à /SILENT ou /VERYSILENT, cette option répond automatiquement avec les réponses par défaut, aux questions du genre « Voulez-vous remplacer le fichier existant? » qui pourraient être posées.
/LOG	SetupLogging, page 45.	Création d'un journal de l'installation.
/NORESTART		Empêche Inno Setup de rebooter le PC s'il le juge nécessaire. Peut être utile pour enchaîner plusieurs setups, et ne rebooter qu'à la fin.
/RESTARTEXITCODE		En complément /NORESTART, cette option de permet au setup de retourner un compte rendu d'exécution pour indiquer si un reboot est nécessaire.
/DIR		Spécifie le chemin donné par défaut dans la boîte de dialogue Dossier de destination, page 30.

<i>Option</i>	<i>Paramètre [Setup]</i>	<i>Description succincte</i>
/TYPE		Spécifie le type d'installation par défaut (boîte de dialogue Composants à installer, page 31).
/COMPONENTS		Liste exhaustivement les composants à installer. Cette option est contradictoire avec l'option /TYPE.
/TASKS		Liste exhaustivement les composants et tâches à exécuter.

Il est toujours possible de définir des options personnalisées, exploitées soit par une routine Pascal, soit par la {constante} « {param:ParamName|DefaultValue} » (page 63). Ceci explique d'Inno Setup ne se plaindre pas, si une option mal orthographiée est passée dans sa ligne de commande.

## 10.2.2 Microsoft Windows Installer (MSI)

Ecran d'aide de msiexec.exe, lorsqu'il est lancé sans arguments:

Windows © Installer. V 3.01.4000.1823

```
msiexec /Option <Paramètre requis> [Paramètre facultatif]
```

Options d'installation

```
</package | /i> <Produit.msi>
```

Installe ou configure un produit

```
/a <Produit.msi>
```

Installation administrative - Installe un produit sur le réseau

```
/j<u|m> <Produit.msi> [/t <Liste de transformations>] [/g <ID de langue>]
```

Publie un produit - m pour tous les utilisateurs, u pour

l'utilisateur actuel

```
</uninstall | /x> <Produit.msi | Code produit>
```

Désinstalle le produit

Options d'affichage

```
/quiet
```

Mode silencieux, aucune interaction avec l'utilisateur

```
/passive
```

Mode automatique - barre de progression uniquement

```
/q[n|b|r|f]
```

Définit le niveau d'interface utilisateur

n - Pas d'interface utilisateur

b - IU de base

r - IU réduite

f - IU complète (option par défaut)

```
/help
```

Affiche des informations

Options de redémarrage

```
/norestart
```

Ne pas redémarrer à la fin de l'installation

```
/promptrestart
```

Proposer à l'utilisateur de redémarrer si nécessaire

```
/forcerestart
```

Toujours redémarrer l'ordinateur après l'installation

#### Options de journalisation

```
/l[i|w|e|a|r|u|c|m|o|p|v|x|+|!|*] <fichier journal>
  i - Messages d'état
  w - Avertissements récupérables
  e - Tous les messages d'erreur
  a - Démarrage des actions
  r - Enregistrements spécifiques à certaines actions
  u - Requêtes de l'utilisateur
  c - Paramètres initiaux de l'interface utilisateur
  m - Mémoire insuffisante ou informations de sortie irrécupérables
  o - Message d'espace disque insuffisant
  p - Propriétés du terminal
  v - Mode documenté
  x - Autres informations de débogage
  + - Ajouter au fichier journal existant
  ! - Forcer l'écriture de chaque ligne dans le journal
  * - Écrire toutes les informations, sauf les options v et x

/log <Fichier journal>
  Identique à /l* <Fichier journal>
```

#### Options MAJ

```
/update <Update1.msp>[;Update2.msp]
  Applique les mises à jour

/uninstall <Guid du code correctif>[;Update2.msp] /package <Product.msi |
Code produit>
```

Supprimer les mises à jour d'un produit

#### Options de réparation

```
/f[p|e|c|m|s|o|d|a|u|v] <Product.msi | Code produit>
```

Réparer un produit

```
  p - uniquement si un fichier est manquant
  o - si un fichier est manquant ou si une version antérieure est
installée (option par défaut)
  e - si un fichier est manquant ou si une version égale ou antérieure
est installée
  d - si un fichier est manquant ou si une version différente est
installée
  c - si un fichier est manquant ou si la somme de contrôle ne
correspond pas à la valeur calculée
  a - force la réinstallation de tous les fichiers
  u - toutes les entrées de Registre requises et spécifiques à
l'utilisateur (option par défaut)
  m - toutes les entrées de Registre requises et spécifiques à
l'ordinateur (option par défaut)
  s - tous les raccourcis existants (option par défaut)
  v - s'exécute depuis la source et remet en mémoire cache le package
```

#### local

#### Définition des propriétés publiques

```
[PROPERTY=valeur de la propriété]
```

Consultez le Kit de développement Windows® Installer pour une aide détaillée concernant la syntaxe de la ligne de commande.

Copyright © Microsoft Corporation. Tous droits réservés.

Certaines parties de ce logiciel sont basées sur des documents créés par la société indépendante JPEG Group.

Les options les plus intéressantes sont « /quiet » et « /norestart ».

### 10.2.3 InstallShield

Les setups InstallShield récents offrent un mode « fichier de réponse ». Dans un premier temps, il faut exécuter le setup en mode « enregistrement du fichier de réponse ». Le fichier de réponse est un

fichier texte nommé « setup.iss » par défaut. Sa structure n'a rien à voir avec celle d'un script Inno Setup bien qu'il soit également doté d'un suffixe « .iss ».

Extraits de la page « Q105473: INFO: Setup.exe Command-Line Parameters » de Macrovision (qui a repris InstallShield):

### **Silent Installations**

[...]

#### ***/r : Record mode (InstallScript and InstallScript MSI projects only)***

*In order to run a Standard-project installation program in silent mode, one must first run Setup.exe with the /r switch to generate a response file, which stores information about the data entered and options selected by the user at run time.*

*Running a Standard installation program with the command Setup.exe /r displays all the run-time dialog boxes, and stores the data in a file called Setup.iss, generated inside the system's Windows folder. To specify an alternative response file name and location, use the /f1 argument, described below.*

*Basic MSI projects do not generate or use a response file for silent installations.*

#### ***/s : Silent mode***

*For a Standard project, the command Setup.exe /s runs the installation in silent mode, by default based on the responses contained in a response file called Setup.iss in the same directory (response files are generated by running Setup.exe with the /r option). To specify an alternative file name or location of the response file, use the /f1 option, described below.*

*The command Setup.exe /s also suppresses the Setup.exe initialization window for a Basic MSI installation program, but does not read a response file. To run a Basic MSI product silently, run the command line Setup.exe /s /v"/qn. (To specify the values of public properties for a silent Basic MSI installation, you can use a command, such as Setup.exe /s /v"/qn INSTALLDIR=D:\Destination". If you would like to use a long file name, use the following example format: Setup.exe /s /v"/qn INSTALLDIR="D:\Program Files\Macromedia\JRun")*

#### ***/f1 : Specify alternative response file name (InstallScript and InstallScript MSI projects only)***

*Using the /f1 argument allows one to specify where the response file is (or where it should be generated) and what its name is, as in Setup.exe /s /f1"C:\Temp\Setup.iss". The /f1 switch is available both when creating a response file (with the /r option) and when using a response file (with the /s option).*

#### ***/f2 : Specify alternative log file name (InstallScript and InstallScript MSI projects only)***

*When running a Standard-project installation program in silent mode (using the /s argument), the log file Setup.log is generated by default in the same directory as the response file. The /f2 argument allows one to specify an alternative log file location and file name, as in Setup.exe /s /f2"C:\Setup.log".*

La page cite une option /SMS, pour dire qu'elle n'est plus nécessaire, mais cette option peut être utilisée avec d'anciens setups, pour leur demander de ne rendre la main qu'à la fin, c'est-à-dire exécuter le setup en mode synchrone.

## **10.2.4 NSIS**

Le détail des options est donné dans la page « Command Line Usage » de NSIS, dont est extrait:

*/S runs the installer or uninstaller silently. [...]*

*/D sets the default installation directory ([\\$INSTDIR](#)), overriding [InstallDir](#) and [InstallDirRegKey](#). It must be the last parameter used in the command line and must not contain any quotes, even if the path contains spaces. Only absolute paths are supported.*

### **10.2.5 Wise Installer**

Selon l'article « What are the command line parameters available for WiseScript? »:

*/S: Silent install.*

### **10.2.6 Windows Hotfix**

La description des hotfixes est donnée dans l'article « Microsoft Windows XP Hotfix Installation and Deployment Guide », en particulier les options:

*/F Forces other applications to close at shutdown.*

*/N Does not back up files for removing hotfixes.*

*/Z Does not restart the computer after the installation is completed.*

*/Q Uses quiet mode; no user interaction is required.*

*/U Uses unattended Setup mode.*

*/L Lists installed hotfixes.*

### **10.2.7 Setups exclusivement interactifs**

Si le setup que l'on cherche à installer n'a pas été prévu pour être exécuté sans assistance, une piste à creuser est de simuler la frappe de touches (ou clics de souris) par un script VBScript. La description de l'API sur MSDN commence à la page « Windows Script Host ».

L'article « Guide d'équipe technique de gestion des applications » illustre cette méthode.

L'inconvénient de cette méthode est que le setup doit avoir un comportement déterministe...

## 11 Le registre Windows

Il est difficile de faire un setup un peu évolué, sans avoir à manipuler le registre de Windows.

La terminologie du registre est précise mais n'est pas intuitive: ce qu'on appelle « clef » dans le contexte du registre Windows est l'équivalent d'un répertoire dans un système de fichiers. L'équivalent du fichier est une « valeur ». La valeur possède un nom, des données,...

### 11.1 Les moyens d'écriture

Les moyens d'écrire dans le registre ne manquent pas.

#### 11.1.1 Section [Registry] d'Inno Setup

Cette section constitue le moyen le plus simple pour écrire dans le registre. Elle permet des manipulations du registre conditionnées par les moyens standard d'Inno Setup, c-à-d par les composants et / ou tâches sélectionnés, par la version de Windows, la langue d'installation et par une fonction Pascal de type `Check()`.

De nombreuses options permettent de manipuler le registre avec prudence, comme par exemple `createvalueifdoesntexist`, `preserverestringtype` ou `uninsdeletekeyifempty`.

L'intégration à Inno Setup permet une destruction optionnelle et éventuellement prudente, des clefs et valeurs lors de la désinstallation.

#### 11.1.2 Commande externe `reg.exe`

Cette commande, disponible depuis Windows 2000, permet une manipulation très complète du registre en fonction des arguments donnés dans la ligne de commande: écriture, lecture, recherche, import, export,...

Par rapport à `regedit.exe`, cette commande présente l'avantage de retourner un compte rendu, et d'être bien documentée.

Pour obtenir la liste des commandes et options applicables, il suffit d'entrer la commande « `reg /?` » dans une boîte DOS.

#### 11.1.3 Commande externe `regedit.exe`

La seule possibilité offerte par cette commande est l'import d'un fichier, ce que permet également `reg.exe`, mais `regedit.exe` a l'avantage d'être utilisable sous NT4, alors que `reg.exe` demande une version de Windows supérieure ou égale à Windows 2000.

Le fichier aura avantage à être au format NT4, c-à-d ASCII (ANSI), par opposition au format version 5 (>= Windows 2000) qui est Unicode. L'ASCII offre plus de possibilités de manipulation par des éditeurs et autres outils *Unix-like*.

☞ Lorsqu'on utilise `regedit.exe` en mode interactif sous XP pour exporter une branche du registre, le **format par défaut est Unicode**. Le format ASCII est du type « Win9x/NT4 ».

Il en est de même, lorsqu'on exporte une branche du registre en ligne de commande, avec l'option standard d'export « `/E` ». Il existe cependant une option non-documentée « `/A` » (comme « ANSI ») identique à l'option « `/E` », mais qui exporte

au format REGEDIT4.

Un fichier au format ASCII/NT4 est reconnaissable par sa signature (1ère ligne):

```
REGEDIT4
```

Un fichier Unicode a le même aspect (si l'éditeur utilisé l'a effectivement reconnu comme étant Unicode), mais la signature est:

```
Windows Registry Editor Version 5.00
```

Dans le fichier importé, on peut bien évidemment créer / écrire des valeurs, mais également détruire des valeurs et des clefs.

La syntaxe pour importer un fichier est:

« `regedit /s fichier.reg` », l'option « `/s` » demandant le mode *silencieux*, car cet outil est avant tout interactif.

### 11.1.4 Pascal

Pascal permet des actions plus fines que les moyens précédents, qui se contentent d'écrire ou de supprimer des branches du registre. Il est ainsi possible avec Pascal de modifier une valeur du registre, par exemple ajouter un chemin dans la variable d'environnement `PATH`.

Pascal est le seul moyen pratique pour lire le registre. On peut par exemple écrire une fonction `Check()` qui conditionne une action quelconque du setup en fonction d'une donnée du registre.

## 11.2 Quelques clefs intéressantes

Les exemples sont donnés en utilisant la syntaxe des fichiers d'export `reg.exe / regedit.exe`.

### 11.2.1 Uninstall

Sous la clef `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall` on trouve toutes les informations nécessaires à la désinstallation des programmes, et donc des informations relatives à l'installation courante.

A noter que si l'application a été installée sous le compte d'un utilisateur non-administrateur (rare, mais possible), les informations sont sous la branche `HKCU` au lieu de `HKLM`.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Inno Setup 5_is1]
```

```
"DisplayName"="Inno Setup QuickStart Pack version 5.2.2"
```

```
"UninstallString"="\"C:\\Program Files\\Inno Setup 5\\unins000.exe\""
```

```
"QuietUninstallString"="\"C:\\Program Files\\Inno Setup 5\\unins000.exe\" /SILENT"
```

La suppression de la valeur `"DisplayName"` fait disparaître le programme du panneau de contrôle. Il reste cependant désinstallable en exécutant la commande présente dans `"UninstallString"` ou `"QuietUninstallString"`.

La valeur `DWORD` « `NoRemove` » égale à 1 empêche la désinstallation depuis le panneau de contrôle, bien l'application y soit toujours présente. Ceci peut être utilisé, lorsqu'un setup *maître* enchaîne plusieurs setups secondaires: on peut désirer faire figurer les applications secondaires dans « Ajout / Suppression de programmes », sans autoriser les désinstallations individuelles, obligeant à passer par la désinstallation du setup *maître*..

## 11.2.2 RunOnce

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

## 12 Trucs et astuces

Ce paragraphe recense quelques uns des pièges dans lequel l'auteur est tombé.

- ```
#define MA_MACRO BlaBla
[Files]
Source: fichier.txt; DestDir: {app}\(#MA_MACRO);
```

Le fichier est installé dans un répertoire nommé littéralement “(#MA\_MACRO)”!  
La cause est l'emploi intempestif d'une **parenthèse** gauche à la place d'une **acolade**...

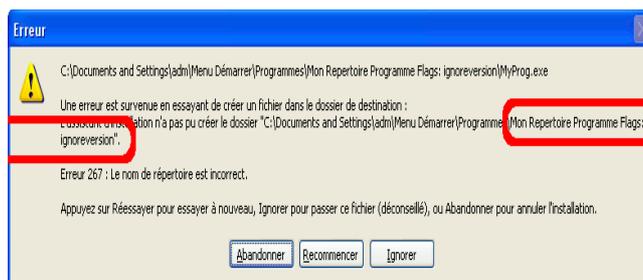
- La fonction Pascal **ExtractFilePath()** n'extrait pas le répertoire comme son nom le laisse supposer, mais **supprime** la dernière partie du chemin donné en paramètre. Si le paramètre est un répertoire, alors la fonction donne le répertoire de niveau supérieur.
- La fonction Pascal **GetShortName()** ne fonctionne correctement que **si le fichier existe**. Attention donc à ne pas utiliser cette fonction avant que le fichier ne soit créé, ou bien après qu'il ait été détruit.
- Le préprocesseur admet la redéfinition d'un symbole avec une valeur différente, sans produire d'avertissement.
- La fonction **FileExists()** du **préprocesseur** ne retourne “vrai” que pour des “*plain files*”, et pas pour des répertoires. La fonction Pascal de même nom a la même caractéristique, mais il existe aussi les fonctions Pascal **DirExists()** et **FileOrDirExists()**.
- La priorité relative des opérateurs Pascal **not** et **=** n'est pas intuitive. Voir en page 71.
- ISTool reformate le fichier script.
- Les boîtes de dialogues ouvertes par les fonctions Pascal devraient systématiquement être conditionnée par le mode silencieux d'exécution du setup (fonction Pascal **WizardSilent()**).
- Des mesures doivent être prises pour que les commandes *DOS* exécutées directement ou indirectement dans la section **[Run]** en mode caché (flag `runhidden`) ne posent pas de questions de confirmation: « `del /f` », « `xcopy /y` », « `regedit /s` », « `reg /f` »,... sinon le setup reste coincé en l'attente d'une réponse à une question invisible.
- Lorsqu'on désigne un fichier `.ini` en relatif, il faut préfixer le chemin par « `.\` ».
- Il n'est pas possible d'utiliser la DLL `zlib` utilisée par IS pour dézipper un fichier ZIP classique: `zlib` travaille sur des flots, par opposition à des archives.
- Lorsqu'on mélange des fichiers livrés dans la section `[Files]`, et des commandes dans la section `[Run]`, la section `[Run]` est exécutée après `[Files]`, et peut potentiellement surcharger ce qui a été fait dans `[Files]`.
- Une erreur souvent non-déTECTABLE par le compilateur, est l'absence de point-virgule dans une section autre que `[Code]`, par exemple:

```
[Files]
Source: MyProg.exe; DestDir: {app} Flags: ignoreversion
```

Point-virgule manquant!

Dans cet exemple, la compilation se passe correctement, mais lors de l'exécution du setup, une erreur se produira parce qu'on cherchera à installer le fichier dans un répertoire nommé

littéralement « Mon Répertoire Programme Flags: ignoreversion » :



Dans ce cas précis, l'erreur ne passe pas inaperçue parce que cela a produit un nom de fichier invalide (contenant un caractère « deux-points »), mais un point-virgule manquant peut se solder par l'absorption discrète d'un paramètre.

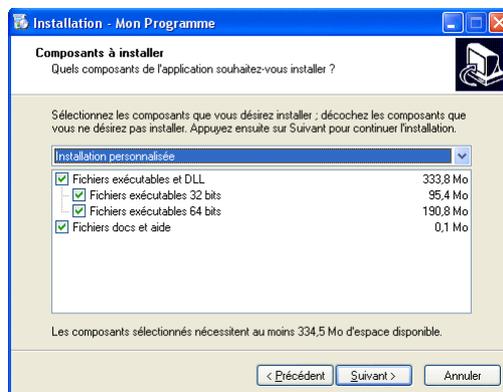
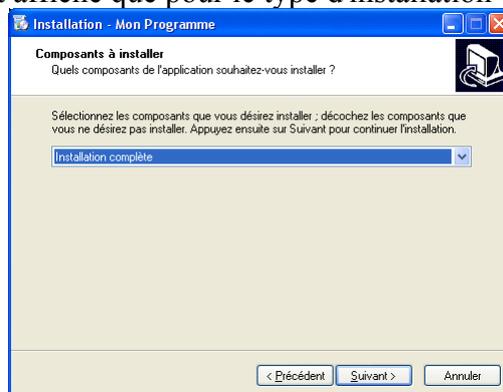
Inversement, un point-virgule superflu en fin de ligne est ignoré.

- Il est parfois nécessaire dans la section [File] de livrer un fichier dans le répertoire temporaire {tmp}, et ensuite d'exécuter ou d'utiliser ce fichier dans la section [Run]. L'exécution est probablement conditionnée par la sélection d'un composant ou d'une tâche. Par économie de moyen, on peut être tenté de livrer inconditionnellement le fichier dans {tmp}, sans le conditionner par composant ou tâche. Dans ce cas, il faut savoir que la taille du fichier livré dans {tmp} est comptabilisée, même si le composant ou la tâche qui l'utilise n'est pas sélectionné.
- Comme il est possible de définir des options personnalisées passées dans la ligne de commande, Inno Setup ne peut pas toujours détecter les erreurs de syntaxe des options standard.

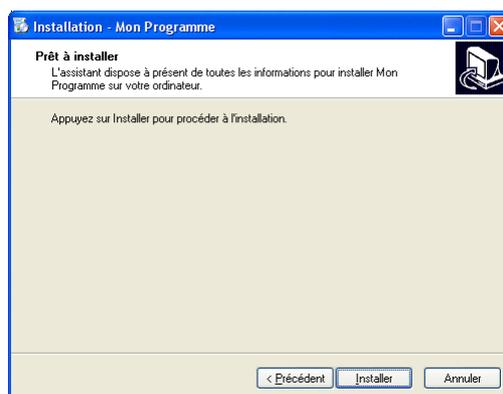
## 13 Annexes

### 13.1 Effet de AlwaysShowComponentsList

Le détail des composants n'est affiché que pour le type d'installation « personnalisée ».



### 13.2 Effet de DisableReadyMemo



### 13.3 Setups splittés

Inno Setup autorise le découpage du setup monolithique, en plusieurs fichiers. Cette possibilité est mise en oeuvre avec les paramètres DiskSpanning et DiskSliceSize (page 40).

Cette option est prévue a priori pour permettre l'utilisation ou le transport du setup sur un support de taille limitée comme une disquette ou un CD-ROM.

L'intérêt de cette possibilité est maintenant limité, mais s'il est nécessaire d'y avoir recours, il faut

savoir que:

- Seul le premier fichier du setup (le .exe) est identifié en version. Les autres fichiers (.bin) ne possèdent pas d'onglet de version. La date des fichiers n'est pas une identification fiable, celle-ci se perdant très facilement.
- Si un des fichiers .bin est malencontreusement remplacé par un autre fichier de même nom (d'une autre version du setup), le résultat peut dans le meilleur des cas être un compte rendu tardif « fichier corrompu », mais il est probable que cette erreur puisse passer inaperçue.

### **13.4 Compression**

Inno Setup compresse les fichiers qu'il installe. IS propose plusieurs algorithmes de compression, plusieurs taux de compression, et une option.

Les algorithmes proposés sont « zip », « bzip » et « lzma », respectivement utilisés par les programmes WinZip (entre autres), bzip2 sous Linux, et 7-zip.

Jordan recommande « lzma » comme étant le meilleur compromis, à une exception près: si les fichiers embarqués dont le setup ne se compressent pas ou peu, la compression zip donne de meilleurs résultats.

Les fichiers qui ne compriment pas sont ceux qui par nature comportent des données déjà comprimées (.jpg, .mp3, .zip, .odt), ou bien sont chiffrés.

Pour un gros setup, le temps de compression peut être significatif (plusieurs heures).

Les mesures ont été faites en créant un setup pour l'arborescence de quelques paquets GnuWin32 (<http://gnuwin32.sourceforge.net/>), comprenant les exécutables, la documentation et les sources.

### **13.5 Principes à respecter**

Un setup qui pose des problèmes d'installation donne une impression d'amateurisme.

Il est possible d'éviter les pièges les plus grossiers, en faisant quelques hypothèses pessimistes:

- la personne qui va procéder à l'installation n'est pas experte en informatique,
- l'utilisateur ne lira pas les instructions qui lui sont données, surtout si elles sont longues, absconses ou bien si elles noient les informations techniques dans un flot d'informations juridiques,
- le PC n'a pas d'accès Internet,
- le PC possède une résolution graphique de 640x480,
- le PC n'a pas les dernières mises à jour de Windows,
- le PC n'a pas de souris,
- le PC n'a pas de carte son,
- si le setup n'est pas auto-suffisant, alors il manquera toujours un élément.

Si le setup a besoin d'une caractéristique particulière, il doit tester la disponibilité de cette caractéristique de la manière la plus automatisée possible, en posant le minimum de questions à l'utilisateur et le cas échéant donner un diagnostic clair, le plus tôt possible.

Entre autres, contrôler:

- la version de Windows,
- la langue de Windows,
- les droits de l'utilisateur courant
- l'existence d'une version précédente des fichiers
- les programmes en cours d'exécution

Les questions posées à l'utilisateur ainsi que les directives qui lui sont données doivent utiliser son vocabulaire, et non pas le jargon du développeur.

A l'inverse, le setup doit pouvoir être utilisé au mieux par un expert:

- mode d'installation non-assistée (voir page [81](#)),
- découpage fin du setup en composants et tâches,
- un rappel des options qu'il est possible de passer dans la ligne de commande doit être donné lorsque le setup est invoqué avec les options `/?`, `/help`, `-h`, `-help` ou `--help`.

Idéalement, le setup devrait pouvoir s'installer sur un PC Pentium 166 sous Windows NT4 SP4, avec la version d'Internet Explorer d'origine (donc avec de nombreuses DLLs obsolètes).

Bien entendu, ces conseils sont à nuancer en fonction du programme à installer.

### 13.6 Check-lists

Dans le source, avant compilation, vérifier:

- Le fichier « LisezMoi » est actualisé.
- L'identification de version est actualisée.
- Chaque paramètre de chaque section (sauf [Setup]) est terminé par un point-virgule.
- L'absence des fichiers référencés par {src} est envisagée.
- L'édition de toute boîte de dialogue dans la section [Code] envisage le mode non-assisté.

Lors du test du setup, vérifier:

- Exécution en mode non-assisté.
- Tentative d'exécution sous un compte non-administrateur.
- Installation sur un PC vierge.
- Installation d'un composant sélectionnable à la fois.

### 13.7 Références

| <i>Réf.</i> | <i>Site</i>                                                                             | <i>Lng</i> | <i>Description</i>                          |
|-------------|-----------------------------------------------------------------------------------------|------------|---------------------------------------------|
| [IS]        | <a href="http://www.jrsoftware.org/">http://www.jrsoftware.org/</a>                     | en         | Page d'accueil du site de Jordan.           |
|             | <a href="http://www.jrsoftware.org/isinfo.php">http://www.jrsoftware.org/isinfo.php</a> | en         | Page principale d'IS sur le site de Jordan. |

| <i>Réf.</i> | <i>Site</i>                                                                                                                   | <i>Lng</i> | <i>Description</i>                                                                                                                                                                                                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | <a href="http://www.jrsoftware.org/iskb.php">http://www.jrsoftware.org/iskb.php</a>                                           | en         | Base de connaissance, FAQ.                                                                                                                                                                                                                                                                         |
|             | <a href="http://www.jrsoftware.org/newsgroups.php#search">http://www.jrsoftware.org/newsgroups.php#search</a>                 | en         | Archives des forums.                                                                                                                                                                                                                                                                               |
| [JRDN]      | <a href="http://news.jrsoftware.org/news/innosetup/msg63485.html">http://news.jrsoftware.org/news/innosetup/msg63485.html</a> | en         | Recommandation de <a href="#">Jordan</a> de n'utiliser que WordPad:<br>[...] <i>But did you actually *save* the file in WordPad? If not, try that.</i><br><br><i>Other editors like Word tend to put a bunch of extraneous tags in the file, which can confuse riched*.dll sometimes.</i><br>[...] |
|             | <a href="http://msdn2.microsoft.com/en-us/library/ms954115.aspx">http://msdn2.microsoft.com/en-us/library/ms954115.aspx</a>   | en         | Application Specification for Microsoft Windows 2000 for Desktop Applications                                                                                                                                                                                                                      |
| [REMOB]     | <a href="http://www.remobjects.com/">http://www.remobjects.com/</a>                                                           | en         |                                                                                                                                                                                                                                                                                                    |
|             | <a href="http://www.vincenzo.net/isxkb/index.php?title=Main_Page">http://www.vincenzo.net/isxkb/index.php?title=Main_Page</a> | en         | « Inno Setup Extensions Knowledge Base ».                                                                                                                                                                                                                                                          |
| [MIRALITY]  | <a href="http://www.mirality.co.nz/inno/tips.php">http://www.mirality.co.nz/inno/tips.php</a>                                 | en         | « Inno Setup: Tips & Tricks »: condensé de Pascal pour Inno Setup.                                                                                                                                                                                                                                 |
| [OOo]       | <a href="http://fr.openoffice.org/">http://fr.openoffice.org/</a>                                                             | fr         | Site officiel.                                                                                                                                                                                                                                                                                     |
|             | <a href="http://www.openofficeportable.org/">http://www.openofficeportable.org/</a>                                           | fr         | Versions portables, c'est-à-dire ne nécessitant pas d'installation, et pouvant être exécuté à partir d'un CD-ROM ou d'une clef USB.                                                                                                                                                                |
|             | <a href="http://portableapps.com/apps/office/openoffice_portable">http://portableapps.com/apps/office/openoffice_portable</a> | en         |                                                                                                                                                                                                                                                                                                    |

| <i>Réf.</i> | <i>Site</i>                                                                                                                                                                                                                                                                         | <i>Lng</i> | <i>Description</i>                                                                                                                                                                |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | <a href="http://knowledge.macrovision.com/selfservice/microsites/search.do?cmd=displayKC&amp;docType=kc&amp;externalId=Q105473&amp;sliceId=">http://knowledge.macrovision.com/selfservice/microsites/search.do?cmd=displayKC&amp;docType=kc&amp;externalId=Q105473&amp;sliceId=</a> | en         | Article « Q105473: INFO: Setup.exe Command-Line Parameters », détaillant les paramètres pouvant être donnés dans la ligne de commande à un setup InstallShield.                   |
|             | <a href="http://www.microsoft.com/france/technet/desktopdeployment/bdd/2007/appmgmt_6.msp">http://www.microsoft.com/france/technet/desktopdeployment/bdd/2007/appmgmt_6.msp</a>                                                                                                     | fr         | Article « Guide d'équipe technique de gestion des applications » de Microsoft TechNet.                                                                                            |
| [MS310516]  | <a href="http://support.microsoft.com/kb/310516">http://support.microsoft.com/kb/310516</a>                                                                                                                                                                                         | fr,<br>en  | Article numéro 310516 de MSDN: « How to add, modify, or delete registry subkeys and values by using a registration entries (.reg) file », spécifiant le format des fichiers .reg. |
|             | <a href="http://en.wikipedia.org/wiki/Windows_Registry">http://en.wikipedia.org/wiki/Windows_Registry</a>                                                                                                                                                                           | en         | Article de Wikipedia de présentation du registre « Windows Registry ».                                                                                                            |
|             | <a href="http://nsis.sourceforge.net/Docs/Chapter3.html#3.2.1">http://nsis.sourceforge.net/Docs/Chapter3.html#3.2.1</a>                                                                                                                                                             | en         | Page « Command Line Usage / Installer Usage / Common Options » de NSIS                                                                                                            |
|             | <a href="http://unattended.sourceforge.net/installers.php">http://unattended.sourceforge.net/installers.php</a>                                                                                                                                                                     | en         | « Unattended, A Windows deployment system »: une base de connaissances sur les installations non-assistées.                                                                       |
|             | <a href="http://wud.jcarle.com/UpdateLists.aspx">http://wud.jcarle.com/UpdateLists.aspx</a>                                                                                                                                                                                         | en         | « Program Files Update Lists »: liste des Hotfixes Microsoft, ainsi que leurs URL <sup>1</sup> de téléchargement.                                                                 |

<sup>1</sup> De nombreux hotfixes ne sont installables qu'avec un accès Internet. Ce site permet de les télécharger une fois depuis le site MS, et de les installer à la demande.

| <i>Réf.</i> | <i>Site</i>                                                                                                                                                                                                                     | <i>Lng</i> | <i>Description</i>                                                                                                                                                                          |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | <a href="https://kb.altiris.com/display/1n/kb/article.asp?aid=3043">https://kb.altiris.com/display/1n/kb/article.asp?aid=3043</a>                                                                                               | en         | Article « What are the command line parameters available for WiseScript? »: liste des options standard d'un WiseScript.                                                                     |
|             | <a href="http://www.heise-online.co.uk/security/features/80682/0">http://www.heise-online.co.uk/security/features/80682/0</a>                                                                                                   | en         | Utilitaire « c't Offline-Update », de constitution d'un CD de mise à jour des produits MS, avec les dernières corrections.                                                                  |
| [MS262841]  | <a href="http://support.microsoft.com/kb/262841">http://support.microsoft.com/kb/262841</a>                                                                                                                                     | fr         | Article « Commutateurs de ligne de commande pour les packages de mise à jour logicielle Windows ».                                                                                          |
| [MS296861]  | <a href="http://support.microsoft.com/?kbid=296861">http://support.microsoft.com/?kbid=296861</a>                                                                                                                               | fr         | Article « Comment faire pour installer plusieurs mises à jour ou correctifs Microsoft Windows en un seul redémarrage » expliquant les risques à installer plusieurs hotfixes simultanément. |
| [XPHOT]     | <a href="http://www.microsoft.com/windowsxp/downloads/updates/sp1/hfdeploy.msp">http://www.microsoft.com/windowsxp/downloads/updates/sp1/hfdeploy.msp</a>                                                                       | en         | Article « Microsoft Windows XP Hotfix Installation and Deployment Guide ».                                                                                                                  |
| [WSH]       | <a href="http://msdn2.microsoft.com/en-us/library/9bbdkx3k(VS.85).aspx">http://msdn2.microsoft.com/en-us/library/9bbdkx3k(VS.85).aspx</a>                                                                                       |            | « Windows Script Host ».                                                                                                                                                                    |
|             | <a href="http://sourceforge.net/tracker/index.php?func=detail&amp;aid=1877926&amp;group_id=206943&amp;atid=999622">http://sourceforge.net/tracker/index.php?func=detail&amp;aid=1877926&amp;group_id=206943&amp;atid=999622</a> | en         | ISTool adds semicolons where they shouldnt be                                                                                                                                               |
|             | <a href="http://sourceforge.net/tracker/index.php?func=detail&amp;aid=1892810&amp;group_id=206943&amp;atid=999623">http://sourceforge.net/tracker/index.php?func=detail&amp;aid=1892810&amp;group_id=206943&amp;atid=999623</a> | en         | multiple same sections                                                                                                                                                                      |
| [REDIS]     | <a href="http://www.jrsoftware.org/iskb.php?redistribute">http://www.jrsoftware.org/iskb.php?redistribute</a>                                                                                                                   | en         | What files do I need to deploy?                                                                                                                                                             |

## 13.8 Glossaire

| <i>Terme</i> | <i>Définition</i>                                                                                                           | <i>Références</i>                                                                                                                                                                   |
|--------------|-----------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DOS          | Disk Operating System.<br>Nom générique utilisé pour désigner MS-DOS.                                                       |                                                                                                                                                                                     |
| IDE          | <b>I</b> ntegrated <b>D</b> evelopment <b>E</b> nvironment.<br>Environnement de développement intégré.                      | <a href="http://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9">http://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9</a> |
| IHM          | <b>I</b> nterface <b>H</b> omme- <b>M</b> achine                                                                            |                                                                                                                                                                                     |
| KiB          | <b>K</b> ibiByte.<br>1024 octets                                                                                            | <a href="http://en.wikipedia.org/wiki/Binary_prefix#IEC_standard_prefixes">http://en.wikipedia.org/wiki/Binary_prefix#IEC_standard_prefixes</a>                                     |
| N/A          | <b>N</b> on <b>A</b> pplicable                                                                                              |                                                                                                                                                                                     |
| OoO          | <b>O</b> pen <b>O</b> ffice. <b>o</b> rg                                                                                    |                                                                                                                                                                                     |
| Registre     |                                                                                                                             |                                                                                                                                                                                     |
| Routine      | Terme générique désignant une procédure ou une fonction Pascal.                                                             |                                                                                                                                                                                     |
| RTF          | <b>R</b> ich <b>T</b> ext <b>F</b> ormat.<br>Format de fichier texte largement répandu permettant une présentation soignée. |                                                                                                                                                                                     |
| UUID         |                                                                                                                             |                                                                                                                                                                                     |