

Déploiement d'une application

Le mécanisme de chargement automatique

Depuis AutoCAD 2013, le déploiement d'applications personnalisées est grandement facilité par l'utilisation du mécanisme de chargement automatique à l'aide d'un simple dossier. Ce dossier porte un nom avec l'extension `.bundle`. Il contient les composants de l'application ainsi qu'un fichier XML dans lequel sont définis les paramètres de l'application.

Installation / désinstallation

Pour que l'application soit automatiquement chargée il faut que le dossier soit placé dans un des répertoires `ApplicationPlugins` d'un disque local.

Dossier d'installation général

`%PROGRAMFILES%\Autodesk\ApplicationPlugins`

Chemin approuvé par défaut, nécessite des droits administrateurs.

Dossier pour tous les utilisateurs

`%PROGRAMDATA%\Autodesk\ApplicationPlugins`

Dossier pour l'utilisateur courant

`%APPDATA%\Autodesk\ApplicationPlugins`

L'installation de l'application consistera simplement à coller le dossier dans un de ces répertoires. Pour la désinstallation, il suffira de supprimer le dossier.

La copie du dossier dans le répertoire choisi peut aussi se faire via un installeur (voir plus bas).

Pour plus de détails, voir [la documentation](#).

Le dossier `.bundle`

Ce dossier contient tous les composants nécessaires à l'application éventuellement répartis dans différents sous-dossiers. Les composants de l'application sont généralement des fichiers de programme (`.lsp`, `.fas`, `.vlx`, `.arx`, `.dbx`, `.xaml`, `.js`, `.dll`, `.dvb`), des fichiers de personnalisation partielle (`.cuix`), des palettes d'outils (`.atc`) et des fichiers de ressources nécessaires à l'application (fichiers DCL, images, icônes, dessins, fichiers d'aide, etc.)

Il doit aussi contenir un fichier `PackageContents.xml` dans lequel sont définis les paramètres de l'application.

Le fichier de métadonnées `PackageContents.xml`

Ce fichier contient les informations sur le chargement des différents composants de l'application. Il peut aussi contenir des informations concernant l'auteur de l'application.

Les éléments et attributs obligatoires dans le fichier XML sont plus nombreux si l'application doit être publiée sur la boutique Autodesk (Autodesk App Store) que si elle est destinée à être directement distribuée.

Le fichier est organisé à partir d'un élément de base (`ApplicationPackage`) dont certains attributs sont obligatoires dans tous les cas. Cet élément contient optionnellement un élément pour les informations sur l'auteur (`CompanyDetails`) et un élément pour les composants (`Components`). Ce dernier élément contient, entre autres, des éléments : `RuntimeRequirements` pour les conditions de chargement ; `ComponentEntry` pour chaque fichier de programme, de personnalisation, de palette

d'outils ; SystemVariable pour créer ou définir les valeurs de variables système ; EnvironmentVariable pour créer ou définir les valeurs de variables d'environnement.

Pour plus de détails, voir [la documentation](#).

Exemple

Structure d'un dossier .bundle de base pour les applications Edit_Bloc et TotalArea (téléchargeable sur [cette page](#)).

Le dossier **EditBloc TotalArea.bundle** contient le fichier de métadonnées PackageContents.xml et un dossier **Contents**.

Le dossier **Contents** contient les fichiers LISP TotalArea.lsp, Edit_bloc_3.5.lsp et un dossier **Resources**.

Le dossier **Resources** contient le fichier DCL Edit_Bloc.dcl, le fichier bloc TotalArea.dwg et le fichier d'aide TotalArea.chm.

EditBloc TotalArea.bundle

- | - PackageContents.xml
- | - **Contents**
 - | - Edit_Bloc_3.5.lsp
 - | - TotalArea.lsp
- | - **Resources**
 - | - Edit_Bloc.dcl
 - | - TotalArea.dwg
 - | - TotalArea.chm

Le fichier PackageContents.xml contient les éléments nécessaires pour une distribution directe de l'application.

Dans l'élément ApplicationPackage, l'attribut ProductCode doit être un identifiant unique (GUID). Il existe de nombreux générateurs de GUID sur le web.

Dans l'élément RuntimeRequirements, l'attribut SupportPath définit un chemin de support vers le dossier Resources et l'attribut OS définit le système d'exploitation (Windows ici), pour plus de détails voir [la documentation](#).

Dans les éléments ComponentEntry, l'attribut ModuleName définit le chemin du fichier LISP et l'attribut PerDocument doit être égal à "True" pour que les fichiers LISP soient automatiquement chargés dans tous les documents. Les chemins spécifiés sont relatifs au dossier .bundle.

L'utilisation d'un éditeur de code (Notepad++, VS Code) facilitera l'écriture du fichier XML.

```
<?xml version="1.0" encoding="utf-8" ?>
<ApplicationPackage
  SchemaVersion="1.0"
  AppVersion="1.0"
  ProductCode="{092FF656-2D2A-4AC4-A6AA-6ECDCCA1AA77}"
  Name="Total Area" >
  <Components>
    <RuntimeRequirements SupportPath="./Contents/Resources" OS="Win32|Win64 />
    <ComponentEntry ModuleName="./Contents/TotalArea.lsp" PerDocument="True" />
    <ComponentEntry ModuleName="./Contents/Edit_Bloc_3.5.lsp" PerDocument="True" />
  </Components>
</ApplicationPackage>
```

Cet exemple est minimaliste et ne renseigne que les éléments et attributs nécessaires et/ou obligatoires pour cette application LISP. On pourra le complexifier en ajoutant d'autres éléments et attributs au fichier PackageContents.xml.

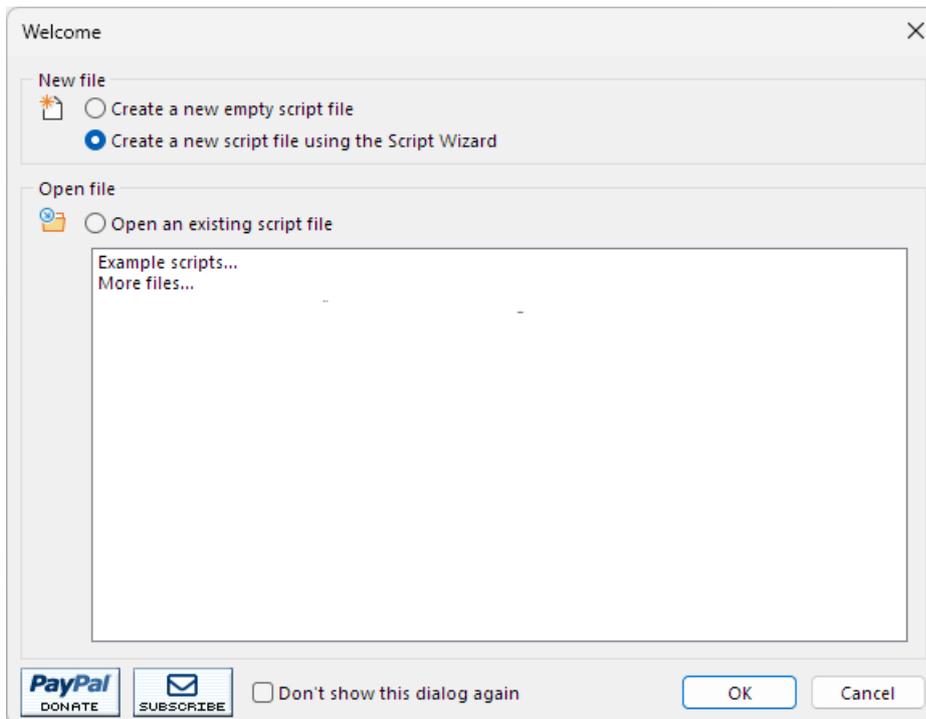
Création d'un installeur

Une fois satisfait du fonctionnement du dossier .bundle, pour le distribuer en évitant à l'utilisateur d'avoir à le coller dans un répertoire, il est possible de créer un installeur qui s'en chargera.

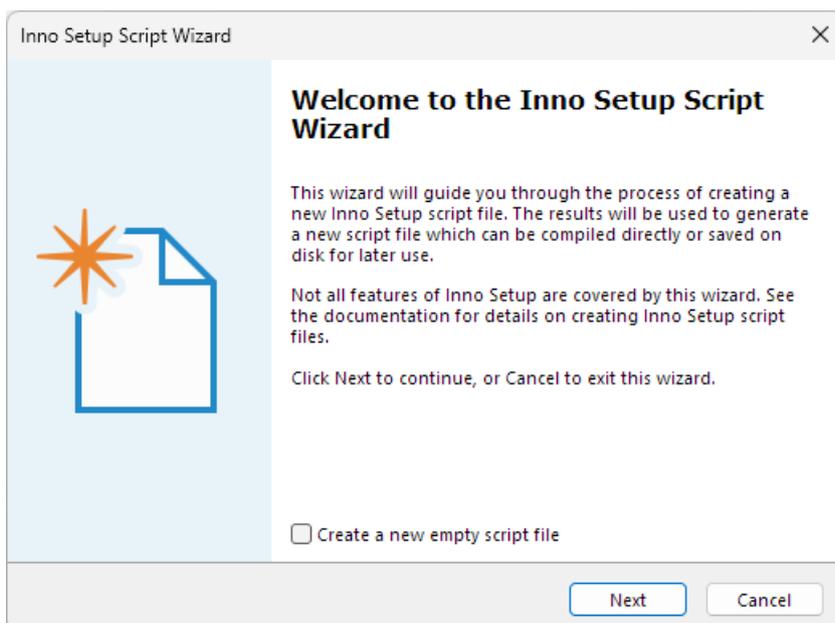
Exemple avec Inno Setup

Inno Setup est un logiciel gratuit et Open Source qui permet de créer facilement un installeur.

Après avoir [téléchargé](#) et installé Inno Setup, lancer le programme.



Cocher la case "Create a new script file using the Script Wizard" et cliquer sur "OK".



Cliquer sur "Next".

Entrer le nom de l'application ("Application name") et la version ("Application version") et, optionnellement les champs "Application publisher" et "Application website".
Cliquez sur "Next".

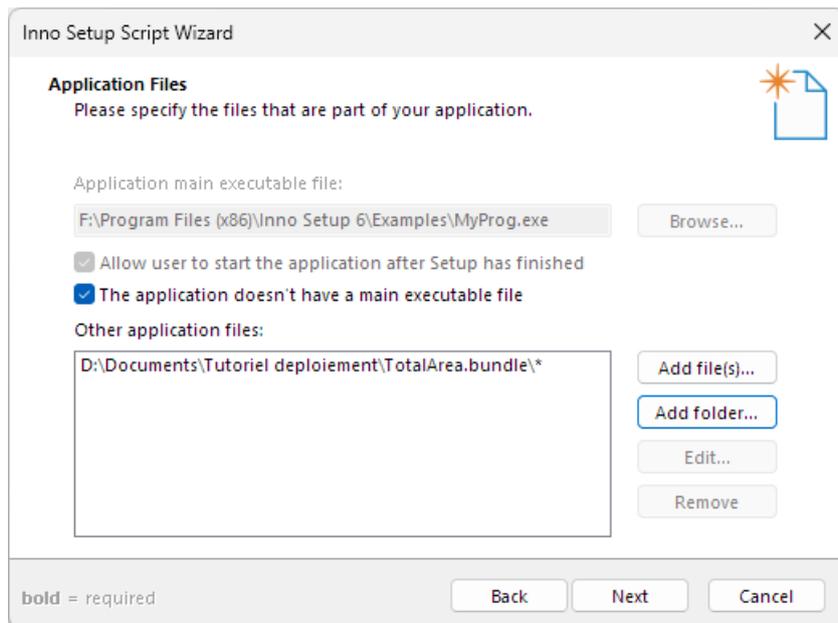
Dans la liste déroulante "Application destination base folder" choisir "(Custom)" et entrer le chemin du dossier ApplicationPlugins dans lequel installer l'application :

- "{userappdata}\Autodesk\ApplicationPlugins" (%appdata%)
- "{commonappdata}\Autodesk\ApplicationPlugins" (%programdata%)
- "{commonpf}\Autodesk\ApplicationPlugins" (%programfiles%)

Ajouter l'extension .bundle à "Application folder name".

Décochez "Allow user to change the application folder".

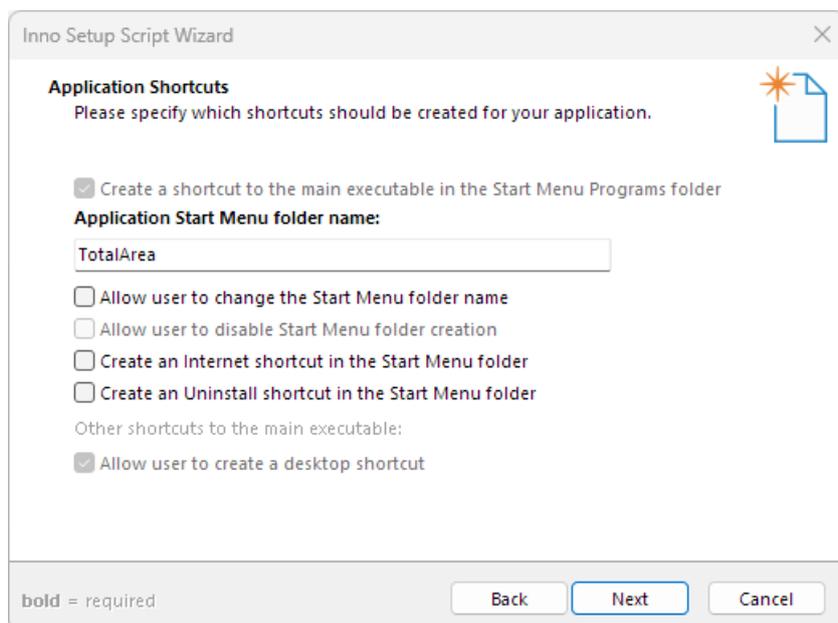
Cliquez sur "Next".



Cocher "The application doesn't have a main executable file". Cliquer sur "Add folder" et ajouter le dossier .bundle.

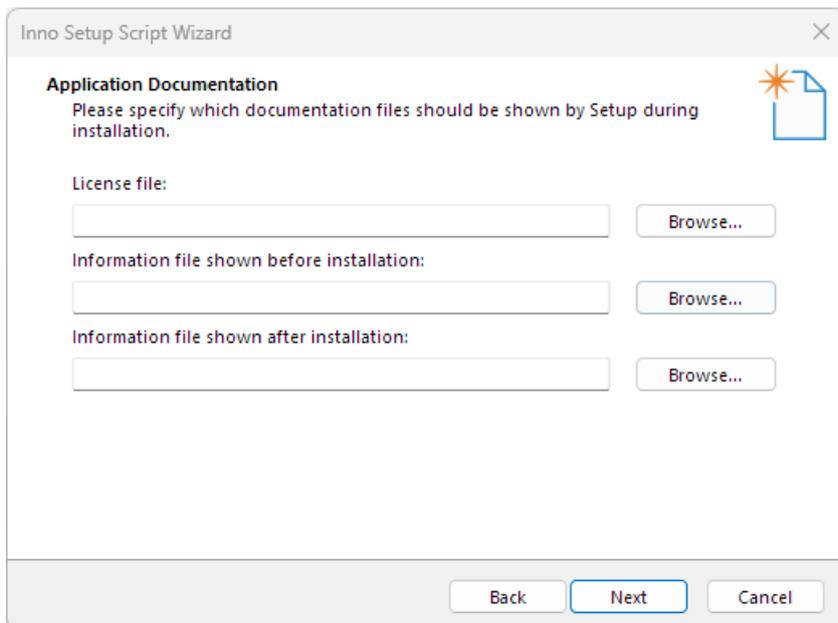
Répondre "Oui" à la boîte de dialogue "Should files in subfolders of [...] also be included?".

Cliquer sur "Next".

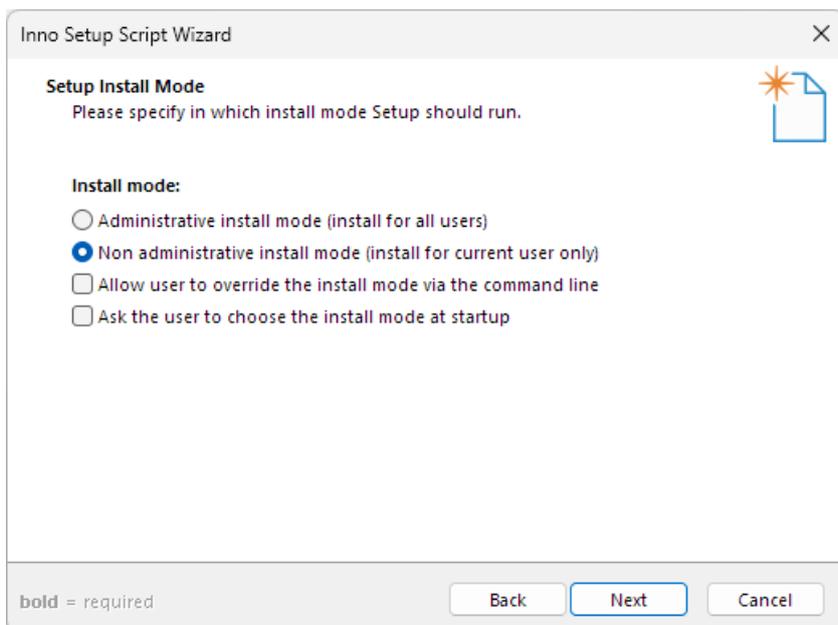


Décocher "Allow user to change the StartMenu folder name".

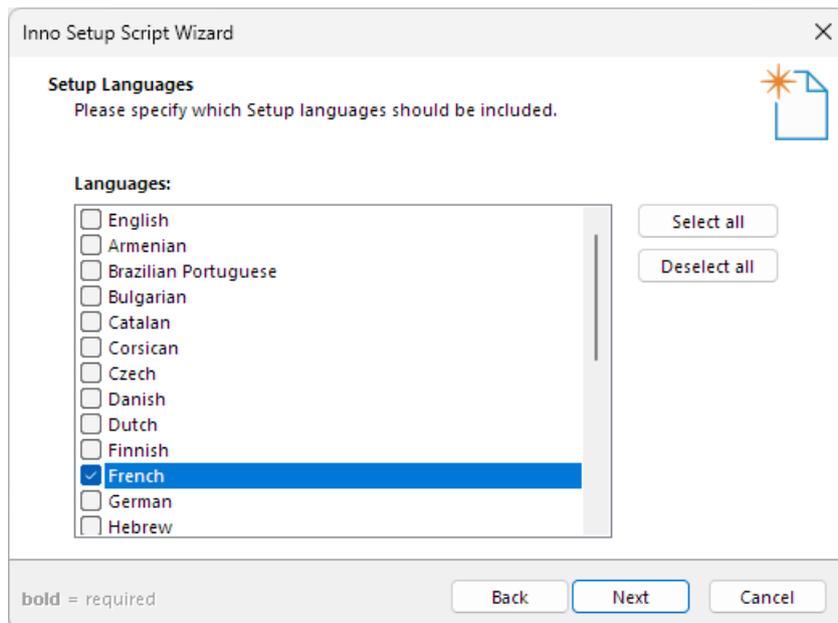
Cliquer sur "Next".



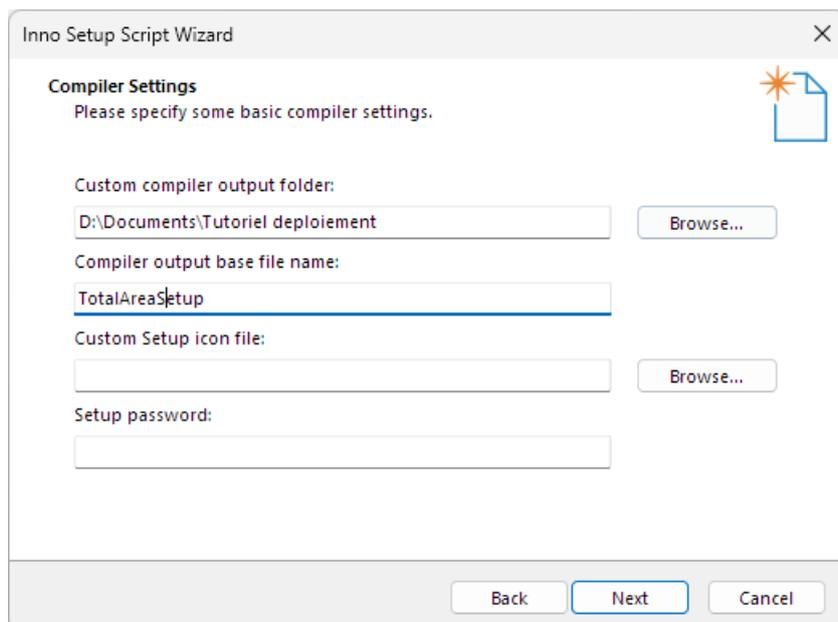
Ajouter optionnellement des fichiers texte pour la licence, les informations à afficher avant et/ou après l'installation.
Cliquer sur "Next".



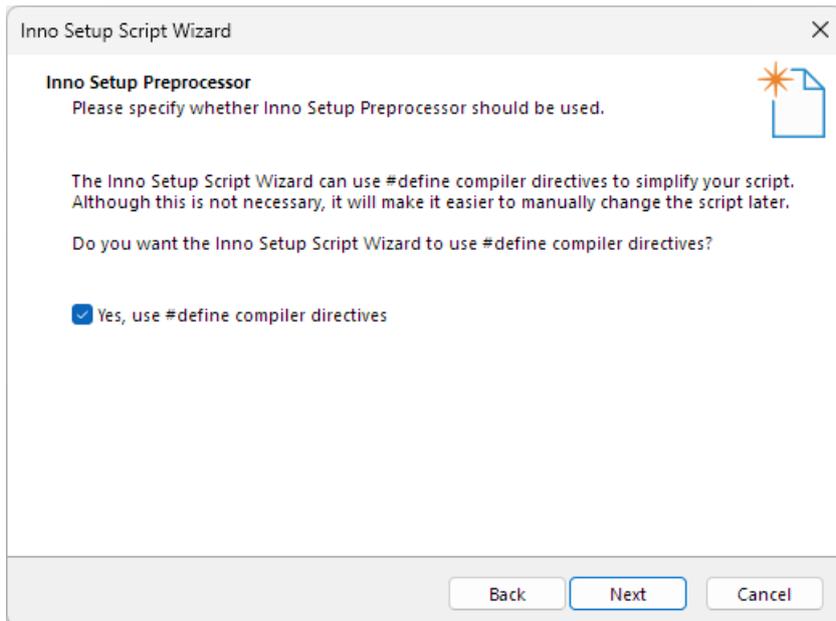
Cocher "Non administrative install mode" si le dossier cible n'est pas dans %programfiles%.
Cliquer sur Next.



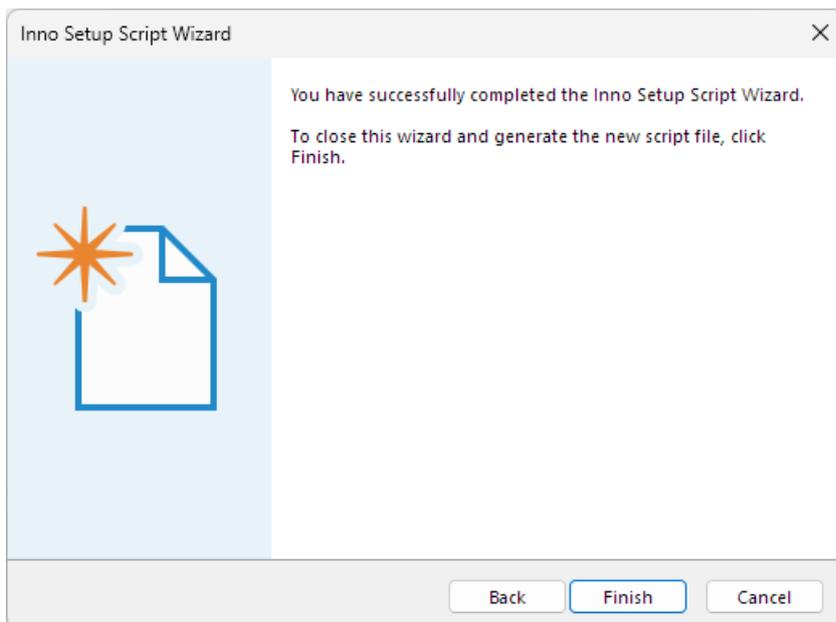
Cocher "French" (et optionnellement d'autres langues).
Cliquer sur "Next".



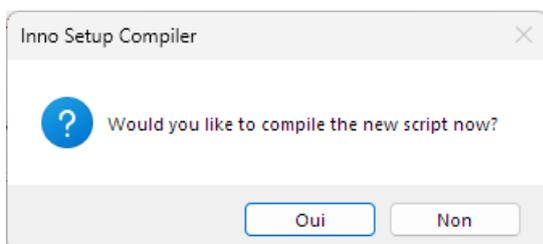
Spécifier le dossier de sortie ("Custom compiler output folder") et le nom ("Compiler output base file name") de l'installeur.
Cliquer sur "Next".



Cliquer sur "Next".



Cliquer sur "Finish".



Cliquer sur "Non".

```
Untitled - Inno Setup Compiler 6.2.2
File Edit View Build Run Tools Help
; Script generated by the Inno Setup Script Wizard.
; SEE THE DOCUMENTATION FOR DETAILS ON CREATING INNO SETUP SCRIPT FILES!

#define MyAppName "TotalArea"
#define MyAppVersion "1.0"
#define MyAppPublisher "gileCAD"
#define MyAppURL "https://gilecad.azurewebsites.net"

[Setup]
; NOTE: The value of AppId uniquely identifies this application. Do not use the same AppId value in installers for
; (To generate a new GUID, click Tools | Generate GUID inside the IDE.)
AppId={092FF6S6-2D2A-4AC4-A6AA-6ECDCC1AA77}
AppName={#MyAppName}
AppVersion={#MyAppVersion}
;AppVerName={#MyAppName} {#MyAppVersion}
AppPublisher={#MyAppPublisher}
AppPublisherURL={#MyAppURL}
AppSupportURL={#MyAppURL}
AppUpdatesURL={#MyAppURL}
DefaultDirName={userappdata}\Autodesk\ApplicationPlugins\TotalArea.bundle
DisableDirPage=yes
DefaultGroupName={#MyAppName}
DisableProgramGroupPage=yes
; Remove the following line to run in administrative install mode (install for all users.)
PrivilegesRequired=lowest
OutputDir=D:\Documents\Tutoriel deployment
OutputBaseFilename=TotalAreaSetup
Compression=lzma
SolidCompression=yes
WizardStyle=modern

[Languages]
Name: "french"; MessagesFile: "compiler:Languages\French.isl"

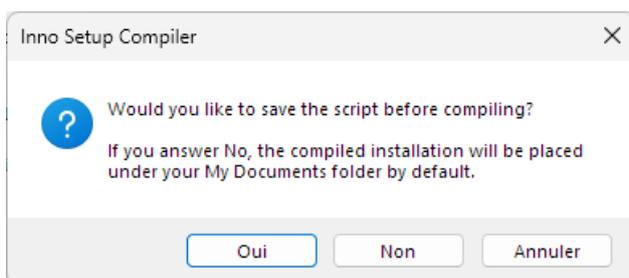
[Files]
Source: "D:\Documents\Tutoriel deployment\TotalArea.bundle\*"; DestDir: "{app}"; Flags: ignoreversion recursesub
; NOTE: Don't use "Flags: ignoreversion" on any shared system files

12: 46 Modified Insert
```

Le script de l'installateur a été généré (langage Pascal).

Dans l'éditeur de script, copier le GUID de AppId pour le coller dans l'attribut ProductCode du fichier PackageContents (ou l'inverse).

Cliquer sur l'icône "Compile".



Cliquer sur « Oui » pour enregistrer le script à un autre emplacement que Documents.

Attendre la fin de la compilation. L'installateur a été créé dans le dossier spécifié.