

Ein Inno Setup Installationsprogramm für eine OpenDCL Anwendung

Dieses Tutorium zeigt Ihnen, wie Sie ein relativ einfaches Installationsprogramm für Ihre OpenDCL Anwendung herstellen können. Das Installationsprogramm ist eine EXE-Datei, die sich die Anwender entweder von einer Webseite herunterladen können oder Sie Ihnen in einer E-Mail oder auf einem Datenträger zusenden können. Das Installationsprogramm beinhaltet alle Ihre .ODCL- und .LSP-Dateien sowie die OpenDCL Laufzeitumgebung, die auf den Arbeitsplätzen der Anwender installiert werden sollen.

Dieses Tutorium verwendet Inno Setup 5, einem *freien* Werkzeug für die Distribution von Windows Programmen. Es wurde von Jordan Russel geschrieben. Sie können es auf der Webseite <http://www.jrsoftware.org/isdl.php>¹ herunterladen. Das für Sie einfachste Paket ist die `ispack-5.4.2.exe`², die sowohl das Programm Inno Setup aber auch einige Anwendungen von Drittanbietern enthält, die die Arbeit mit Inno Setup noch einfacher gestalten. Das für Sie wichtigste Programm aus dem Paket ist der Inno Setup Compiler, einem Skript-Editor, das Erstellen von Installationsanweisungen einfach gestaltet.

Der Inno Setup Compiler beinhaltet einen Assistenten, der Ihnen hilft, die wichtigsten Teile des Skripts vorab zusammenzustellen.

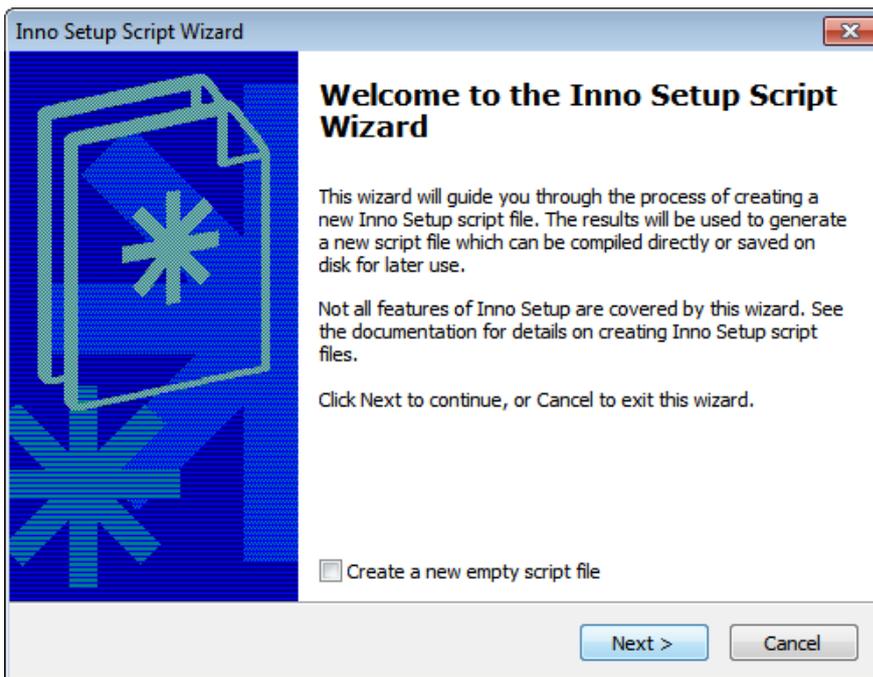


Abbildung 1. Der Inno Setup Script Assistent.

¹ Die Autoren dieses Tutorium übernehmen keine Gewährleistung und Haftung für den Inhalt der verlinkten Webseite. Für die Inhalte und Richtigkeit der Informationen ist stets der jeweilige Informationsanbieter der verlinkten Webseite verantwortlich.

² Der Dateiname kann in Abhängigkeit von der Produktversion variieren.

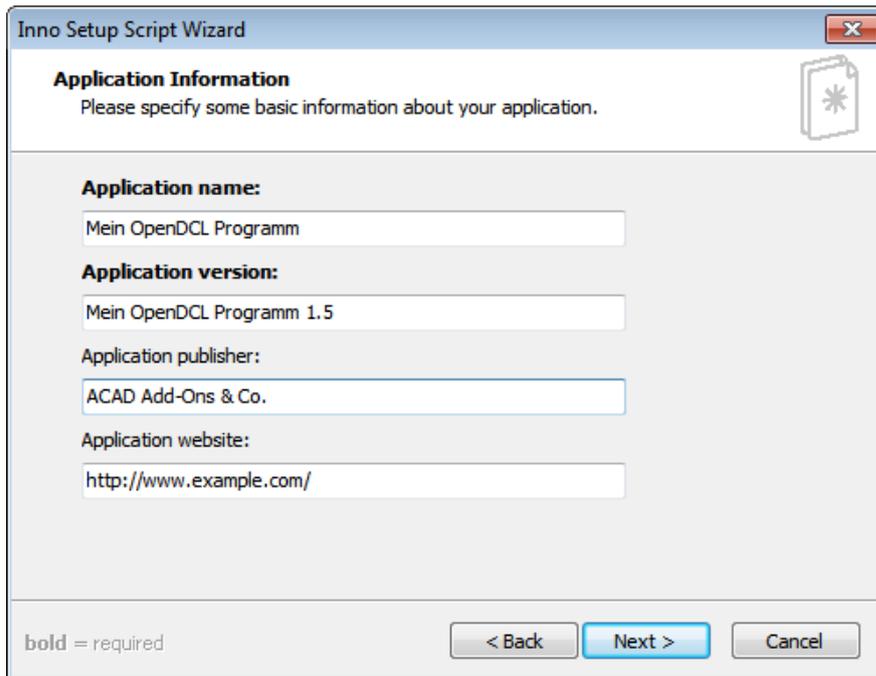


Abbildung 2. Diese Informationen werden bei der Ausführung des Installationsprogramms angezeigt.

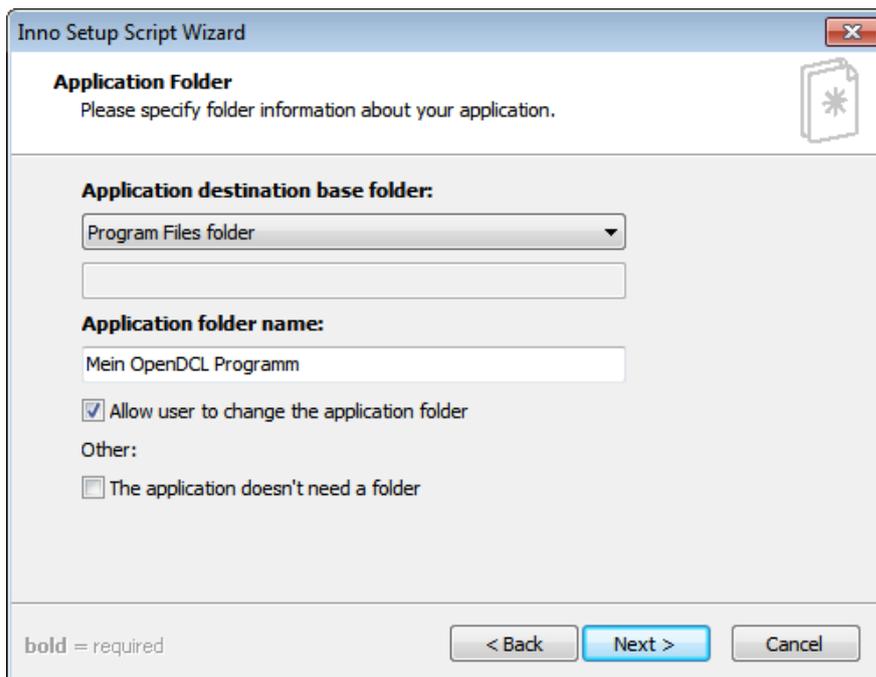


Abbildung 3. Hier entscheiden Sie, wo das Produkt auf dem Arbeitsplatz des Anwenders installiert werden soll.

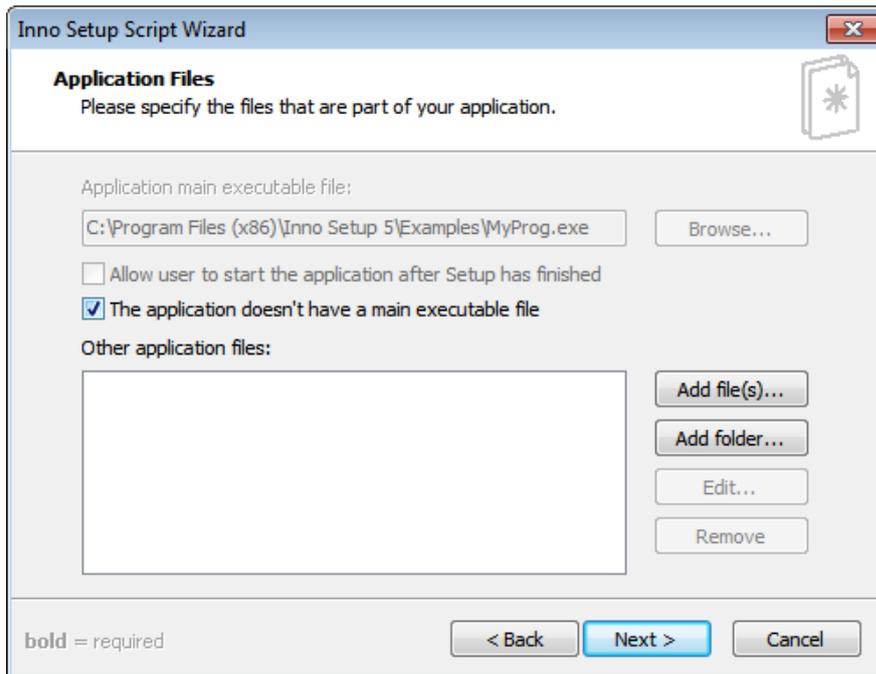


Abbildung 4. Hier können Sie Dateien und Ordner auswählen, die das Installationsprogramm auf den Arbeitsplatz kopieren soll. Sie können dies aber auch später im Skript ergänzen. Verwendet Ihre OpenDCL-Anwendung keine ausführbare Datei, aktivieren Sie den Zustandsschalter "The application doesn't have a main executable file."

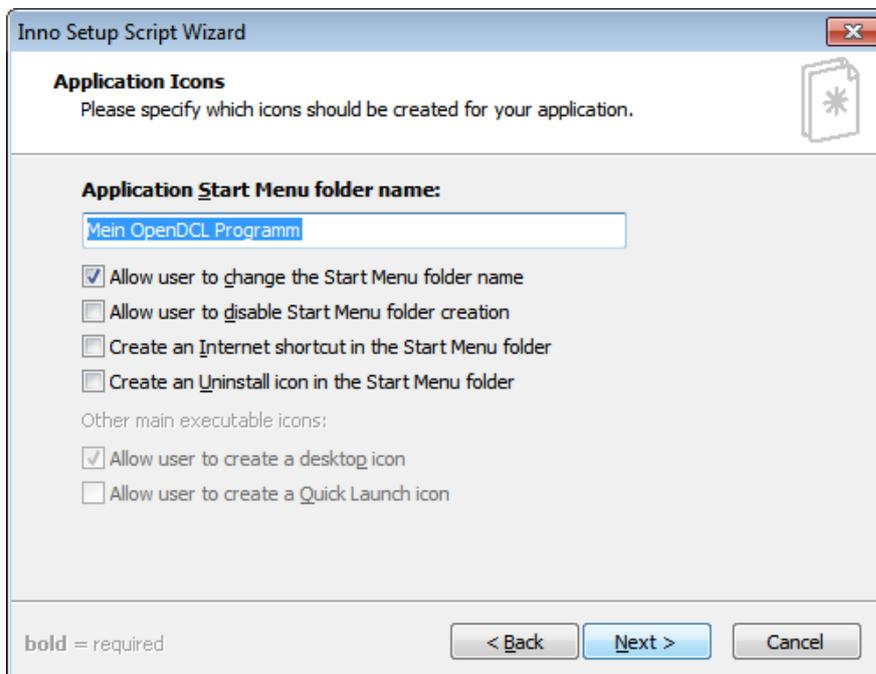


Abbildung 5. Hier können Sie Startmenü-Optionen auswählen.

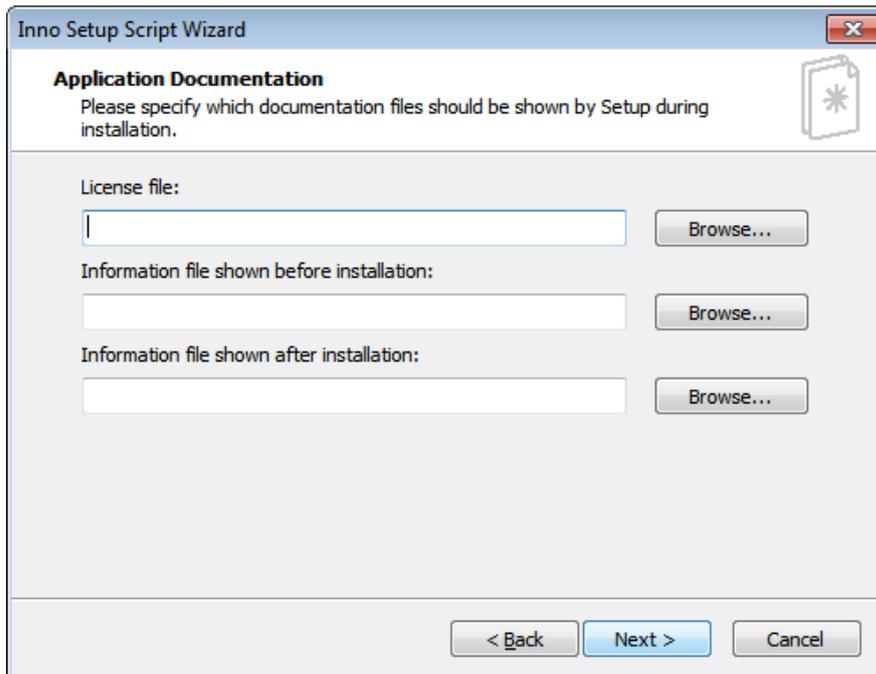


Abbildung 6. Hier können Sie einige Informationen auswählen, die dem Anwender bei der Installation angezeigt werden und ihn bei den Einstellungen unterstützen sollen. Die Textdatei “before installation” sollte Informationen über das Hinzufügen von Verzeichnissen zu den AutoCAD-Suchpfaden beinhalten.

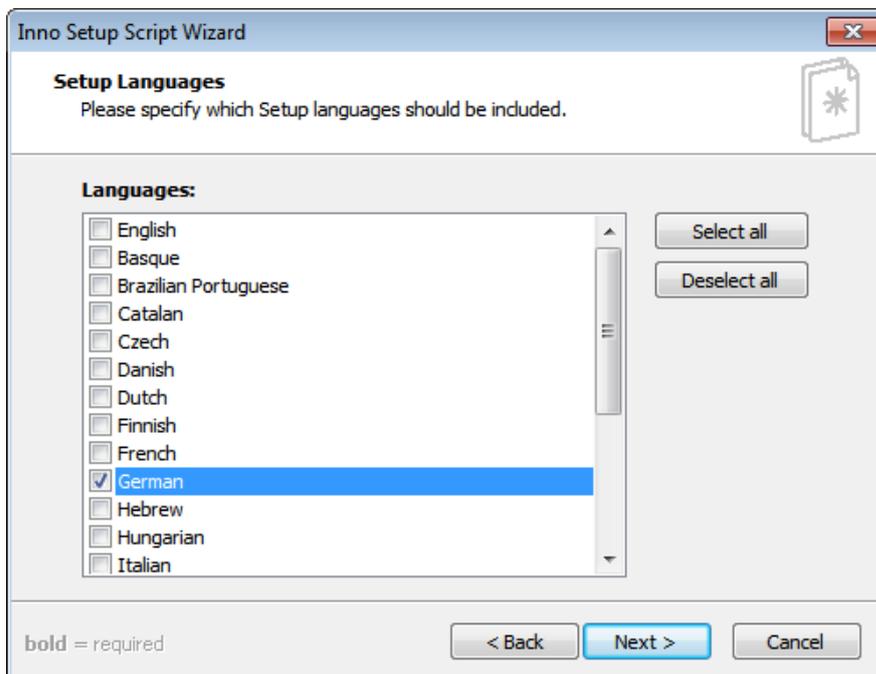


Abbildung 7. Hier bestimmen Sie, welche Sprachen das Installationsprogramm unterstützen soll. Damit ist jedoch nicht Ihre Anwendung gemeint.

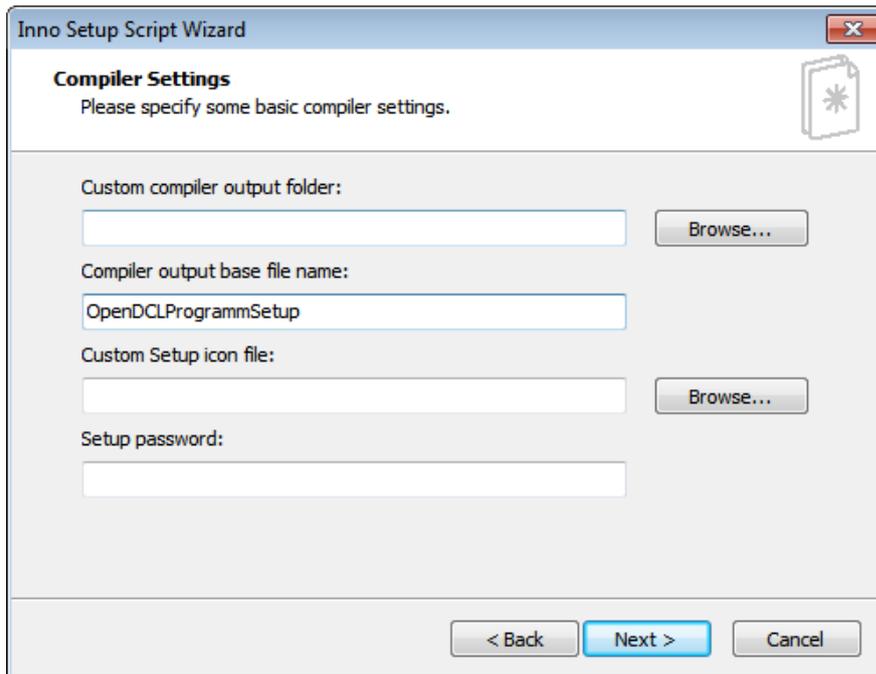


Abbildung 8. Hier können Sie bestimmen, wo die ausführbare Datei des Installationsprogramms auf dem Arbeitsplatz des Entwicklers abgelegt wird und welchen Namen die EXE-Datei haben soll. Sie sollten einen geeigneten Namen eingeben, um später diese Datei von denen anderer Projekte unterscheiden zu können. Diese Einstellungen haben keine Auswirkungen auf die installierten Dateien oder die Anwenderarbeitsplatz.

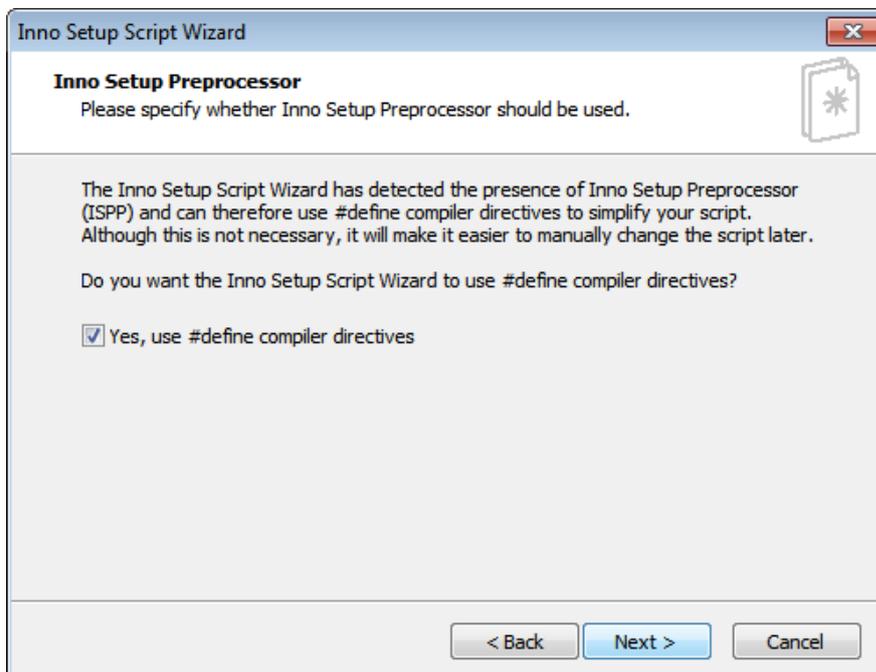


Abbildung 9. Wurde der Inno Setup Preprocessor gefunden, können Compiler-Anweisungen gesetzt werden.

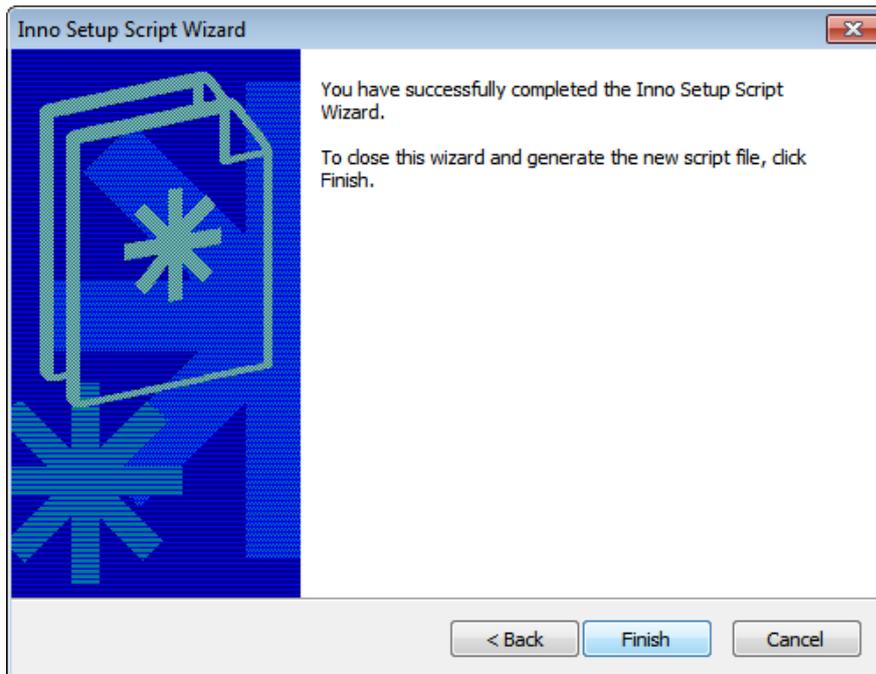


Abbildung 10. Fertig!

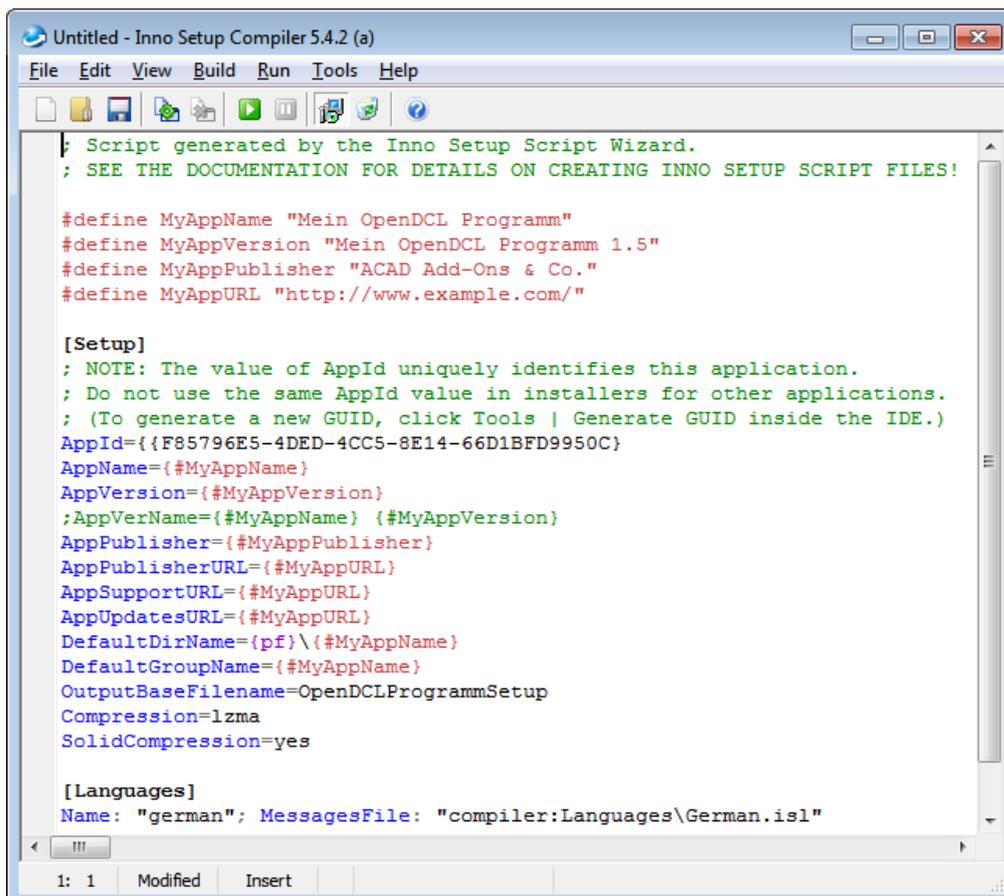


Abbildung 11. Das fertige Skript des Anwendungs-Erstellungs-Assistenten.

Abbildung 11 zeigt genau das Skript, das der Assistent aus Ihren Eingaben generiert hat. Sie sind bereits fast fertig. Einzig die Installation der OpenDCL-Laufzeitumgebung fehlt noch.

Zu diesem Zweck sind zwischen den Abschnitten [**Languages**] und [**Icons**] noch ein paar weitere Zeilen erforderlich. Hier dazu ein paar Anmerkungen über die Arbeitsweise von Inno Setup:

- Wenn Sie Ihre Dateien unterhalb des Installationsverzeichnisses des Inno Setup Compilers ablegen, müssen Sie nicht den vollen Pfad angeben, sondern können mit *relativen Pfadangaben* auskommen. Erstellen Sie einfach einen Unterordner **lspfiles**, auf den Sie sich später im Skript beziehen können.
- Die Abschnitte tragen *spezifische Bezeichnungen*, die nicht verändert werden dürfen. Sie erkennen dies daran, dass sie nach korrekter Schreibweise automatisch **fett** dargestellt werden.
- Der Abschnitt [**Files**] verwendet die spezifischen Ausdrücke **Source:** , **DestDir:** und **Flags:** . Auch diese müssen in dieser Schreibweise verwendet werden, damit sie der Compiler erkennt. Es gibt weitere Ausdrücke, die in diesem Tutorium nicht näher betrachtet werden.
- In den Abschnitten [**Run**] und [**UninstallRun**] verwenden Sie Anführungszeichen, um die Parameter der MSIEXEC.EXE zu übergeben. Der gesamte Ausdruck mit den Parametern ist somit mit doppelten Anführungszeichen zu umschließen. Da Pfadangaben – auch die der Variable {app} – Leerzeichen enthalten können, müssen diese dringend *doppelt mit doppelten Anführungszeichen* umschlossen werden (""), da andernfalls der Compiler nach dem ersten doppelten Anführungszeichen innerhalb der Parameterkette das Parsen der Parameter beendet. Für die Angabe der Parameter mit der Pfadangabe sind demnach *3 Paare doppelter Anführungszeichen* notwendig, andernfalls wird MSIEXEC später nicht korrekt ausgeführt.

Vervollständigen Sie den Code unter Berücksichtigung der o.g. Hinweise.

Nach der letzten Zeile des Abschnitt Language fügen Sie eine Leerzeile ein und in einer weiteren beginnen Sie mit dem Abschnitt [**Files**]. Der Name sollte sofort fett dargestellt werden, wenn Sie es richtig geschrieben haben. Starten Sie in der nächsten Zeile mit dem Ausdruck **Source:** . Ergänzen Sie ein Leerzeichen und schreiben Sie dahinter den relativen Pfad bezogen auf die Skriptdatei, die Sie editieren, und natürlich den Namen der ersten Datei. In diesem Tutorium werden Sie eine .LSP- und eine .ODCL-Datei hinzufügen, auch wenn Sie später eine .VLX-Anwendung weitergeben, die beides beinhaltet. Letztlich ist das Vorgehen identisch, und Sie erhalten in beiden Fällen dieses Ergebnis:

```
Source: lspfiles\Mein OpenDCL Program.lsp; DestDir: {app}; Flags: ignoreversion
```

Die Definition sollte auf einer Zeile stehen. Mit **DestDir:** wird das Zielverzeichnis bestimmt. Dabei kommt die Variable {app} zur Anwendung, die im Abschnitt [**Setup**] mit **DefaultDirName** belegt ist – in diesem Falle Programme\Mein OpenDCL Programm. Mit

ignoreversion bestimmen Sie, dass existierende Dateien unabhängig von der Version überschrieben werden.

Fügen Sie anschließend die .ODCL-Datei hinzu. Kopieren Sie dazu einfach die letzte Zeile und tauschen Sie die Dateierweiterungen aus.

```
Source: lspfiles\Mein OpenDCL Program.odcl; DestDir: {app}; Flags: ignoreversion
```

Abschließend soll noch die OpenDCL-Laufzeitumgebung auf dem Arbeitsplatz abgelegt werden, um alle Dateien im Installationsverzeichnis der Applikation bereitzustellen. Verwenden Sie dieselbe Syntax. Damit wird sichergestellt, dass die .MSI auch beim Zeitpunkt der Deinstallation verfügbar ist. Inno Setup installiert die .MSI von einem temporären Ordner aus, der nach der Installation automatisch gelöscht wird und zum Zeitpunkt der Deinstallation somit nicht mehr vorhanden ist. Damit würde die Deinstallation fehlschlagen, weil die .MSI nicht mehr existiert.

Die Zeile lautet somit

```
Source: lspfiles\OpenDCL.Runtime.6.0.2.3.msi; DestDir: {app}; Flags: ignoreversion
```

Bitte beachten Sie, dass der Dateiname der OpenDCL-Laufzeitumgebung von der Version abhängt, die Sie ausliefern wollen.

Damit haben Sie die Änderungen im Abschnitt **[Files]** abgeschlossen. Lassen Sie eine Zeile frei. Die Leerzeile ist nicht zwingend erforderlich, erleichtert jedoch das Lesen des Quellcodes. Fügen Sie in die nächste Zeile den Abschnitt **[Run]** ein.

Der Inhalt des Abschnitts **[Run]** ist sehr wichtig. Wird der Ausdruck des Aufrufs der .MSI nicht korrekt eingegeben, wird MSIEXEC beim Ausführen der .MSI fehlschlagen. Die benötigte Syntax lautet:

```
Filename: msiexec.exe; Parameters: "/i ""{app}\OpenDCL.Runtime.6.0.2.3.msi"" /qn"
```

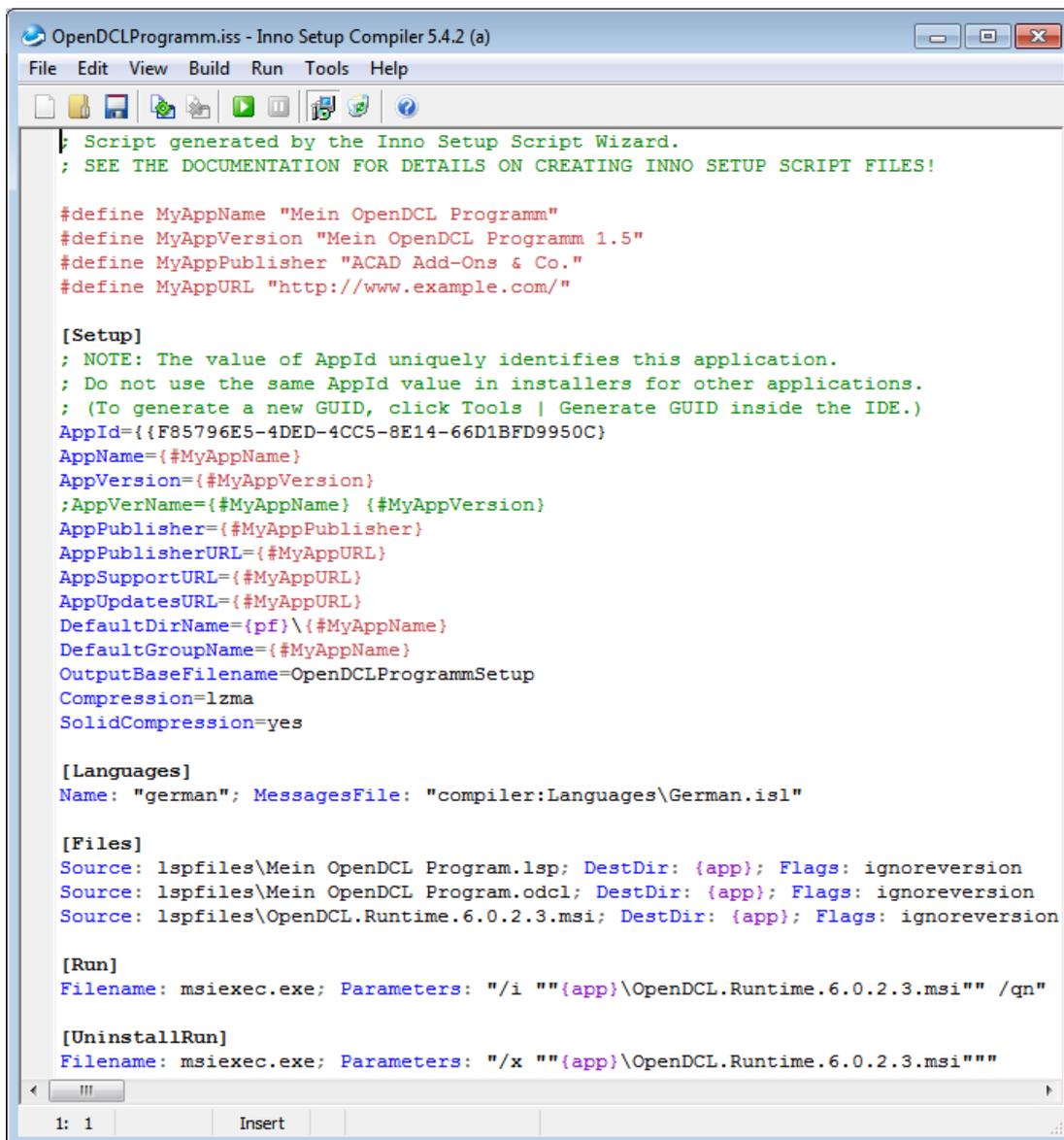
Der gesamte Ausdruck muss auf einer Zeile stehen. Bitte achten Sie auf die Leerzeichen und die doppelten Anführungszeichen. Der Parameter **/i** veranlasst die Installation der .MSI-Datei, **/qn** unterdrückt die Ausgabe von Dialogen. Weiterführende Angaben zu den Schaltern und Parametern der MSI-Datei erhalten Sie, wenn Sie die Eingabeaufforderung starten und den Befehl MSIEXEC aufrufen. Windows zeigt Ihnen daraufhin einen Dialog mit allen Schaltern und Parametern. Gleiches gilt im Übrigen, wenn Sie den o.g. Ausdruck nicht richtig schreiben: auch dann wird anstelle der Installation der .MSI dieser Dialog angezeigt.

Lassen Sie erneut eine Zeile frei und beginnen Sie den Abschnitt **[UninstallRun]**. Der benötigte Ausdruck ist ähnlich dem vorherigen:

```
Filename: msiexec.exe; Parameters: "/x ""{app}\OpenDCL.Runtime.6.0.2.3.msi"""
```

Mit dem Schalter **/x** wird bestimmt, dass während der Deinstallation MSIEXEC die .MSI-Datei deinstallieren soll. Der Parameter **/qn** wird bei der Deinstallation nicht berücksichtigt, weswegen Sie ihn getrost weglassen können.

Das Installationsskript sieht nun so aus:



```
OpenDCLProgramm.iss - Inno Setup Compiler 5.4.2 (a)
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 "Mein OpenDCL Programm"
#define MyAppVersion "Mein OpenDCL Programm 1.5"
#define MyAppPublisher "ACAD Add-Ons & Co."
#define MyAppURL "http://www.example.com/"

[Setup]
; NOTE: The value of AppId uniquely identifies this application.
; Do not use the same AppId value in installers for other applications.
; (To generate a new GUID, click Tools | Generate GUID inside the IDE.)
AppId={F85796E5-4DED-4CC5-8E14-66D1BFD9950C}
AppName={#MyAppName}
AppVersion={#MyAppVersion}
;AppVerName={#MyAppName} {#MyAppVersion}
AppPublisher={#MyAppPublisher}
AppPublisherURL={#MyAppURL}
AppSupportURL={#MyAppURL}
AppUpdatesURL={#MyAppURL}
DefaultDirName={pf}\{#MyAppName}
DefaultGroupName={#MyAppName}
OutputBaseFilename=OpenDCLProgrammSetup
Compression=lzma
SolidCompression=yes

[Languages]
Name: "german"; MessagesFile: "compiler:Languages\German.isl"

[Files]
Source: lspfiles\Mein OpenDCL Programm.lsp; DestDir: {app}; Flags: ignoreversion
Source: lspfiles\Mein OpenDCL Programm.odcl; DestDir: {app}; Flags: ignoreversion
Source: lspfiles\OpenDCL.Runtime.6.0.2.3.msi; DestDir: {app}; Flags: ignoreversion

[Run]
Filename: msiexec.exe; Parameters: "/i ""{app}\OpenDCL.Runtime.6.0.2.3.msi"" /qn"

[UninstallRun]
Filename: msiexec.exe; Parameters: "/x ""{app}\OpenDCL.Runtime.6.0.2.3.msi""

1: 1 | Insert
```

Abbildung 12. Das Installationskript nach dem Editieren.

Korrigieren Sie Ihr Skript entsprechend der Abbildung, falls nötig. **Sichern Sie das Skript.**

Nun können Sie das Skript testen. Drücken Sie die Taste **F9** oder klicken Sie die Bildschaltfläche Run . Für den theoretischen Fall, dass die Dateien im in dem Abschnitt **[Files]** angegebenen Verzeichnis nicht existieren, wird der Kompilervorgang fehlschlagen, eine Meldung ausgeben und die entsprechende Zeile hervorheben.

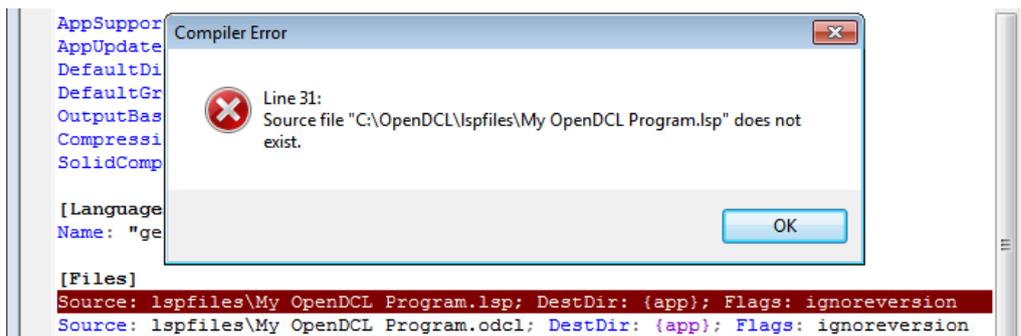


Abbildung 13. Der Inno Setup Compiler findet einen Fehler

Sie können sehen, wie der Compiler zeilenweise das Skript abarbeitet, wird – wie oben – ein Fehler gefunden, wird der Vorgang gestoppt und die Zeile hervorgehoben, die den Fehler verursacht. Wie bereits vermutet, fehlen hier die kopierenden Dateien. Haben Sie alle Dateien in die entsprechenden Verzeichnisse kopiert oder etwaige Schreibfehler korrigiert, läuft der Compiler korrekt durch.



Abbildung 14. Der Inno Setup Compiler nach dem erfolgreichen Erstellen der Anwendung.

Anschließend wird das Installationsprogramm automatisch gestartet. Es wird dringend empfohlen, das Installationsprogramm vor der Weitergabe zu testen. Sie können entweder das Installationsprogramm weiter ausführen, um das Ergebnis zu kontrollieren.

Eine andere Möglichkeit besteht darin, im Debugging-Verfahren die Ausführung des Installationskripts Zeile für Zeile zu verfolgen, so wie Sie es aus VisualLISP und anderen Entwicklungsumgebungen kennen. Mit Hilfe von Haltepunkten, die Sie mit der Funktionstaste F5 ein- bzw. ausschalten, können Sie den Ablauf stoppen. Mit den Funktionstasten F8 und F7 können Sie zeilenweise oder sogar innerhalb der Zeile springen und den Ablauf verfolgen.

Starten Sie den Vorgang durch Drücken der Funktionstaste F9 oder F4, wenn der Programmablauf bis zu der Zeile, in der der Cursor sich gerade befindet fortschreiten soll.

Zunächst werden die Voreinstellungen gesetzt, dann wird der Willkommens-Dialog angezeigt:

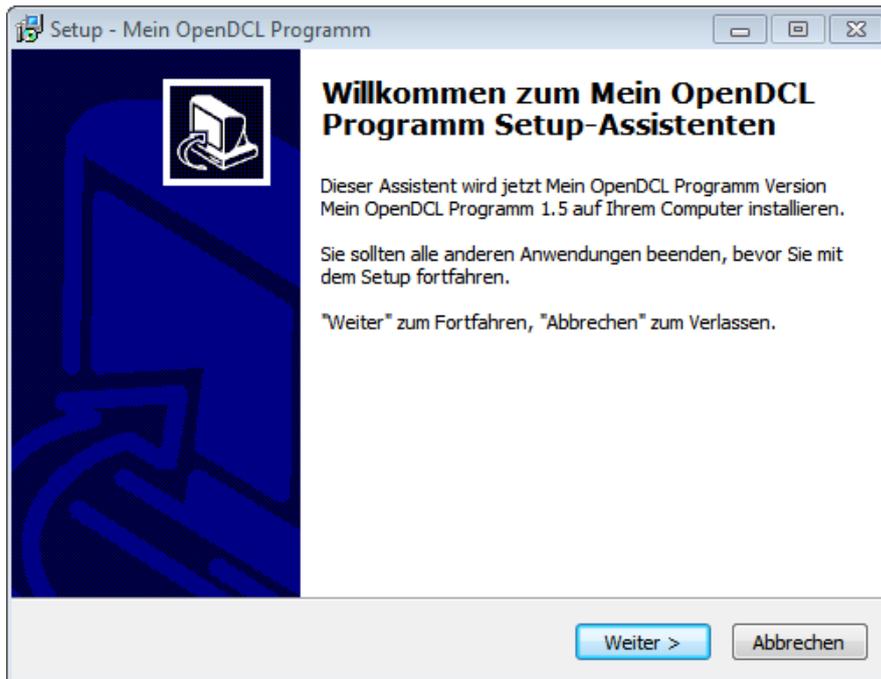


Abbildung 15. Der Willkommens-Dialog des Installationsprogramms.

Sie erkennen, dass in dem Dialog Ihre Programm-Informationen eingetragen sind. Klicken Sie auf die Schaltfläche **Weiter**.

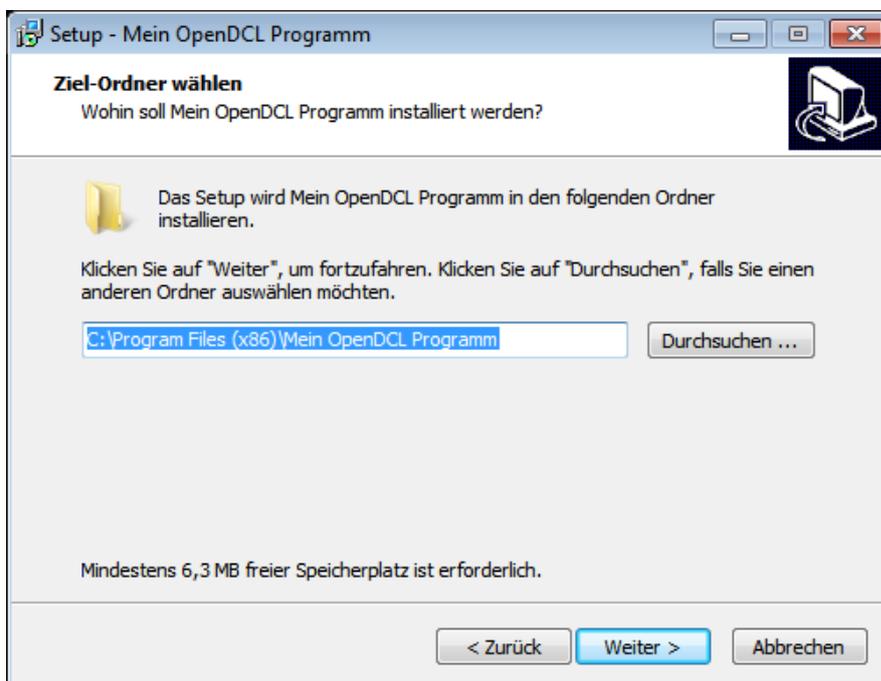


Abbildung 16. Ziel-Ordner der OpenDCL-Anwendung wählen.

Aufgrund der Einstellungen im Skript kann der Anwender ein freies Zielverzeichnis der Anwendung wählen. Ist die Auswahl korrekt, klicken Sie die Schaltfläche **Weiter**.

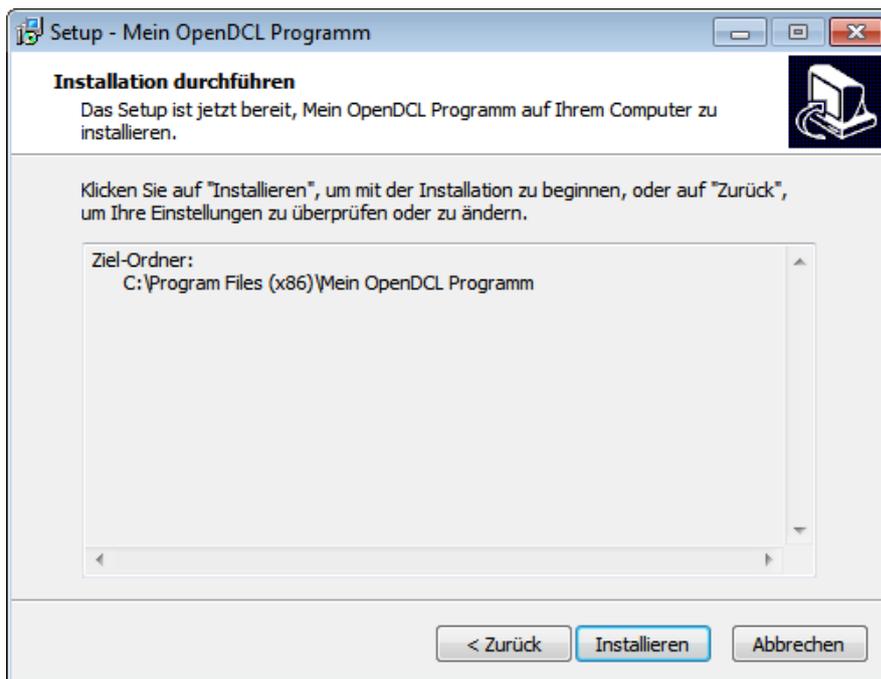


Abbildung 17. Übersicht der Änderungen im Rahmen des Installationsprogramms

Klicken Sie auf die Schaltfläche **Installieren**, wird der Skript-Editor des Compilers in den Vordergrund geholt und die erste Zeile des Abschnitts **[Files]** markiert.

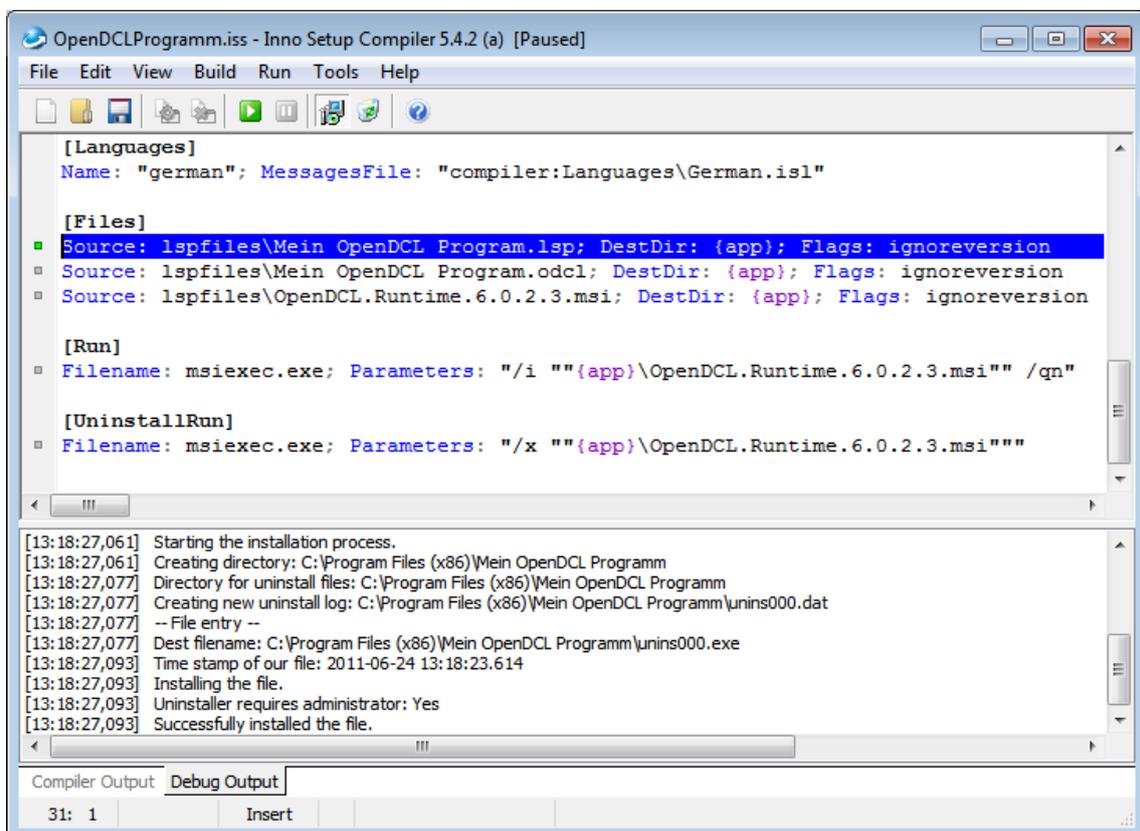


Abbildung 18. Der Inno Setup Compiler läuft durch den Quellcode.

Mit den Funktionstasten F7 und F8 können Sie Prozedur- oder Einschnitte machen und überprüfen, ob der jeweilige Vorgang erfolgreich war. Ist die Installation erfolgreich, erhalten Sie folgenden Abschlussdialog.



Abbildung 19. Der Abschlussdialog des Installationsprogramms.

Jetzt können Sie Ihre Applikation weitergeben!