

# Starthilfe für Visual Eiffel

---

## Inhaltsverzeichnis

Allgemeines .....	2
Bezugsquelle.....	2
Lizenz .....	2
Installation.....	2
Installation der Entwicklungsumgebung .....	2
Speicher für die Konfiguration der IDEs .....	3
Arbeit mit Projekten .....	4
Anlegen eines neuen Projekts .....	4
Projektoptionen.....	6
Beispiel Code: „Hello World“ .....	6
Arbeit mit der Entwicklungsumgebung.....	7
Ausführen der .exe .....	7
Klassenbrowser .....	8
Mehrere Klassen verwenden .....	8
Übersetzen .....	8
Bekannte Fehler .....	9

## Allgemeines

Visual Eiffel ist eine integrierte Entwicklungsumgebung für die Programmiersprache Eiffel. Sie enthält eine Eiffel-Bibliothek, einen Übersetzer und einen Editor. Die Installation benötigt Administrationsrechte auf dem lokalen Rechner.

## Bezugsquelle

Visual Eiffel wird von der Firma Object Tools zur Verfügung gestellt. Deren Webseiten sind leider schlecht gepflegt. Deshalb bitte nur diese Anleitung als Referenz benutzen.

Die aktuelle Installationsdatei für Windows ist die Version 5.0b Build 2508 (16.06.2005). Sie ist über diese URL direkt zu erreichen:

<http://tools.otsys.de/vsetup5-1.exe>

Als allgemeinen Zugang empfehle ich diese URL: <http://visual-eiffel.org>. Dort sind alle nötigen Referenzen angegeben.

## Lizenz

Visual Eiffel ist für „nicht kommerzielle Projekte“ kostenlos, bedarf jedoch trotzdem eines Lizenzschlüssels, der über den folgenden Link bezogen werden darf:

<http://tools.otsys.de/ve.key>

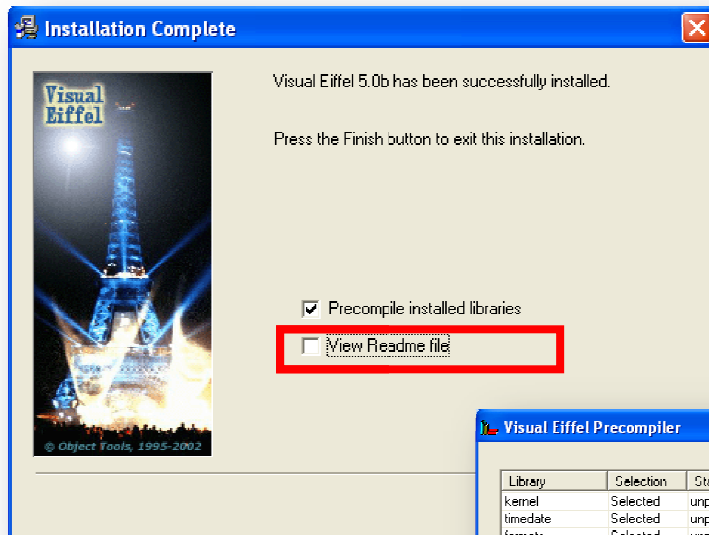
## Installation

Die Installation von Visual Eiffel verläuft in zwei Schritten:

- **Installation der Entwicklungsumgebung durch Ausführen der Datei**  
vsetup5-1.exe
- **Kopieren des Lizenzschlüssels in das Installationsverzeichnis der Entwicklungsumgebung**  
...\Object Tools\Visual Eiffel\bin

## Installation der Entwicklungsumgebung

Nach der standardisierten Installationsroutine werden Sie gefragt, ob Sie die installierten Bibliotheken vorkompilieren möchten.

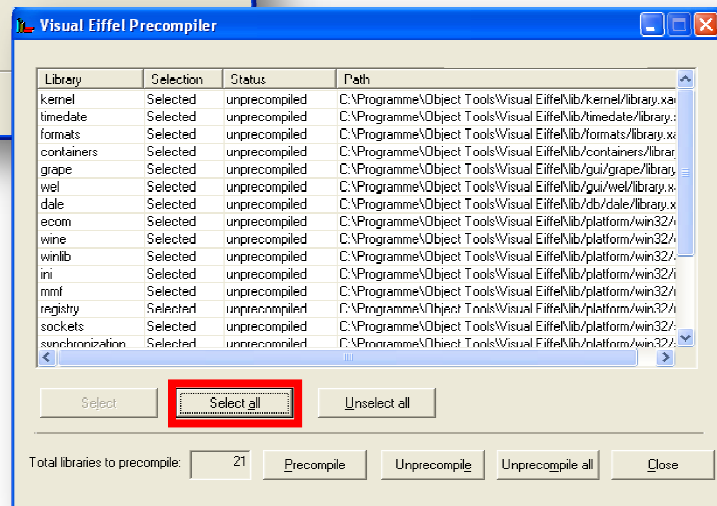


Dieses Dialogfenster wird mit

**„FINISH“**

bestätigt um zu den  
 „Precompiler“ Einstellungen zu  
 wechseln.

Bei den „Precompiler“-Einstellungen sind in der Regel keine Bibliotheken ausgewählt. Belässt man diesen Zustand, so dauert der spätere Übersetzungsprozess unnötig lang. Aus diesem Grund klicken wir einmal auf **„SELECT ALL“** und dann auf **„PRECOMPILE“**. Damit starten wir den Vorkompilierungsprozess, der je nach Rechnerleistung relativ lange dauern kann.



### Speicher für die Konfiguration der IDEs

Das RZ hat für jeden Windows Benutzer einen Bereich eingerichtet, der in Grenzen benutzbar für Konfigurationen der Anwendungen ist. Dieser Bereich soll mit **U:\** zugeordnet werden. Visual Eiffel ist so konfiguriert dass im Labor 3-201 dieses Laufwerk benutzt wird, um globale Konfigurationen des Benutzerapplikationen abzulegen. Dieses Laufwerk wird beim Login automatisch verbunden:

U:\inf3praktikum\eiffel\data

**Dieser Arbeitsbereich muss existieren, sonst startet die IDE nicht!**



## Arbeit mit Projekten

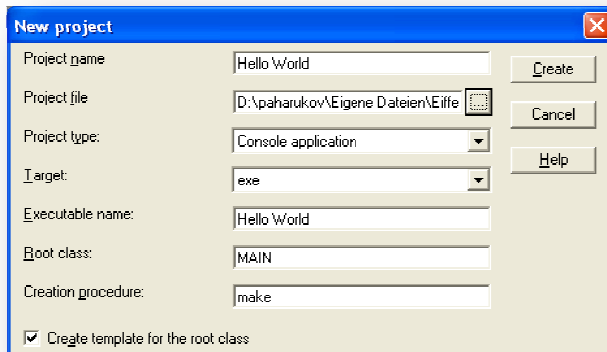
1. Für jedes Programm, das man entwickeln möchte, muss ein neues Projekt angelegt werden.
2. Jedes Projekt hat
  - a. Einen **Projektnamen** (Project Name)
  - b. Eine **Projektdatei** (Project File) mit einem Namen und Endung in folgender Form  
`Projektname.xacc`
  - c. Einen **Ordner** mit einem Namen
  - d. Eine **Wurzelklasse** (Root Class) mit einem Namen. Bei dieser Wurzelklasse beginnt der Programmablauf
  - e. Eine **Erzeugungsprozedur** (Creation procedure ) mit einem Namen. Diese wird bei Programmstart an erster Stelle ausgeführt.
  - f. Eine **Ausführbare Code-Datei** (Target) mit einem Namen (Executable Name) und Endung in folgender Form:  
`Projektname.exe`

Der Programmierer muss sich also zunächst **sechs (!)** Namen überlegen, bevor er mit der Arbeit beginnen kann. Deshalb ein Vorschlag:

- **Vier Namen gleich**
  - o `Projektname = Verzeichnisname = Executablename = Projektdateiname`
- **Wurzelklasse**
  - o `<Projektname>` oder `MAIN`
- **Erzeugungsprozedur**
  - o `make`

## Anlegen eines neuen Projekts

Zum Anlegen eines neuen Projekts in Visual Eiffel klickt man am besten auf „PROJECT“ -> „NEW“. Darauf öffnet sich eine Dialogbox, in der man die Grundinformationen zu einem Projekt angeben kann.



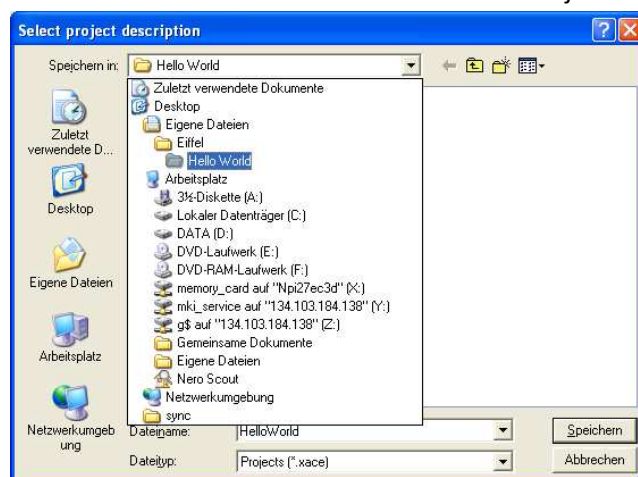
Die Dialogbox enthält mehrere Eingabefelder die **alle ausgefüllt** werden müssen.

- Unter „**Project Name**“ und „**Executable Name**“ den gewünschten Projektname eintragen
- Unter „**Project File**“ muss
  - o entweder der gesamte Pfad in der Form :

**D:\paharukov\Eigene Dateien\Eiffel\HelloWorld\HelloWorld.xace**

(also auch mit Dateinamen am Ende) eingetragen werden

- o oder man klickt auf „...“, es öffnet sich ein Dialogfenster, in dem man den gewünschten Ordner auswählen und den Namen der Projektdatei eingeben kann:



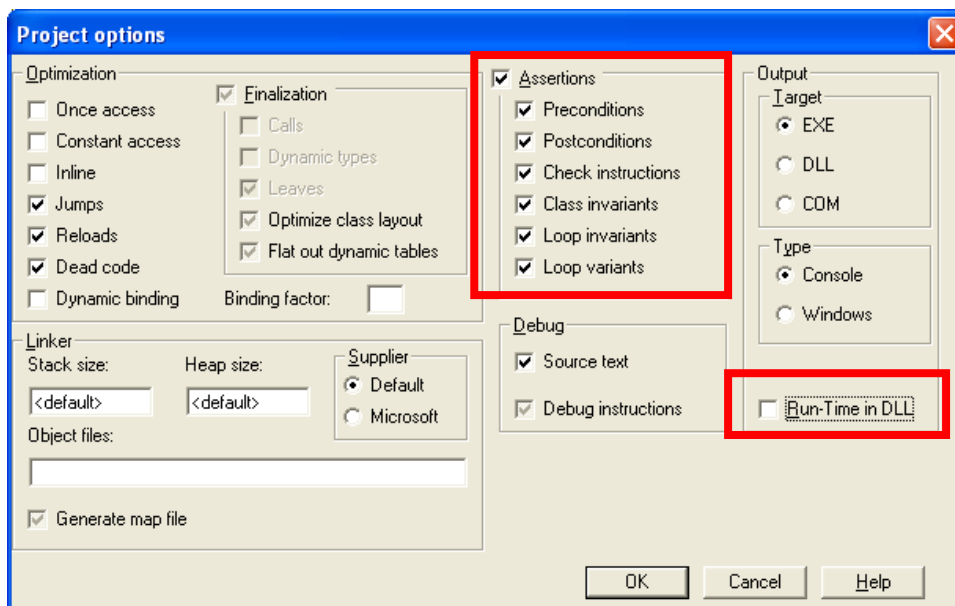
- „**Project type**“ gibt die Anwendungsart an. Für I3P genügt „**Console application**“
- „**Target**“ gibt die Codeart, ob .exe , .com oder .dll, an. Für das I3P ist .exe angebracht
- Durch das Aktivieren von „**Create template for the root class**“ wird die Wurzelklasse nach einem Muster erstellt, an dem man sich orientieren kann.

Wenn alles eingetragen ist, „**Create**“ anklicken.

## Projektoptionen

Bevor man mit der Arbeit anfängt sollte man für jedes neue Projekt die Projektoptionen einstellen. Das Dialogfenster für die Projektoptionen wird mit Klick auf „Project“ -> „Options...“ geöffnet. Die Einstellungen müssen wie auf dem folgenden Bild vorgenommen werden.

Die Änderungen sind hervorgehoben



Alle Zusicherungen sollten zum Testen angeschaltet sein.

Die Laufzeitumgebung (Run-Time) soll zur exe-Datei dazugebenden werden, sonst wird zur Laufzeit nach einer nicht vorhandenen DLL gesucht.

Nach diesem Schritt ist die Entwicklungsumgebung und das Projekt fertig konfiguriert.

## Beispiel Code: „Hello World“

Den folgenden Code können Sie zu Testzwecken in ein Projekt einfügen.

### indexing

description: *"Simple example for standard output"*  
 author: *"Mrs. Anybody"*  
 history: *"07-11-19"*

class MAIN

### create

make

```
feature {NONE}  
    Message : STRING is "Hello World"
```

```
feature  
    make is  
        -- Print current object with a simple message in a line.  
        do  
            print (Current)  
            io.put_string (" says: " + Message)  
            io.put_new_line  
        end -- make
```

```
end -- class MAIN
```

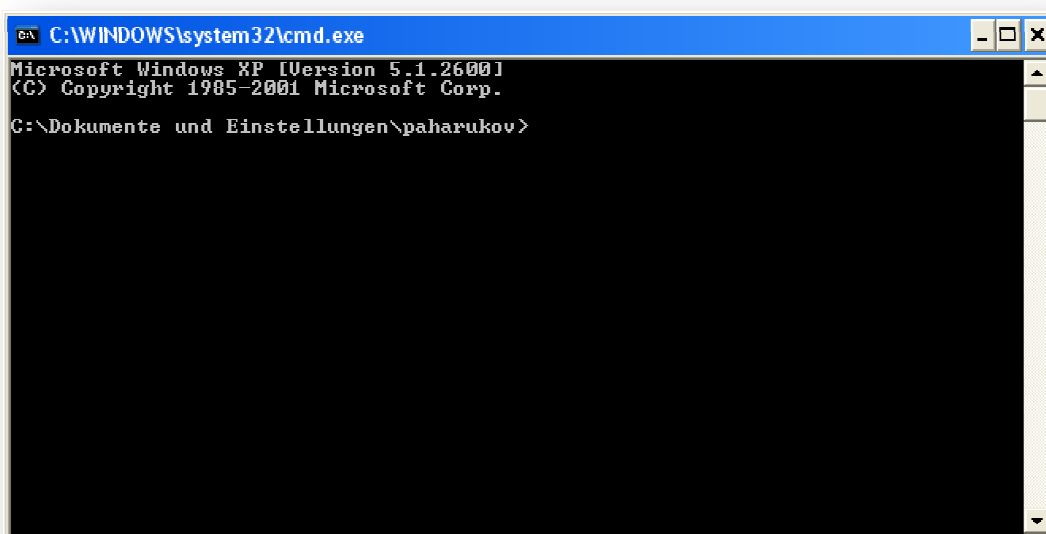
## Arbeit mit der Entwicklungsumgebung

### Ausführen der .exe

Starten Sie Ihre Konsolenanwendung immer aus der Eingabeaufforderung heraus. Dies hat den Vorteil, dass Sie zum einen mit der Argumentenübergabe arbeiten können und zum anderen auch dann die Ausgabe sehen, wenn die Anwendung unerwartet beendet wird. Zum Starten der Eingabeaufforderung START -> AUSFÜHREN. In dem geöffneten Eingabefenster geben Sie:

„cmd“

ein, um die Konsole zu starten.

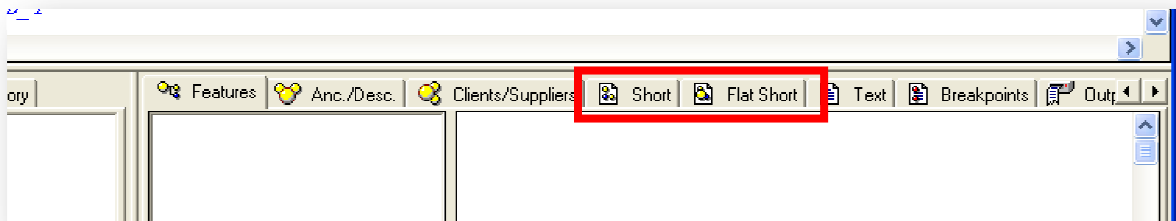


Hier navigieren Sie nun zu Ihrem Projektordner und führen die entsprechende .exe aus. Die wichtigsten Befehle auf der Konsole lauten.

<code>cd „C:/Ordner/Ordner2“</code>	Navigiert direkt zu dem angegebenen Pfad. Sollten im Pfad Leerzeichen vorhanden sein, muss er in „“ angegeben werden
<code>cd ..</code>	Springt in der Ordner-Hierarchie um eine Stufe zurück. Im oberen Beispiel also in „Ordner“
<code>dir</code>	Zeigt den Inhalt des aktuellen Ordners oder des Laufwerks an
<code>cd &lt;TAB-drücken&gt;</code>	Schreibt die Ordnernamen zum auswählen. Spart das Tippen.

## Klassenbrowser

Der Klassenbrowser liefert Informationen zu den Klassenschnittstellen, die während der Implementierung benötigt werden. Dieser ist in der Entwicklungsumgebung eingebaut und verbirgt sich hinter den Begriffen „Short“ und „Flat Short“ im Unteren Bereich des Visual Eiffel Fensters.



Um Informationen zu erhalten, markieren Sie einen Klassennamen, z.B. den Typ „STRING“, und ziehen Sie ihn per Drag & Drop in das mittlere von den drei unteren Fenstern.

## Mehrere Klassen verwenden

Wenn Sie in Ihrem Projekt mit mehreren Klassen arbeiten, beachten Sie, dass die neu erstellten Klassen nur dann im Projekt erscheinen, wenn sie auch im Projekt verwendet werden.

## Übersetzen

Kompilieren sollte man ein Projekt grundsätzlich mit dem Befehl „**Rebuild all**“, zu finden unter „**Project**“->“**Rebuild all**“ oder Tastenkombination „**CTRL**+“**SHIFT**+“**F7**“.



**INF**

Studiengang  
Medien- und  
Kommunikationsinformatik



**Hochschule Reutlingen**

Reutlingen University

Medien- und Kommunikationsinformatik (B.Sc.)  
Alexander Paharukov

**Informatik 3 Praktikum**

## **Bekannte Fehler**

### **Fehler beim Angelegen eines Projektes. Beschädigte Projektdatei!**

Dieser Fehler wird meist dann angezeigt wenn man die Projektdateien in ein anderes Verzeichnis verschiebt. Zum beheben des Fehler führen Sie folgende Schritte durch:

1. Entwicklungsumgebung beenden
2. Inhalt des „**data**“ Ordners löschen
3. Entwicklungsumgebung starten
4. Projekt öffnen

**Vorschläge zur Verbesserung dieser Starthilfe nehme ich gerne entgegen.**