



Lazarus 1.9.0

Installieren und fürs
Windows-Cross-Compiling einrichten

Autor : Heiko Rompel

Version 1.6

1. **Lazarus 1.9.0** unter **Linux Mint 18.2** installieren
und fürs **Windows**-Cross-Compiling einrichten

Einleitung

In dieser Anleitung wird erklärt welche Schritte nötig sind um unter **Linux Mint 18.2** die Entwicklungsoberfläche **Lazarus 1.9.0** zu installieren.

Des weiteren wird beschrieben wie man **Lazarus 1.9.0** und **Freepascal 3.0.2** einrichten muss, um damit auch Programme für **Windows 32-Bit** und **Windows 64-Bit** Oberflächen zu erstellen.

Ohne einen Artikel im **Blaise Pascal Magazin 10/2015**

<http://www.blaisepascal.eu/> ,

der Hilfe der Mitglieder im **Freepascal-Forum**

<http://forum.lazarus.freepascal.org/index.php/topic,31515.0.html>

und dem deutschen **Lazarus-Forum**

<http://www.lazarusforum.de/viewtopic.php?f=16&t=9411>

hätte ich es nie geschafft.

Ich hoffe, Ihr könnt den Anweisungen folgen und kommt damit auch ans Ziel.

Vielleicht, liefert ja mal jemand auf Basis dieser Anleitung eine Anleitung um **Android**-Apps, **iOS**-Programme und so weiter zu erstellen.

Bremerhaven, den 13.07.2017

Inhaltsverzeichnis

Einleitung.....	2
Lazarus installieren.....	4
Das Script für Lazarus.....	4
Nach dem Script.....	8
Crosscompiling für Windows 32-Bit einrichten.....	12
Lazarus für Windows 32-Bit und 64-Bit einrichten.....	14
1. Hinzufügungen und Beeinflussungen.....	14
Der Erstellmodus.....	14
2. Pfade.....	19
3. Konfiguration und Ziele.....	20
Für Win32-Anwendungen.....	20
Für Win64-Anwendungen.....	21
Crosscompiling für Windows 64-Bit einrichten.....	22
WINE in Lazarus einbinden.....	23
Debugger Fehlermeldung.....	25
Erstellmodus in der IDE ändern.....	26
Crosscompiling Linux 64 nach Linux 32.....	26
Schlusswort.....	27
Copyright-Hinweise.....	27
Danksagung.....	27

1. **Lazarus 1.9.0** unter **Linux Mint 18.2** installieren und fürs **Windows-Cross-Compiling** einrichten

Lazarus installieren

Ich gehe davon aus, dass **Linux Mint 18.2 schon fertig installiert ist**.

Als nächstes müssen Ihr also Lazarus installieren.

1. Ihr nutzt hierfür das nachfolgendes Script das ursprünglich aus dem **Blaise Pascal Magazin 10/2015** stammt.
(Das Script wurde schon auf Lazarus 1.9 umgestellt.)

Markiert Euch das Script und fügt es in eine Textdatei ein.
Benennt die Datei in getlaz.sh.

Ihr startet das Script in einem Terminal-Fenster (in Eurem Home-Ordner) mit dem Befehl:
sudo sh getlaz.sh

Anmerkungen und Änderungen von mir sind mit „(HR)“ gekennzeichnet.

Nach dem Start des Scriptes kommt die Frage ob „/usr“ oder „/usr/local“.
Bitte den Default-Wert „/usr“ übernehmen.

Übernehmt Ihr den Default-Wert nicht, müsst Ihr im weiteren ALLE „/usr“ durch „/usr/local“ ersetzen!

Das Script für Lazarus

```
#!/bin/bash
#
# run in your HOME-directory with: sudo sh getlaz.sh (HR)
#
# (C)by: Michael Van Canneyt (michael@freepascal.org)
#
# Release from: 2017-07-12 (HR)
# additional comments from: Heiko Rompel (HR)
#####
# Some variables. Set this to whatever you want
#####
#
# Where to download/install everything ? (below home directory)
#
INSTALLDIR=fpc-install
#
# Which FPC version to use ?
```

1. **Lazarus 1.9.0** unter *Linux Mind 18.2* installieren
und fürs **Windows**-Cross-Compiling einrichten

```
#
# information about the existing version you will find here: (HR)
# ftp://ftpmaster.freepascal.org/pub/fpc/dist/ (HR)
#
VERSION=3.0.2
CPUARCH=`uname -p`
#
# Install FPC/Lazarus as root ? (YES or NO)
#
USEROOT=YES
#
# Which lazarus version ?
# set either tag or branch variable.
# If neither is set, trunk is used.
# When lazarus 1.9 is out, this becomes lazarus_1_9
#
# information for the latest version you will find here: (HR)
# http://svn.freepascal.org/svn/lazarus/ (HR)
# TAG=lazarus_1_9/
TAG=
BRANCH=

#####
# No variables after this point.
#####
#
# Install preliminaries. This must be done as root.
#
sudo apt-get install subversion make binutils gdb gcc libgtk2.0-dev

#
# Check if the rest must be done as root.
#
if [ "$USEROOT" = YES ]; then
    SUDO=sudo
fi
#
#####
#####
# Get and install FPC.
#####
#####
#
# Create installation directory
#
mkdir ~/$INSTALLDIR
```

1. **Lazarus 1.9.0** unter *Linux Mind 18.2* installieren
und fürs **Windows**-Cross-Compiling einrichten

```
cd ~/$INSTALLDIR
#
# Fetch the necessary files for FPC.
#
wget ftp://ftpmaster.freepascal.org/pub/fpc/dist/$VERSION/source/fpc-
$VERSION.source.tar.gz
wget ftp://ftpmaster.freepascal.org/pub/fpc/dist/$VERSION/$CPUARCH-
linux/fpc-$VERSION.$CPUARCH-linux.tar
#
# extract installer.
#
tar xvf fpc-$VERSION.$CPUARCH-linux.tar
cd fpc-$VERSION.$CPUARCH-linux
#
# Install FPC (possibly as root)
#
$SUDO sh ./install.sh
#
# Extract sources.
#
cd ~
tar xvzf $INSTALLDIR/fpc-$VERSION.source.tar.gz
cd ~/$INSTALLDIR
#
#####
#####
# Get and install Lazarus
#####
#####
#
# Determine SVN url
#
BASEURL=http://svn.freepascal.org/svn/lazarus/
#
if [ ! -z "$TAG" ]; then
    SVNURL=$BASEURL/tags/$TAG
else
    if [ ! -z "$BRANCH" ]; then
        SVNURL=$BASEURL/branches/$BRANCH
    else
        SVNURL=$BASEURL/trunk
    fi
fi
#
# Check out sources
#
```

1. **Lazarus 1.9.0** unter *Linux Mind 18.2* installieren
und fürs **Windows**-Cross-Compiling einrichten

```
svn co $SVNURL lazarus
#
# Build the IDE
#
cd lazarus
make bigide
#
# Install lazarus (possibly as root)
#
$SUDO make install
#
# That's all folks !
#
# After install Lazarus Restart Linux.
```

1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren
und fürs **Windows**-Cross-Compiling einrichten

Nach dem Script

Auch wenn es ungewöhnlich für **Linux** ist, meldet Euch an dieser Stelle einmal ab und wieder an.

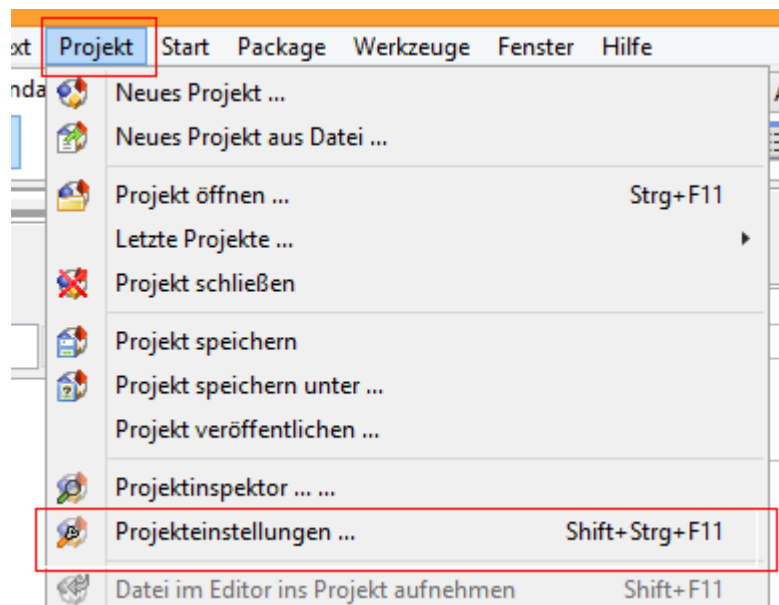
(Bei mir wurde sonst nicht die Programmgruppe „Entwicklung“ im Hauptmenü angelegt.)

Jetzt legen wir uns noch einen Starter für Lazarus auf den Schreibtisch.

Dazu geht Ihr in das Hauptmenü in die Programmgruppe „Entwicklung“ dann auf das Lazarus-Symbol. Nach einem „Rechts-Klick“ wählt Ihr den Punkt „Zum Scheibtisch hinzufügen“ aus.

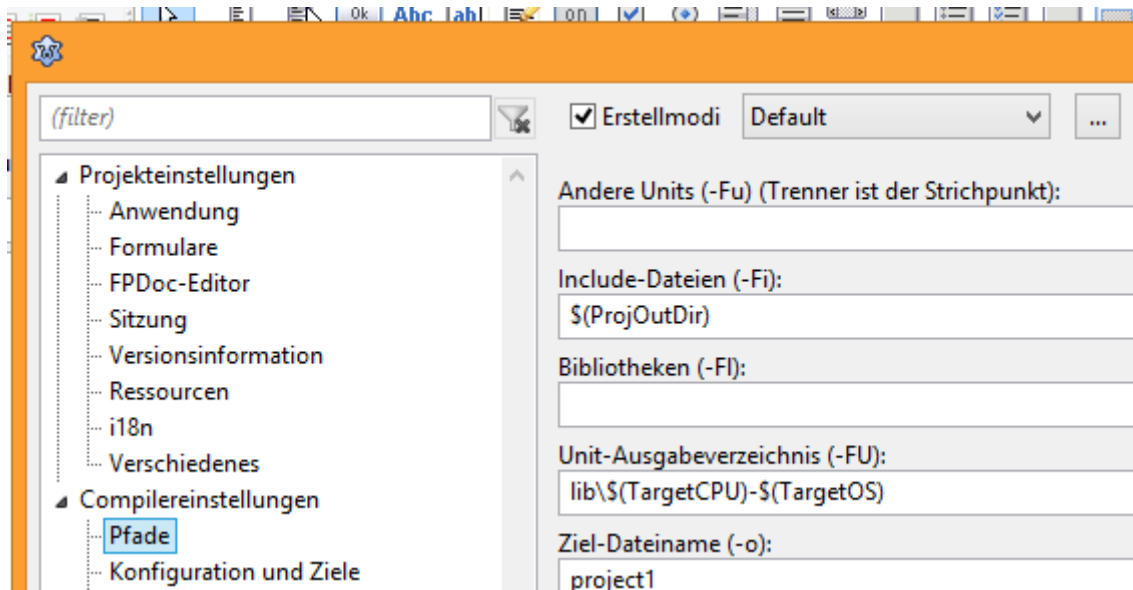
Nun starten wir Lazarus das erste Mal.

Nun wechselt Ihr in das Menü „Projekt“ und dort in „Projekteinstellungen ...“.

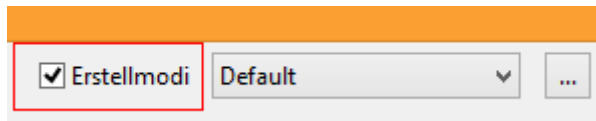


1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren
und fürs **Windows**-Cross-Compiling einrichten

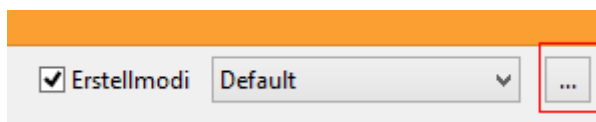
Hier wechselt Ihr nun in die „Compilereinstellungen \ Pfade“.



Rechts oben befindet sich die Checkbox „Erstellmodi“.

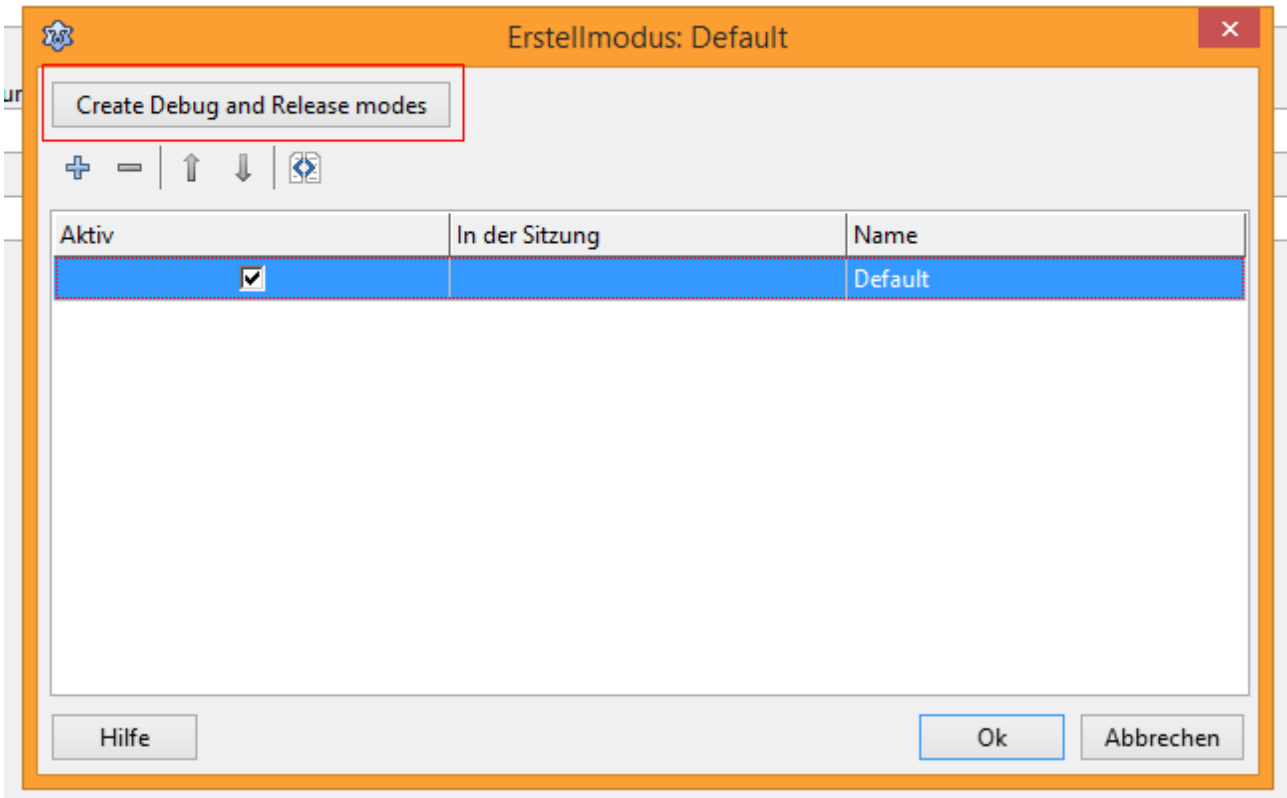


Bitte aktiviert diese jetzt.
Nun klickt Ihr auf die drei Punkte, die rechts sichtbar werden.

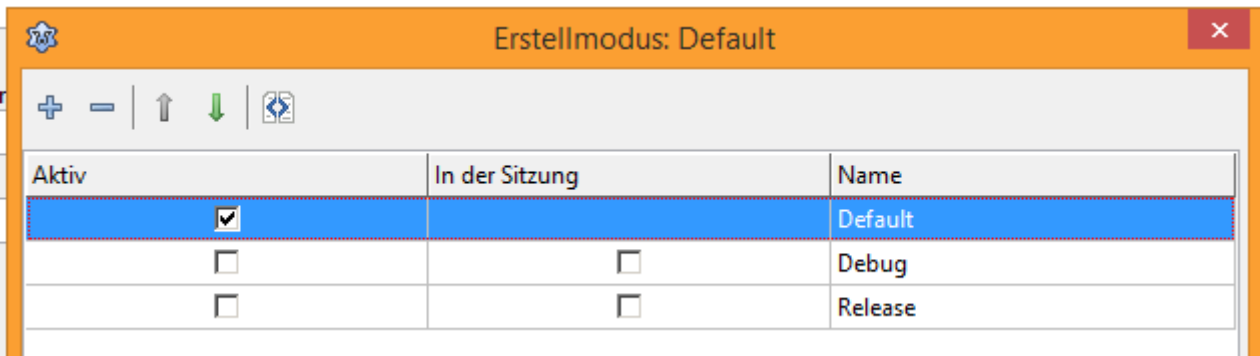


1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren
und fürs **Windows**-Cross-Compiling einrichten

Klickt den Button „Create Debug und Release Mode“.



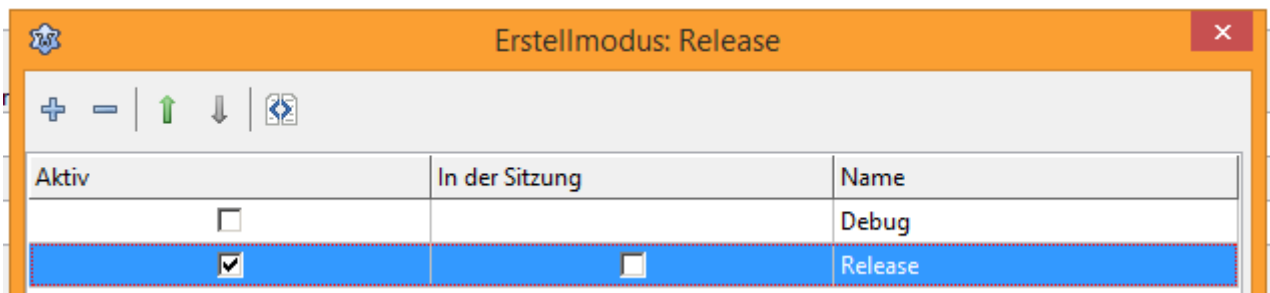
Markiert nun die Zeile mit dem Namen „Default“



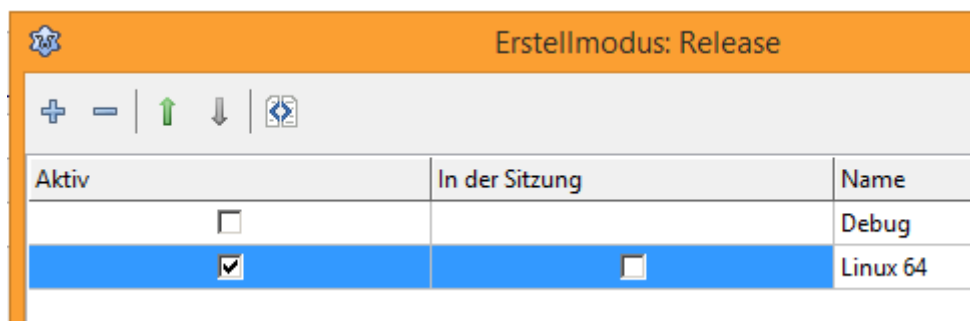
und löscht die Zeile mit einem klick auf das „Minus“-Zeichen.

1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren
und fürs **Windows**-Cross-Compiling einrichten

Führt jetzt in der Tabelle unten einen Doppelklick auf das Wort „Release“ aus.



Benennt den Punkt jetzt um in „Linux 64“ bzw. wenn Ihr ein 32-Bit Linux nutzt in „Linux 32“ um.



Das erleichtert später die Übersicht.

Nun schließt Ihr bitte Lazarus.

So, weiter geht es mit dem Einrichten.

1. **Lazarus 1.9.0** unter *Linux Mind 18.2* installieren
und fürs **Windows**-Cross-Compiling einrichten

Crosscompiling für Windows 32-Bit einrichten

So jetzt habt Ihr schon ein lauffähiges **Lazarus** mit dem Ihr unter **Linux** Programme erstellen könnt. Aber, Ihr wollt ja mehr.

Jetzt kommen die Informationen aus dem **Lazarus-Wiki**
http://wiki.lazarus.freepascal.org/Cross_compiling_for_Win32_under_Linux zum Einsatz.

Nachfolgende Zeilen (beginnend mit einem „\$“) werden nacheinander in einem Terminal-Fenster eingegeben, das in diesem Verzeichnis „/home/laz-user/fpc-3.0.2“ (**laz-user** gegen Deinen Usernamen ersetzen) gestartet wird :

Das „\$“-Zeichen wird nicht mit eingeben.

```
$ sudo make clean all OS_TARGET=win32 CPU_TARGET=i386
```

und dann

```
$ sudo make crossinstall OS_TARGET=win32 CPU_TARGET=i386  
INSTALL_PREFIX=/usr
```

Auch wenn man laut Wiki damit schon fertig ist, fehlen noch ein paar Anweisungen.

In der Datei „/etc/fpc.cfg“ müssen im Abschnitt

```
# -----  
# Set Filename und Paths  
# -----
```

noch diese Zeilen eingefügt werden:

```
# For Crosscompiling  
-Fu/usr/lib/fpc/$fpcversion/units/$fpctarget/*
```

Die Datei ist zumindest bei mir schreibgeschützt und kann nur in einem Terminal-Fenster geöffnet werden, das mittels „Als Systemverwalter öffnen“ geöffnet wurde.

Jetzt kommt noch ein symbolischer Link der in einem Terminalfenster eingegeben wird:

```
$ sudo ln -sf /usr/lib/fpc/3.0.2/ppcross386 /usr/bin/ppcross386
```

1. Lazarus 1.9.0 unter *Linux Mind 18.2* installieren und fürs **Windows**-Cross-Compiling einrichten

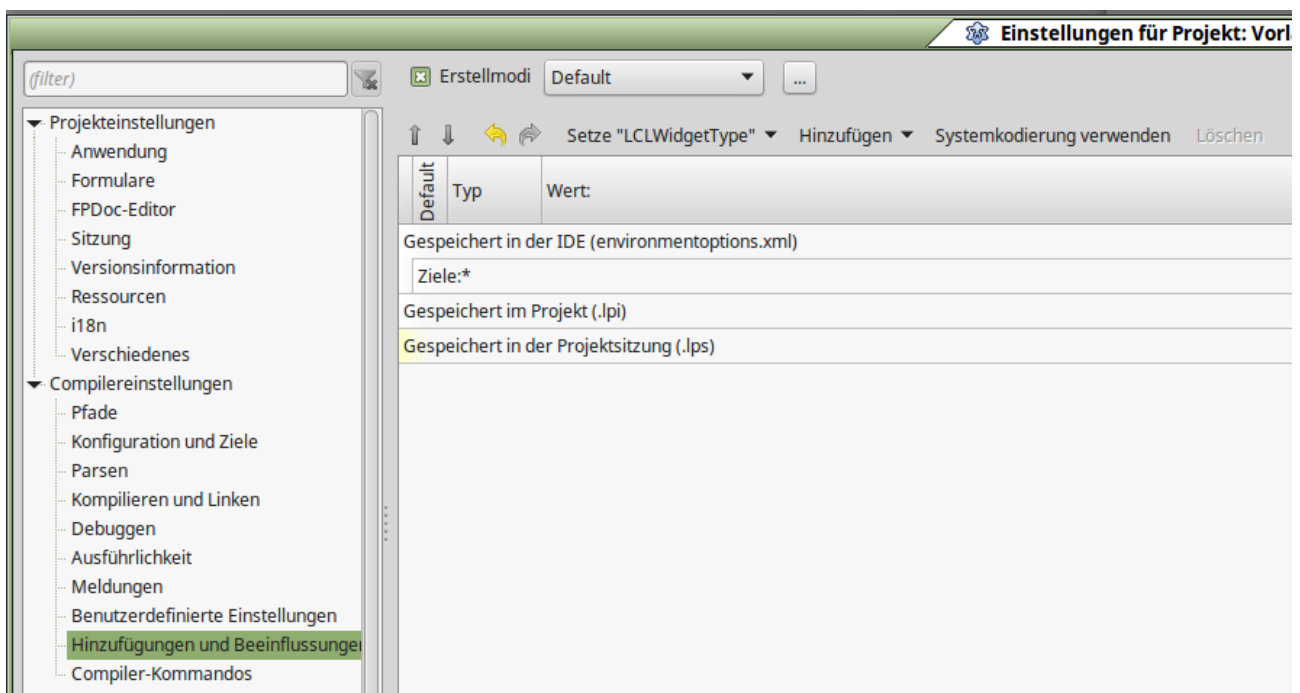
Lazarus für Windows 32-Bit und 64-Bit einrichten

Jetzt geht es mit der Einrichtung in Lazarus weiter.

Die nachfolgenden Einstellungen finden statt unter „Projekt / Projekteinstellungen“.

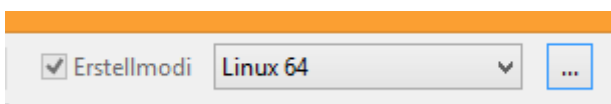
1. Hinzufügungen und Beeinflussungen

Wenn man diesen Punkt das erste Mal aufruft, dann sieht der so aus:



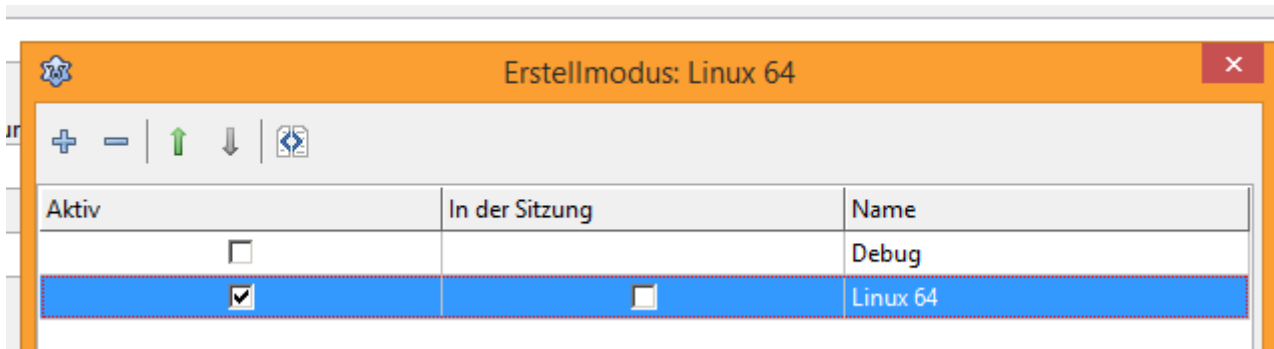
Der Erstellmodus

Wählt nun wieder die Einstellungen für den Erstellmodus aus (die 3 Punkte).



1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren
und fürs **Windows**-Cross-Compiling einrichten

Er erscheint wieder dieser Dialog:



Hier klicken Ihr jetzt auf das „+“-Zeichen.

Es erscheint eine neue Zeile.

In der Spalte „Name“ tragt Ihr „**Win32**“ ein.

Ihr erstellt noch eine neue Zeile und benennen sie „**Win64**“.

Jetzt habt Ihr vier „Erstellmodi“ zur Auswahl.

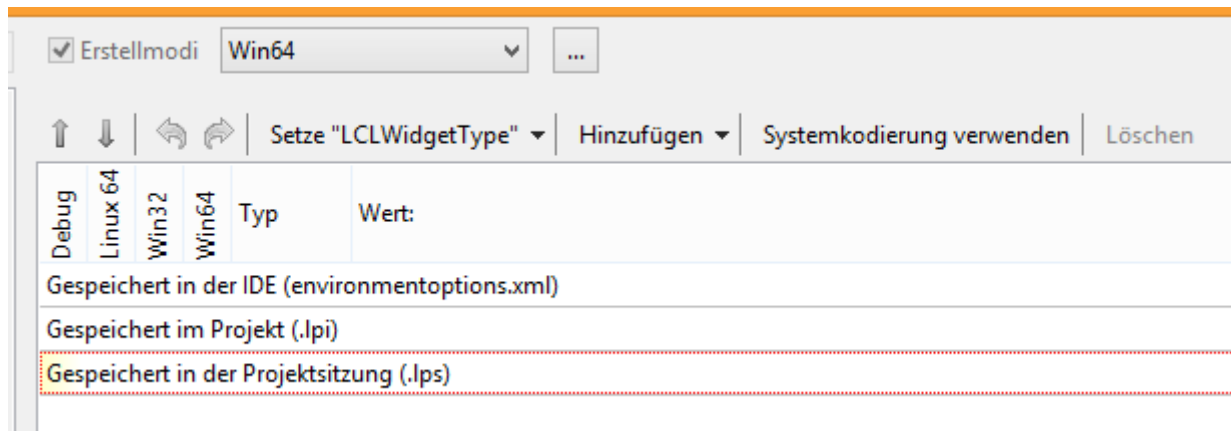
Linux 64 bzw Linux 32: Hier sind die Werte so wie direkt nach der Installation und in diesem Fall für das erstellen von Linux-Programmen.

Win32 und **Win64** sind selbsterklärend und werden nachfolgend mit weiteren Werten eingerichtet.

Und **Debug** – der Modus der beim Programmieren eigentlich immer aktiv ist.

1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren
und fürs **Windows**-Cross-Compiling einrichten

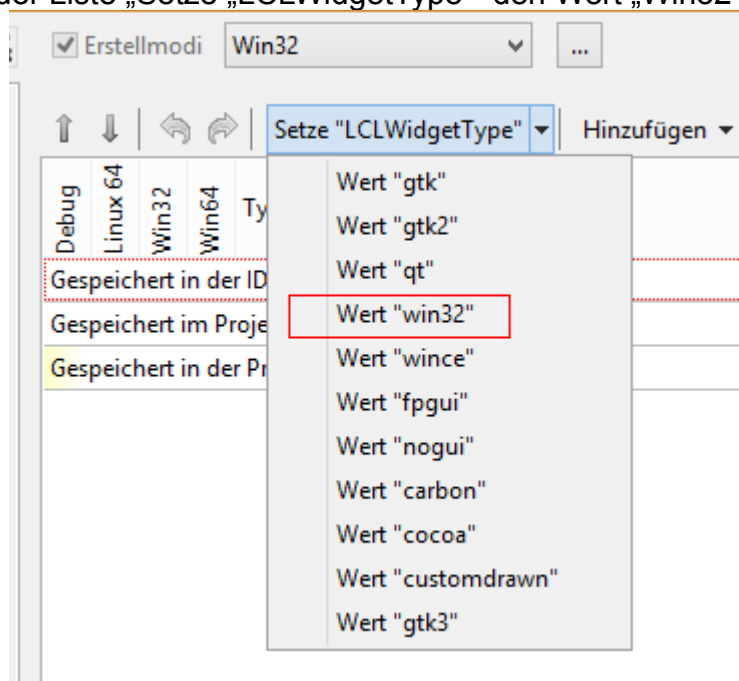
Unser Einstellungsdialog sieht jetzt so aus:



Jetzt müsst Ihr erst der „Erstellmodi“ „Win32“ auswählen.

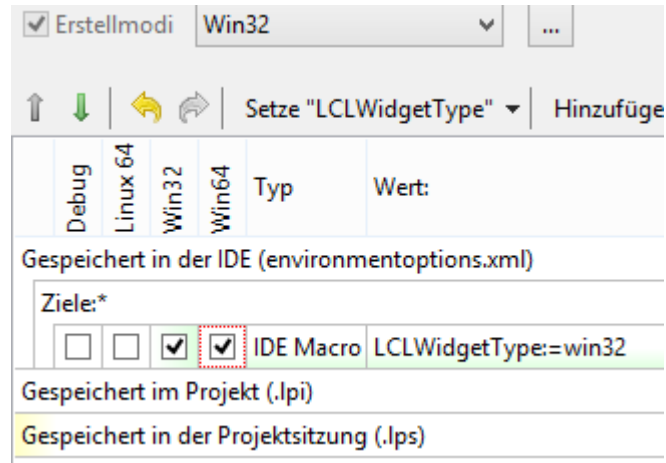
Die weiteren Einstellungen wollen wir in der IDE speichern.
Ihr klickt deshalb in die Zeile „Gespeichert in der IDE ...“.

Nun wählt Ihr aus der Liste „Setze „LCLWidgetType““ den Wert „Win32“ aus.

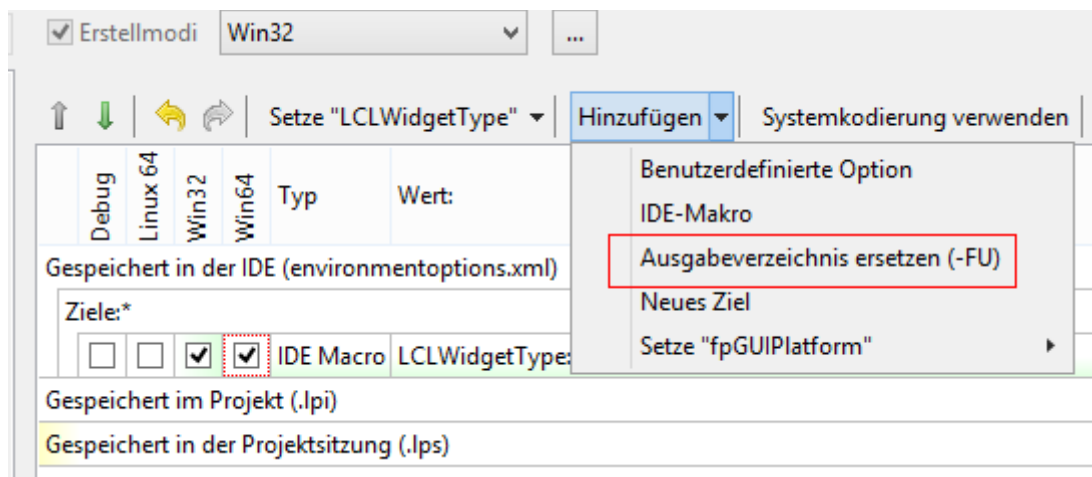


1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren
und fürs **Windows**-Cross-Compiling einrichten

Wenn Ihr den Wert ausgewählt habt, setzt Ihr in der Zeile je einen Haken bei „Win32“ und „Win64“. Das bedeutet, dass dieser Wert für beide „ErstellModi“ gilt.



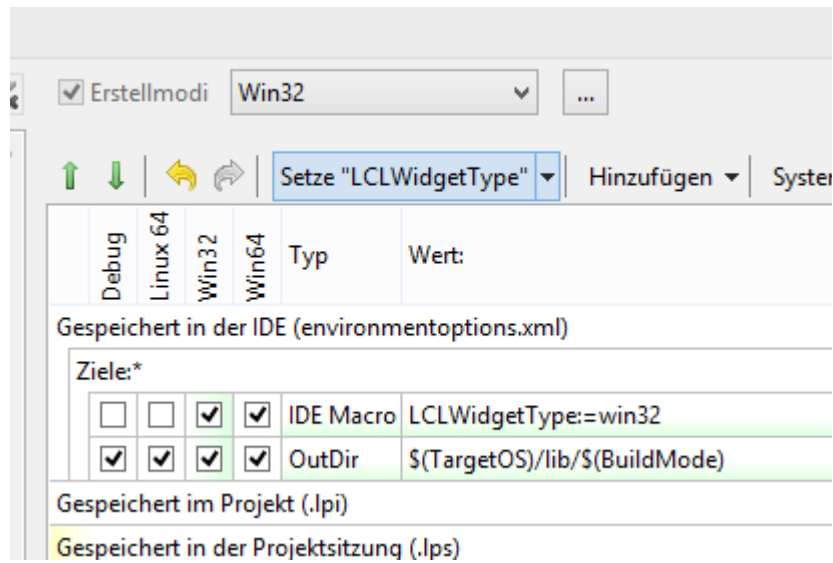
Damit Ihr nachher im Projektverzeichnis etwas mehr Übersicht habt, wählt Ihr jetzt noch aus der Liste „Hinzufügen „ den Punkt „Ausgabeverzeichnis ersetzen“



und füllt die Spalte „Wert“ mit : \$(TargetOS)/lib/\$(BuildMode).

1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren
und fürs **Windows**-Cross-Compiling einrichten

Eurer Dialog sollte jetzt so aussehen:

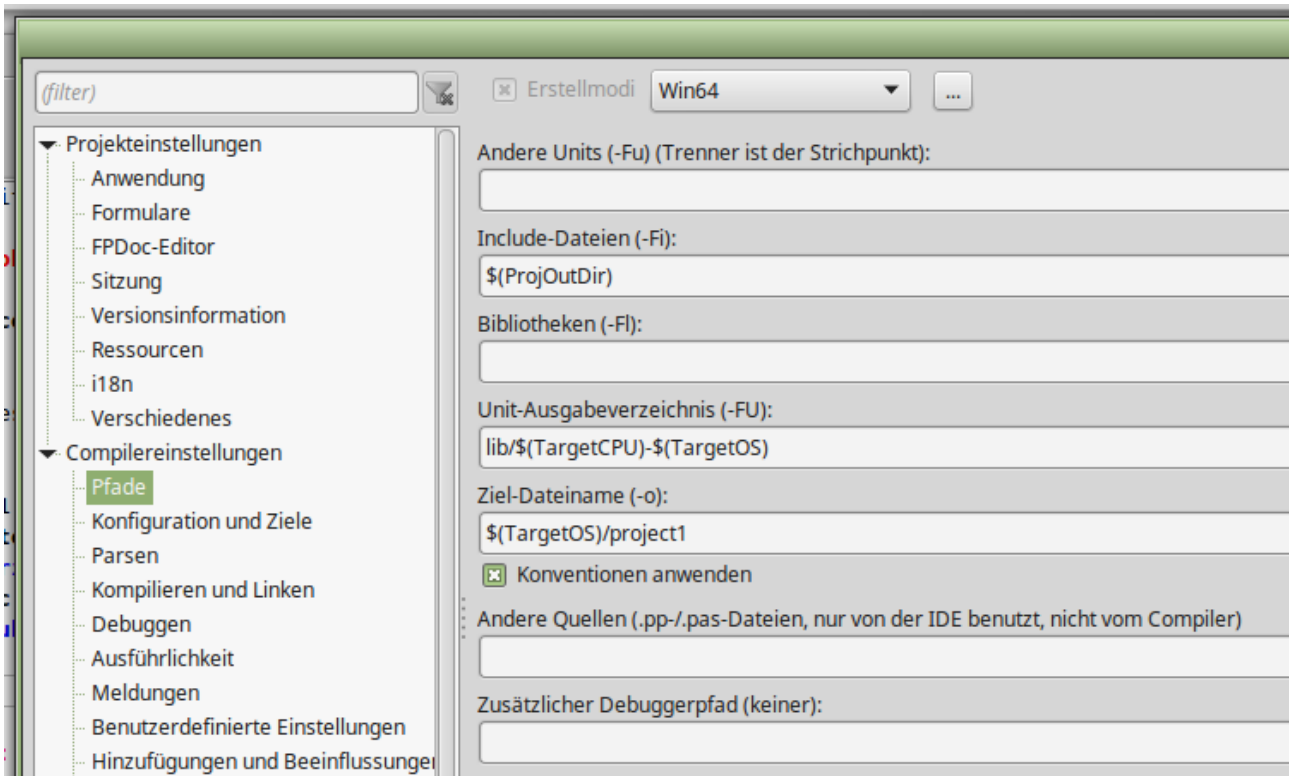


So wird für jedes Zielbetriebssystem und jeden Modus ein Unterverzeichnis erstellt.

Als nächstes wendet Ihr Euch den Pfad-Einstellungen zu.

1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren und fürs **Windows-Cross-Compiling** einrichten

2. Pfade



Hier wählt Ihr oben nacheinander alle „Erstellmodi“ aus und tragt in die Zeile „Ziel-Dateiname (-o)“ folgendes ein:

`$(TargetOS)/project1`

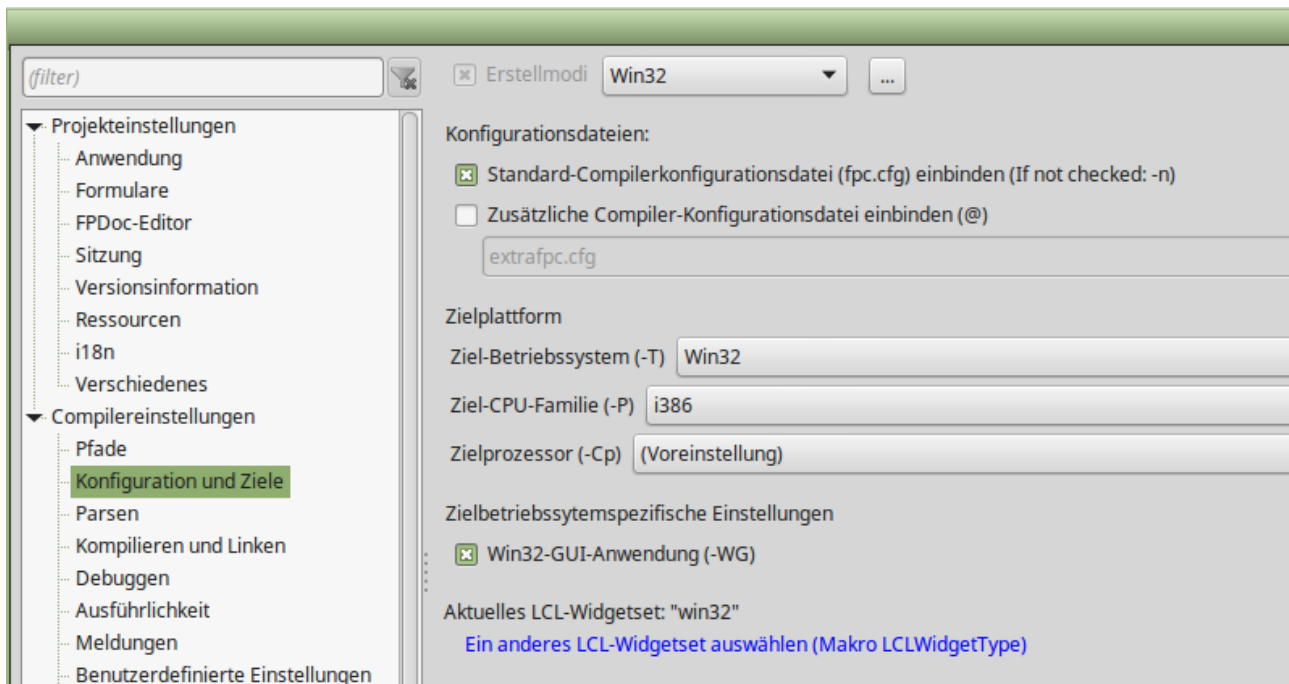
Hierdurch werden auch die erstellten, ausführbaren Dateien im Projektverzeichnis in entsprechende Unterverzeichnisse einsortiert.

1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren und fürs **Windows-Cross-Compiling** einrichten

3. Konfiguration und Ziele

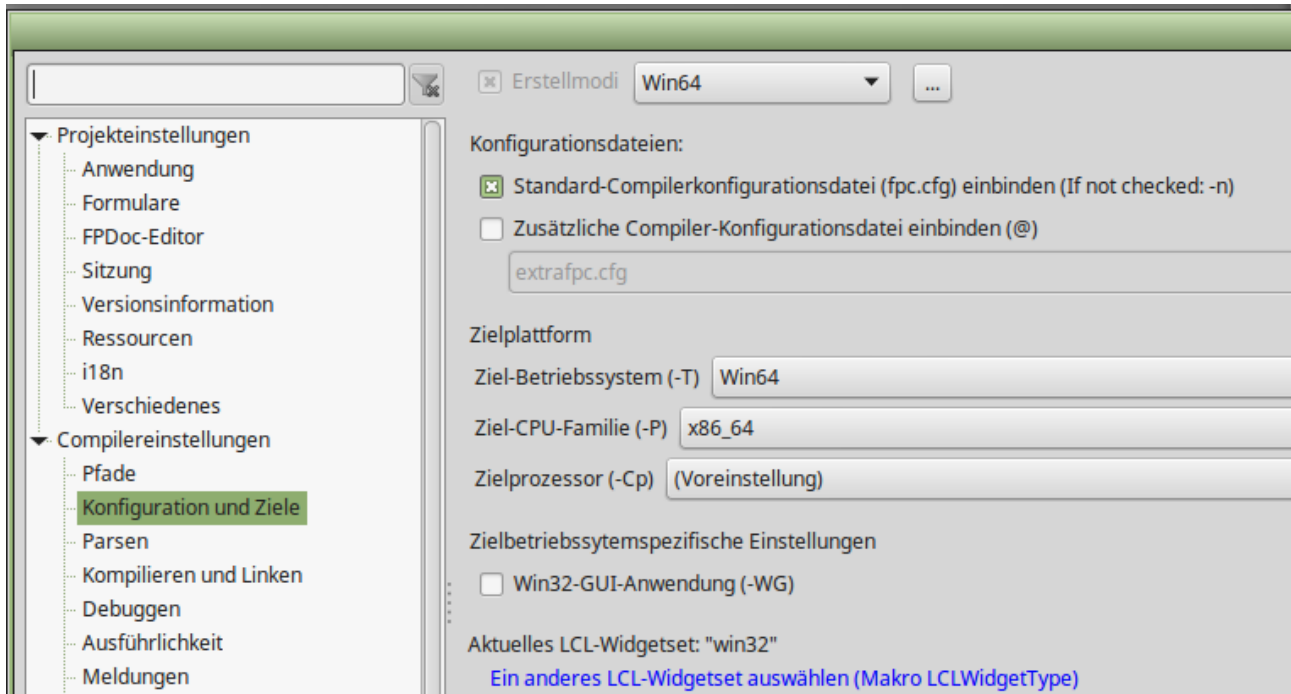
Diese Einstellungen sind nötig, wenn Ihr auf einem 32-Bit-System für ein 64-Bit-System oder umgekehrt, entwickelt.

Für Win32-Anwendungen



1. **Lazarus 1.9.0** unter **Linux Mind 18.2** installieren
und fürs **Windows**-Cross-Compiling einrichten

Für Win64-Anwendungen



Wenn Ihr alles richtig gemacht habt, könnt Ihr jetzt schon Programme für 32-Bit **Windows**-Systeme erstellen.

Die **Windows** Programme die Ihr unter **Linux** erstellt, werden natürlich nicht nach dem Erstellen in der **Linux**-Maschine ausgeführt.

Um **Windows**-Programme direkt unter Linux laufen zu lassen, braucht Ihr entweder **WINE** (dazu später mehr) oder ein echtes **Windows** in einer virtuellen Maschine.

So jetzt zu der Win64-Konfiguration.

1. **Lazarus 1.9.0** unter *Linux Mind 18.2* installieren
und fürs **Windows**-Cross-Compiling einrichten

Crosscompiling für Windows 64-Bit einrichten

Bitte Lazarus schließen.

Jetzt kommen die zusätzlichen Informationen aus dem *Lazarus-Wiki*
http://wiki.lazarus.freepascal.org/Cross_compiling_for_Win32_under_Linux zu Einsatz.

Nachfolgende Zeilen (Beginnend mit einem „\$“) werden nacheinander in einem Terminal-Fenster eingegeben, das in diesem Verzeichnis: „/home/**laz-user**/fpc-3.0.2“ (**laz-user** gegen Deinen Usernamen ersetzen) gestartet wird :

Das „\$“-Zeichen wird nicht mit eingeben.

```
$ sudo make clean all OS_TARGET=win64 CPU_TARGET=x86_64
```

und dann

```
$ sudo make crossinstall OS_TARGET=win64 CPU_TARGET=x86_64  
INSTALL_PREFIX=/usr
```

Jetzt kommt noch ein symbolischer Link:

```
$ sudo ln -s /usr/lib/fpc/3.0.2/ppcrossx64 /usr/bin/ppcrossx64
```

Wenn Ihr an dieser Stelle die Fehlermeldung
„ln: failed to create symbolik link '/usr/local/bin/ppcrossx64': File exists“
bekommt, könnt Ihr sie ignorieren.
Das Crosscompiling für Windows 64-Bit funktioniert trotzdem.

1. **Lazarus 1.9.0** unter **Linux Mint 18.2** installieren und fürs **Windows-Cross-Compiling** einrichten

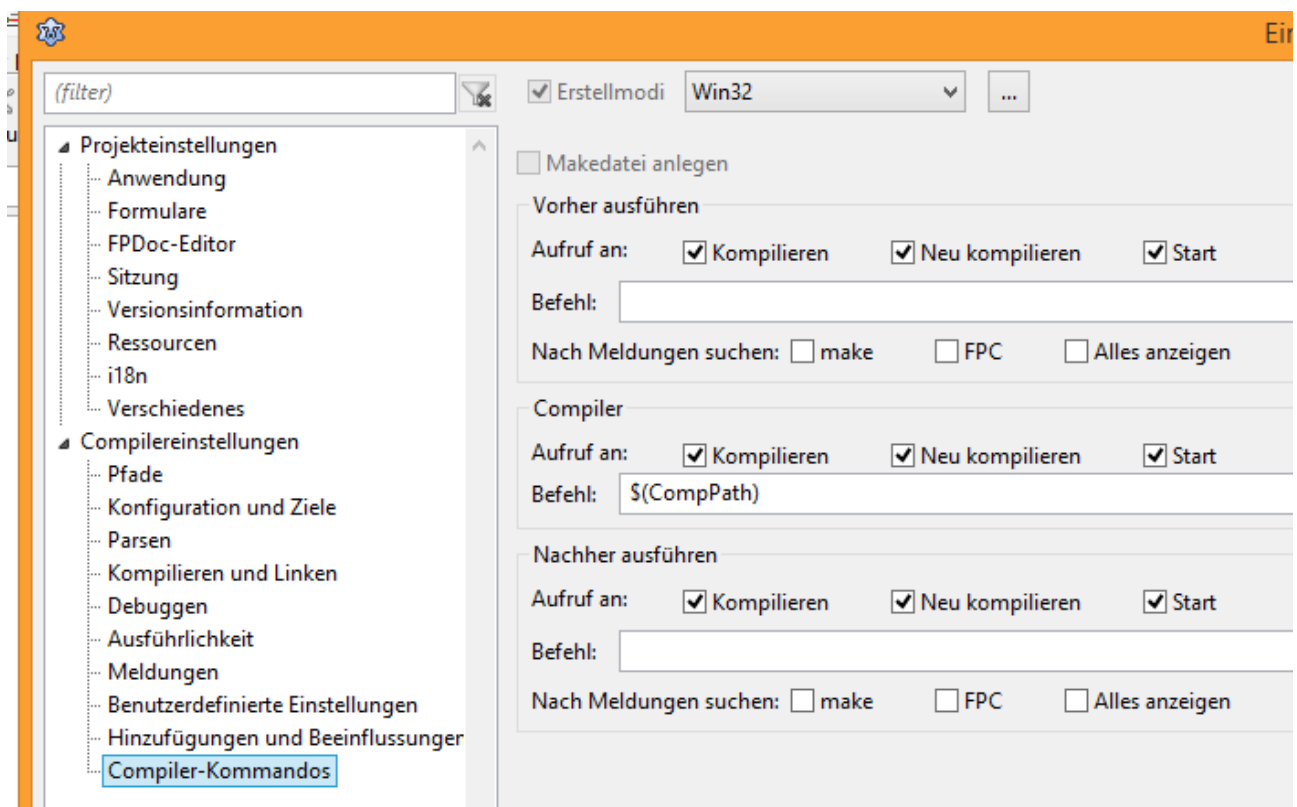
WINE in Lazarus einbinden

Um auch **Windows**-Programme durch F9 starten zu lassen (*besser ist es die Crosscompiling-Anwendungen mittels STRG+Umsch+F9 zu starten – da der Debugger mit **WINE** nicht funktioniert*), muss man **WINE** in **Lazarus** einbinden.

Als erstes muss man **WINE** durch die Anwendungsverwaltung von **Linux Mint** installieren lassen.

Wenn man das erledigt hat, muss man **Lazarus** noch mitteilen, dass es Win32/64-Programme an **WINE** weiterreichen soll.

Dieses geschieht unter „Projekt / Projekteinstellungen / Compiler-Kommandos“.

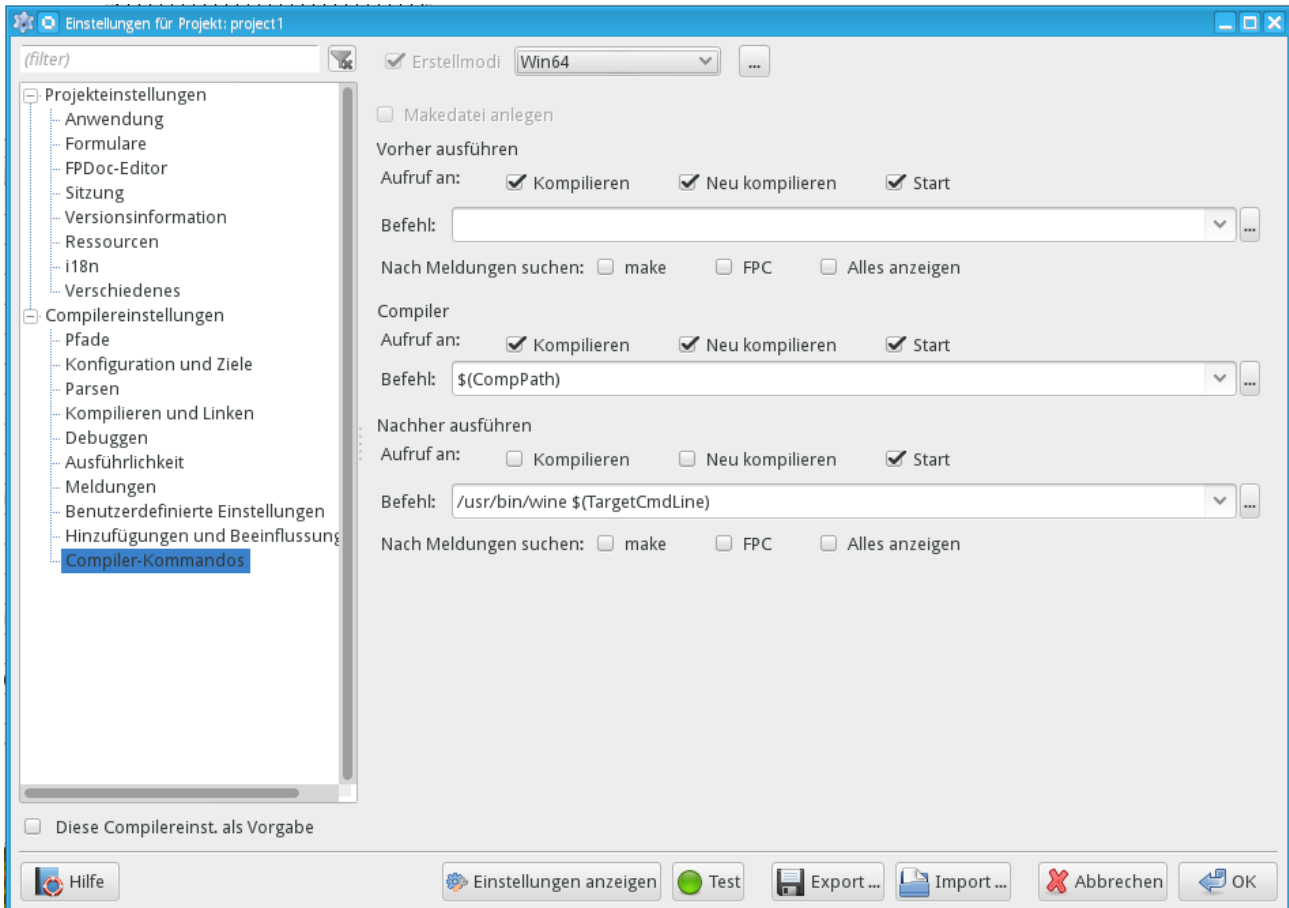


Wie schon zuvor kann man hier Einstellungen für jeden definierten Erstellmodus vornehmen.

1. Lazarus 1.9.0 unter *Linux Mind 18.2* installieren und fürs **Windows**-Cross-Compiling einrichten

Für Win32 und Win64 muss man unter „Nachher ausführen: / Befehl“ den Befehl:

„/usr/bin/wine \$(TargetCmdLine) eintragen und „[] Kompilieren“ und „[] Neu kompilieren“ deaktivieren.



Wenn Ihr die Befehlszeile per Copy&Paste einfügt, passt auf das am Anfang **kein Leerzeichen** steht – sonst funktioniert das alles nicht.

Wenn man dann das erste Mal eine **Windows**-Anwendung mittels STRG+Umsch+F9 startet, teilt **WINE** noch mit, das es „Wine-mono“ und zweimal „Wine-Gecko“ braucht und bietet auch an, diese zu installieren.

Jetzt sollten Win32-Anwendungen und Win64-Anwendungen automatisch in **WINE** ausgeführt werden.

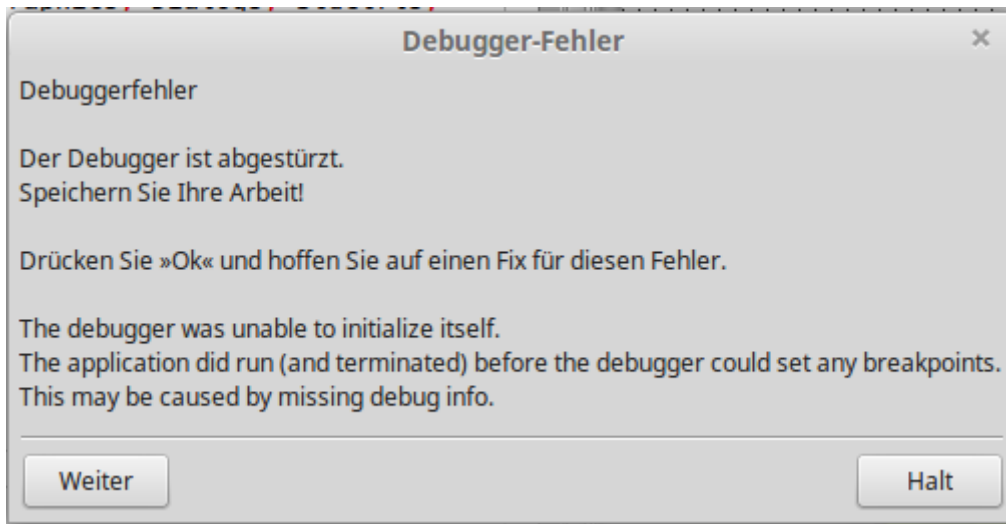
Das Ausführen in **WINE** erfolgt durch die Benutzung von STRG+Umsch+F9 ohne Debug.

Ein Debugging unter **WINE** geht nicht.

1. **Lazarus 1.9.0** unter *Linux Mind 18.2* installieren
und fürs **Windows**-Cross-Compiling einrichten

Debugger Fehlermeldung

Wenn man eine Win32 / Win64-Anwendung einfach mittels F9 im Debug-Modus startet, erhält man nach dem schliessen der Anwendung folgende Meldung:



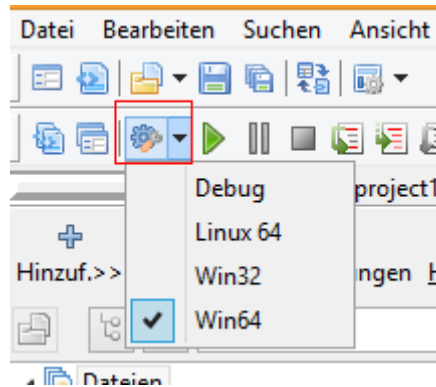
Einfach auf „Halt“ klicken und Lazarus läuft normal weiter.

Vielleicht hat ja jemand eine Lösung für dieses kleine Problem und teilt sie mir mit.

1. **Lazarus 1.9.0** unter *Linux Mint 18.2* installieren und fürs **Windows**-Cross-Compiling einrichten

Erstellmodus in der IDE ändern

Um den Erstellmodus in der IDE zu ändern, klickt Ihr auf den kleinen Pfeil neben das Zahnrad.



In einer Liste könnt Ihr dann den Modus auswählen.

Crosscompiling Linux 64 nach Linux 32

In der vorherigen Anleitung (unter Linux Mint 17.3) gab es noch die Möglichkeit in einem 64-Bit Linux eine 32-Bit Linux-Anwendung zu erstellen.

Das ist durch Änderungen seitens Linux Mint ab Version 18.1 nicht mehr möglich.

Die Anleitung im Lazarus-Wiki

http://wiki.lazarus.freepascal.org/Cross_compiling#From_Linux_x64_to_Linux_i386

funktioniert jedenfalls nicht mehr. Falls jemand eine Lösung hat, möge er mir diese bitte mitteilen.

Ich hoffe, Ihr habt jetzt viel Spaß und Erfolg mit Eurem neuen Lazarus

Gruß Heiko

info@rompelsoft.de

1. **Lazarus 1.9.0** unter *Linux Mint 18.2* installieren und fürs **Windows**-Cross-Compiling einrichten

Schlusswort

Ich hoffe, ich kann vielen mit dieser Anleitung weiterhelfen.

Copyright-Hinweise

- Lazarus
- Freepascal
- Windows
- Linux
- Linux-Mint
- Blaise Pascal Magazin
- WINE

Sind Markenbezeichnungen der verschiedenen Hersteller.

Durch die Nutzung dieser Bezeichnungen habe ich keinerlei finanzielle oder sonstige Vorteile.

Danksagung

Ich danke meiner Frau für das Korrekturlesen und meinem Sohn für die englische Übersetzung (der Vorgängerversion).

Ich danke auch Michael Van Canneyt für das Ursprungs-Script und die Erlaubnis es weiterzugeben und hier zu nutzen.

Ich danke auch den Mitgliedern der Lazarus- und Linux-Foren für die Beantwortung meiner Fragen.