

**OUtil**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> OUtil	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		March 16, 2022
<i>SIGNATURE</i>		

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>OUtil</b>	<b>1</b>
1.1	OUtil.guide	1
1.2	OUtil.guide/Copyright	2
1.3	OUtil.guide/Adresse des Autors	3
1.4	OUtil.guide/Einleitung	3
1.5	OUtil.guide/Systemanforderungen	5
1.6	OUtil.guide/Installation	5
1.7	OUtil.guide/Bedienung	5
1.8	OUtil.guide/Starten	6
1.9	OUtil.guide/Fenster	7
1.10	OUtil.guide/MainWindow	8
1.11	OUtil.guide/MainMenu	8
1.12	OUtil.guide/MainGadgets	10
1.13	OUtil.guide/MainSonst	12
1.14	OUtil.guide/CompilerSettings	13
1.15	OUtil.guide/CompilerMenu	13
1.16	OUtil.guide/CompilerGadgets	13
1.17	OUtil.guide/CompilerSonst	14
1.18	OUtil.guide/DebuggerSettings	15
1.19	OUtil.guide/DebuggerMenu	15
1.20	OUtil.guide/DebuggerGadgets	15
1.21	OUtil.guide/EditorSettings	16
1.22	OUtil.guide/EditorMenu	16
1.23	OUtil.guide/EditorGadgets	16
1.24	OUtil.guide/GeneralSettings	18
1.25	OUtil.guide/GeneralMenu	18
1.26	OUtil.guide/GeneralGadgets	18
1.27	OUtil.guide/Der ARexx Port	19
1.28	OUtil.guide/Geschichte	25
1.29	OUtil.guide/ToDo	29
1.30	OUtil.guide/Sonstiges	30
1.31	OUtil.guide/Index	30

# Chapter 1

## OUtil

### 1.1 OUtil.guide

OUtil Release 2.3 Dokumentation

\*\*\*\*\*

Rechtliches:

Copyright

Copyright und andere rechtliche Dinge

Adresse des Autors

Wohin man Bug reports, Kommentare & Spenden schickt

Das Oberon Utility (kurz OUtil):

Einleitung

Warum eigentlich OUtil?

Systemanforderungen

68040, 16 MB RAM, ... ;-)

Installation

Was kommt wohin?

Bedienung

Allgemeine Grundlagen zu Bedienung

Starten

Wie starte ich OUtil eigentlich?

Fenster

Die Fenster von OUtil und deren Bedeutung

Der ARexx Port

Na, der ARexx Port halt.

Anhänge:

---

Geschichte	Die Entwicklung von OUtil
Was noch?	Was es noch zu tun gibt.
Sonstiges	Der Autor hat noch etwas kund zu geben
Index	Index für dieses Dokument

## 1.2 OUtil.guide/Copyright

Copyright und andere rechtliche Dinge  
\*\*\*\*\*

Copyright (C) 1992,1993 Albert Weinert

Diese Dokumentation darf kopiert und weitergegeben werden solange die Copyright-Notiz und diese Erlaubnis unverändert auf allen Kopien enthalten ist.

Es wird keine Garantie gegeben, daß die Programme, die in dieser Dokumentation beschrieben werden, 100%ig zuverlässig sind. Sie benutzen diese Programme auf eigene Gefahr. Die Autoren können auf \*keinen\* Fall für irgendwelche Schäden verantwortlich gemacht werden, die durch die Anwendung dieser Programme entstehen.

Das Paket ist \*freely distributable\*, aber das Copyright liegt weiterhin bei Albert Weinert. Dies bedeutet, daß es von jedem kopiert werden darf solange er nicht mehr als eine angemessene Kopiergebühr dafür verlangt. Diese Gebühr \*darf nicht\* höher sein als US \$5 oder 5 DM.

\*Dieses Limit gilt auch für deutsche Public-Domain Händler!!\*

Dieses Paket darf in Public-Domain Sammlungen aufgenommen werden (CD ROM Versionen dieser Sammlungen eingeschlossen). Die Distributionsdatei darf in Mailboxsystemen oder auf FTP Servern abgelegt werden. Wenn Sie dieses Paket weitergeben wollen, dann \*müssen\* Sie die originale Distributionsdatei 'OUtil\_V23.lha' benutzen.

Die Programme und der Quelltext (oder Teile davon) dürfen \*auf keinen Fall\* auf irgendeiner Maschine benutzt werden, die für die Forschung, Entwicklung, Konstruktion, Tests oder Produktion von Waffen oder anderen militärischen Gütern benutzt wird. Dies gilt natürlich auch für alle Maschinen, die für das Training von Personen in \*irgendeiner\* der obengenannten Tätigkeiten benutzt werden.

### 1.3 OUtil.guide/Adresse des Autors

Adresse des Autors

\*\*\*\*\*

Der Autor kann unter folgenden Adressen erreicht werden:

Postadresse:

Albert Weinert  
Krähenweg 21  
50829 Köln (bis July 1993, W-5000 Köln 30)  
GERMANY

Electronic Mail:

Im Subnet (Internet):  
aweinert@darkness.gun.de  
oder im Z-Netz  
a.weinert@darkness.zer  
oder von Fido aus  
Albert Weinert of 2:241/4033

Die Fido--Adresse wurde noch nie getestet, wenn kein Reply kommt bitte andere Adresse benutzen.

### 1.4 OUtil.guide/Einleitung

Einleitung

\*\*\*\*\*

Das OUtil ist ein Programm das geschrieben worden ist, um den Umgang mit dem Amiga Oberon Compiler von der A+L AG zu erleichtern. So das dieses Programm auch nur für Programmierer interessant ist die diesen Compiler benutzen, falls sie es nicht tut dann empfehlen sie dieses Programm doch einer/m Freund/in von ihnen so das er in den Genuß dieses Programms kommt.

Die Entwicklungsumgebung von Amiga Oberon ist eigentlich in Ordnung, denn man kann den Quelltext Kompilieren, das Programm Linken und Starten, dies geht sogar alles vom Editor aus. Es ist sogar möglich die Compileroptionen von Editor aus einzustellen.

Aber wenn man längere Zeit damit arbeitet werden die Mängel des System sichtbar:

- \* Die Compileroptionen können nicht abgespeichert werden, so das sie jedesmal wenn der Editor aufgerufen wird eingestellt werden müssen. Es gibt zwar den Weg daß über einen Alias zu machen und dem Editor die Option beim Aufruf mitzugeben. Aber dieses ist dieses ist wieder ein Allgemeingültiger Aufruf, den man dann wieder an die verschiedenen Projekte, an denen man Arbeitet, anpassen müsste.
- \* es können vom Editor aus keine Objektdateien aus anderen Programmiersprachen zu seinem Programm hinzugelinkt werden. So das wenn man zum Beispiel einen Assemblerteil in sein Programm einbauen will,

jedesmal für das Linken in eine Shell wechseln muss.

- \* um ein Programm zulinke muss sich der Haupt Quelltext immer im Speicher befinden auch wenn er im Moment nicht gebraucht wird, weil er evtl. schon fertig ist.
- \* man kann ODebug nicht aus dem Editor heraus starten.
- \* man kann OMake nicht mit der Option ALL starten.
- \* man kann nicht mal schnell eine Kopie von seinen Quelltext oder Programm machen, so das man auf eine evtl. Lauffähige Version zurückgreifen kann wenn das Programm mal nicht mehr das macht was es machen sollte.
- \* MakeFiles sind für das AmigaOberon packet ein Fremdwort

Diese Mängel werden von OUtil zu gut es geht beseitigt und es gibt noch einige Verbesserungen um die Arbeit noch weiter zu vereinfachen.

- \* einstellen von Optionen für den Compiler, Editor und Debugger.
  - \* starten von Oberon, OLink, OMake, ODebug, ModToDef und des Programms selber.
  - \* eine Kopie vom Quelltext und des ausführbaren Programms erstellen.
  - \* starten des Programm mit einer Parameterzeile die übergeben werden kann.
  - \* verwalten von beliebig vielen Quelltexten
  - \* verwalten von beliebig vielen Linke Parametern
  - \* verwalten von beliebig vielen Parameterzeilen für den Programmstart
  - \* verwalten von beliebig vielen Compileroptionen für die bedingte Compilierung.
  - \* laden eines Quelltextes direkt in den Editor, vom OUtil aus.
  - \* alle Einstellungen werden in einer Prefs-Datei gespeichert so das man später die Einstellungen nicht nochmal vollziehen muss.
  - \* starten eines Editors beim Start von OUtil wobei die Quelltexte an denen man gerade arbeitet direkt mit eingeladen werden.
  - \* starten des Residentmanagers beim start von OUtil, wenn dieser noch nicht im Speicher sein sollte.
  - \* ARexx Unterstützung
  - \* Unterstützung von beliebigen Editoren mit ein ARexx Schnittstelle.
  - \* Möglichkeit der Makefile Ausführung
-

## 1.5 OUtil.guide/Systemanforderungen

Systemanforderungen

\*\*\*\*\*

Hardware

=====

An die Hardware werden so gut wie keine Anforderungen gestellt, OUtil sollte auf jedem Amiga laufen.

Software

=====

Bedingung zur Lauffähigkeit von OUtil ist mindestens AmigaOS 2.04, d.h. mindestens Kickstart 37.175 und Workbench 37.67.

Um die Funktionen von OUtil alle nutzen zu können muss das AmigaOberon-Paket 2.1 oder besser 3.0 vorhanden sein.

## 1.6 OUtil.guide/Installation

Installation

\*\*\*\*\*

Die Installation von OUtil ist sehr einfach, einfach das Programm 'OUtil' inkl. Piktogramm in das Verzeichnis 'Oberon:' und das Standard-Piktogramm 'OUtilPrefs.info' \*muss\* in das Verzeichnis 'Oberon:Icons' kopiert werden. Und schon ist OUtil installiert.

## 1.7 OUtil.guide/Bedienung

Bedienung

\*\*\*\*\*

Die Bedienung von OUtil ist auch sehr einfach, wenn man sich einmal daran gewöhnt hat.

Die Modulnamen beginnen immer mit "TXT/" und Enden mit ".mod" wenn dies nicht der Fall ist werden sie von OUtil vor der Benutzung des Namens (z.B: beim kompilieren, laden usw) erzeugt.

Die Stringgadgets die Quelltexte, Linkerparameter, Programmparameter und Compileroptionen beinhalten sind mit einem EditHook versehen, so das man wenn das StringGadget aktiv ist mit dem Cursortasten die Einträge in der Liste rauf und runter gehen kann um sich die Einträge auszuwählen. Auch ist Rechts neben diesen StringGadgets ein Rechteckiger Button, wenn man diesen anklickt erscheint eine Liste mit den Einträgen die dort vorhanden sind. Dort kann man sich mit der Maus einen Eintrag auswählen. Diese Liste kann man falls man den Button ausversehen Angeklickt hat durch nochmaliges



Anklicken wieder schliessen.

Die Gadgets haben in ihrem Namen einen unterstrichenen Buchstaben, wenn man diesen auf der Tastatur (ohne jeglichen Qualifier) drückt, hat dies dieselbe Funktion wie wenn man das Gadget mit der Maus anwählen würde.

In den Settings Fenstern gibt es immer das "Project" Menu dort sind zwei Menüpunkte.

Der erste Menüpunkt "Use" bedeutet das die aktuellen eingestellten Settings für den weiteren Programmbetrieb verwendet werden, und das das Settings Fenster geschlossen wird.

Der zweite Menüpunkt "Cancel" bedeutet das die aktuellen eingestellten Settings nicht für den weiteren Programmbetrieb übernommen werden, auch hier wird das Settings Fenster geschlossen.

Außerdem gibt es zwei Gadgets am unterem Windowrand ("Use" und "Cancel") die in der Funktion mit dem gleichnamigen Menüpunkten übereinstimmen.

Das Verzeichnis von dem aus OUtil gestartet worden ist muss ein "TXT" Verzeichnis haben, denn dort müssen bzw. werden die Quelltexte stehen.

Für den Aufbau des Verzeichnisses empfehle ich folgende Direktories.

```
TXT (Dir)
SYM (Dir)
OBJ (Dir)
REF (Dir)
DOC (Dir)
```

Desweiteren sollte OUtil wenn es von einer Shell gestartet wird mit 'run >NIL: OUtil' gestartet werden (oder halt mit RunBack).

Das File "OUtilPrefs.info" muss in das Verzeichnis "Oberon:Icons" gelegt werden. Sonst kann dem Konfigurationsfile kein Icon gegeben werden.

Bei CheckBoxGadgets wird immer die Funktion beschrieben, die bewirkt wird wenn das Gadget abgehakt ist.

## 1.8 OUtil.guide/Starten

Wie starte ich OUtil eigentlich  
\*\*\*\*\*

Shell Start  
=====

Der Parameterstring der OUtil übergeben wird hat folgende template:

```
TEMPLATE : Prefs/K,SCREEN=PUBSCREEN/K
```

Wenn man OUtil über die Shell startet, sieht der Aufruf ungefähr so aus.

---

```
run >NIL: OUtil PREFS=OUtil_V2.prefs PUBSCREEN="Oberon Screen"
```

Als PrefsFile kann man angeben welches PrefsFiles von OUtil geladen werden so. Wird dies nicht gefunden so wird versucht in dem aktuellen Verzeichnis die Datei "OUtil.prefs" zu laden (dieses wird auch geladen wenn man kein PrefsFile angibt). Wenn dieses PrefsFile dann auch nicht gefunden wird, versucht OUtil noch das PrefsFile "ENV:OUtil/OUtil.prefs" zu laden und wenn dies alles nicht vorhanden ist werden interne Voreinstellungen benutzt.

Als ScreenName kann man angeben auf welchen Public Screen sich OUtil öffnen soll. Hier ist voreingestellt "Oberon Screen" man kann zum Beispiel hier auch "CygnusEdScreen1" eintragen. Dann öffnet sich OUtil auf dem CygnusEdScreen. Voraussetzung ist natürlich das man den CygnusEdScreen als Public deklariert hat. Wenn der Screen sich immer auf der Workbench öffnen soll kann man "Workbench" angeben.

Workbench Start  
=====

Wenn das PrefsFile mit einem Icon versehen ist, so lässt sich OUtil auch von der Workbench starten. Einfach ein Doppelklick auf des PrefsFile und dann wird OUtil gestartet. (Die Voraussetzung ist natürlich das als DefaultTool OBERON:OUtil eingetragen ist). Dieses Icon kann denn ToolType PUBSCREEN = ScreenName enthalten so das auch hier den Screen angeben kann auf den sich OUtil öffnen soll.

Praktisch ist es wenn man das Icon auf die Workbench auslagert, denn so kann direkt nach hochfahren des System das Icon Anklicken und so sehr schnell an seinem Projekt arbeiten.

## 1.9 OUtil.guide/Fenster

Die Fenster von OUtil

\*\*\*\*\*

Beim OUtil gibt es verschiedene Fenster, in diesem Fenster können Parameter eingestellt werden oder Funktionen ausgeführt werden. Insgesamt gibt es folgende Fenster:

MainWindow

Die Zentrale

CompilerSettings

Alles für den Compiler

DebuggerSettings

Ja, da kann man auch was einstellen

EditorSettings

Rund um den Editor

GeneralSettings

---

Der allgemeine Kram

## 1.10 OUtil.guide/MainWindow

Das MainWindow Fenster

\*\*\*\*\*

Menüs

Die Menüs vom MainWindow Fenster

Gadgets

Die Gadgets vom MainWindow Fenster

Sonstiges

Was es sonst noch zu sagen gibt

## 1.11 OUtil.guide/MainMenu

Die Menüs des MainWindow Fensters

\*\*\*\*\*

Das Project Menü

=====

`Save Settings`

Speichert die Konfiguration unter dem bei Starten angegebenen Namen ab, wenn kein Namen angegeben worden ist wird als Name "OUtil.prefs" benutzt. Es sei denn ist die Konfiguration aus dem ENV: Verzeichnis gelesen worden oder die internen Parameter benutzt worden. Wenn dies der Fall ist, wird verfahren wie mit "Save Settings As..."

`Save Settings As...`

Speichert die Konfiguration ab aber vorher wird nach einen Namen dafür gefragt unter dem gespeichert werden soll, voreingestellt ist "OUtil.prefs" oder der Name den man beim Start von OUtil angegeben hat.

`Save As Default`

Speichert die Konfiguration in zwei Files ab, als "ENV:OUtil/OUtil.prefs" und als "ENVARC:OUtil/OUtil.prefs". Es wird die komplette Konfiguration gespeichert bis auf die Module, die OLinkParameter und die Kommandozeilen für das Programm.

`About...`

Gibt einen Requester aus, indem die Versionsnummer und der Copyright Vermerk von OUtil angegeben sind.

`'Exec ARexx...'`

Öffnet einen Filerequester mit dem Pfad "REXX:" und dem Pattern "#?.outil" so das man sich ein REXXSkript auswählen kann welches ausgeführt werden soll. Dies Skript wird Asynchron gestartet

`'Stop ARexx'`

Stop ALLE ARexx Skript die Momentan am laufen sind. Dies wird erreicht in dem der Shell Befehl "HI" gestartet wird.

`'Quit'`

Beendet das Programm und entfernt es aus dem Speicher. Es ist möglich das wenn man ARexx Kommandos sendet, das diese noch nicht wieder bestätigt worden sind. Falls das so ist erscheint ein Requester und man kann das Programm nicht beenden. In diesem Falle einfach den Requester betätigen, dann ein bisschen warten und dann nochmal quitten. Diesen Requester dürfte man eigentlich nie zu sehen bekommen. ARexx kommandos werden von OUtil gesendet wenn man von OUtil aus den Editor veranlasst einen Quelltext zu laden, und wenn man Skript mit "Exec ARexx..." gestartet hat. Man kann evtl. versuchen alle Skripts mit "Stop ARexx" zu unterbrechen so und danach nochmal zu quitten.

## Das Settings Menü

=====

`'Editor...'`

Öffnet ein Fenster auf dem man Parameter für den Editor einstellen kann.

`'Compiler...'`

Öffnet ein Fenster auf dem man die Compileroptionen einstellen kann.

`'Debugger...'`

Öffnet ein Fenster auf dem man Parameter für den Debugger einstellen kann.

`'General...'`

Öffnet ein Fenster auf dem man allgemeingültige Parameter einstellen kann.

Die einzelnen Fenster und deren Funktionen werden weiter unten noch beschrieben.

## Das Special Menü

=====

`'Backup Module...'`

Macht ein Backup von dem aktuellen Quelltext. d.h. OUtil kopiert den Quelltext und hängt ihm ein ".bak" an wobei die Endung ".mod" dann weggelassen wird.

`'Backup Executable...'`

Macht ein Backup vom dem aktuellen Programm, d.h.: OUtil kopiert das Programm und hängt dem Programm ein ".bak" an.

`'Module List'`

Hier gibt es zwei Untermenüpunkte für die Liste der Quelltexte die neben dem "Compile" Gadget in einem StringGadget angezeigt wird.

- \* Add an Entry

Fügt einen leeren Eintrag in der List hinter dem aktuellen Eintrag an.

- \* Remove Entry

Entfernt den aktuellen Eintrag aus der Liste, wenn der letzte Eintrag aus der Liste entfernt ist wird "Unused Entry" angezeigt.

#### 'Linker List'

Hier gibt es zwei Untermenüpunkte für die Liste der OLink Parameter die neben dem "Link" Gadget in einem StringGadget angezeigt wird.

- \* Add an Entry

Fügt einen leeren Eintrag in der List hinter dem aktuellen Eintrag an.

- \* Remove Entry

Entfernt den aktuellen Eintrag aus der Liste, wenn der letzte Eintrag aus der Liste entfernt ist wird "Unused Entry" angezeigt.

#### 'Parameter List'

Hier gibt es zwei Untermenüpunkte für die Liste der Programmparameter die neben dem "Start" Gadget in einem StringGadget angezeigt wird.

- \* Add an Entry

Fügt einen leeren Eintrag in der List hinter dem aktuellen Eintrag an.

- \* Remove Entry

Entfernt den aktuellen Eintrag aus der Liste, wenn der letzte Eintrag aus der Liste entfernt ist wird "Unused Entry" angezeigt.

#### 'Make All...'

Startet OMake mit der ALL Option so das alle Module die Importiert werden, neu kompiliert werden.

#### 'Load LinkFile'

Lädt die Link-Datei für das aktuelle Programm So, das man sie edieren kann.

#### 'Load MakeFile'

Lädt die Make-Datei für das aktuelle Programm So, das man sie edieren kann.

## 1.12 OUtil.guide/MainGadgets

---

Die Gadgets in dem MainWindow Fenster

\*\*\*\*\*

Die StringGadgets

=====

Das obere StringGadget beinhaltet die Module die mit OUtil bearbeitet werden können. Dabei ist der angezeigte Quelltext das aktuelle Module. Wenn mit dem Quelltext Funktionen ausgeführt werden wird falls nicht vorhanden, ein "TXT/" vorangestellt und ein ".mod" angehängt. Allerdings wird wenn ein "/" im Namen enthalten ist kein "TXT/" vorangesetzt und wenn ein "." oder "-" in Namen enthalten ist kein ".mod" angehängt. Dieses StringGadget kann mit CTRL-c aktiviert werden. Wenn man der Eintrag mit einem ';' beginnt, dann wird diese Modul nicht beim Programmstart geladen. Bei allen anderen Funktionen wirkt sich dieses nicht aus, es wird dann einfach ignoriert.

Das mittlere StringGadget beinhaltet die Parameter die OLink übergeben werden. Der angezeigte Eintrag ist der aktuelle Parameter. Der Eintrag muss genauso aussehen wie als wenn man OLink von der Shell aufrufen würde (wobei OLink nicht mit angebenem wurd darf), nur die Optionen brauchen nicht angegeben werden. Ich denke ein Beispiel schafft Klarheit.

Rechnung OBJ OBJ/Password.o

Hier würde das Programm "Rechnung" gelinkt wobei die Objektdatei "Password.o" da noch zusätzlich hinzugelinkt würde. Der OBJ Parameter ist optional es MUSS nur der Programmname drinstehen. Die Optionen für SmallCode oder SmallData werden automatisch OLink übergeben. Dieses StringGadget kann mit Ctrl-k aktiviert werden. Auch hier gibt es eine Spezial Funktion, wenn ein Eintrag mit einem '@' beginnt, dann wird beim Link und Make nicht direkt OLink bzw. OMake gestartet sondern ein LinkFile bzw. ein Makefile gestartet, der Name der Dateien ergibt sich aus dem den Programmnamen an denen ein ".lkf" für Linkfiles und ein ".mkf" für Makefiles anhängt wird. So ein Makefile \*muss\* ein "%s" enthalten, denn an dessen stelle wird der OLink oder OMake Aufruf gesetzt, ein Beispiel für so ein Makefile ist folgendes

```
ECHO Dies ist eine MakeDatei für das OUtil
FileDate txt/Arb2TeXInfo.cd txt/Arb2TeXInfoLocale.mod
IF NOT WARN
  Cd2Oberon txt/Arb2TeXInfo.cd txt/Arb2TeXInfoLocale.mod
ENDIF
%s
ECHO Dies ist das Ende
```

Hier würde getestet ob die die Datei 'TXT/Arb2TeXInfo.cd' neuer als die Date 'TXT/Arb2TeXInfoLocale.mod' wäre, und wenn ja dann würde mit 'Cd2Oberon' die Datei 'TXT/Arb2TeXInfoLocale.mod' auf den aktuellen Stand gebracht werden, anschließend wurde OLink oder OMake gestartet werden.

Wenn nur ein LinkFile vorhanden ist dann wird auch beim Make das LinkFile ausgeführt.

Das untere StringGadget beinhaltet die Parameter die dem Programm beim Start übergeben werden. Diese parameter werden komplett so an den Programmnamen gehangen wie sie eingegeben sind. Dieses StringGadget kann mit Ctrl-s aktiviert werden.

Die StringGadgets zeigen immer den aktuellen Eintrag aus einer Liste die wie in "Allgemeines zum Arbeiten mit OUtil" beschrieben ist gewechselt werden können.

Die Buttongadgets

=====

`Compile`

Startet den Compiler mit dem aktuellen Modul

`Link`

Startet den Linker mit den aktuellen Paramter für OLink

`Start`

Startet das Programm und übergibt um die aktuellen Paramter

`Make`

Startet OMake, wobei sich auf das bezieht was für OLink abgeben ist.

`Debug`

Startet den Debugger und anschliessend das Programm, wenn der Debugger schon im Speicher vorhanden ist wird nur das Programm gestartet.

`Load`

Läßt den Editor das Aktuelle Modul laden

`ModToDef`

Startet `ModToDef` um aus dem aktuellen Modul eine Definitionsdatei zu erstellen.

## 1.13 OUtil.guide/MainSonst

Was es sonst noch über das MainWindow Fenster zu sagen gibt

\*\*\*\*\*

Es ist möglich wenn das Window aktiv ist mit den Pfeiltasten, die Liste der Module und der OLinkparamter rauf und runter zu gehen.

Mit den Cursortasten rauf/runter wechselt man die Einträge in der Quelltextlist und mit den Cursortasten rechts/links wechselt man die Einträge om der OLinkparamterliste.

Mit der Tab-Taste ist es möglich das aktive Stringgadget zu wechseln und zwar mit "TAB" das nächste Stringgadget und mit "Shift-TAB" das vorherige Stringgadget.

Mit der Leertaste wird das vordereste Window was über die Mitte des Screens lugt aktiviert. Dies kann man dazu benutzen um das Quelltext Window von OEd zu aktivieren so das man wieder am Quelltext arbeiten kann.

---

## 1.14 OUtil.guide/CompilerSettings

Das CompilerSettings Fenster

\*\*\*\*\*

Menüs

Die Menüs vom CompilerSettings Fenster

Gadgets

Die Gadgets vom CompilerSettings Fenster

Sonstiges

Was es sonst noch zu sagen gibt

## 1.15 OUtil.guide/CompilerMenu

Die Menüs des CompilerSettings Fenster

\*\*\*\*\*

Das Projekt Menu

=====

Das Menü entspricht der Beschreibung die in "Allgemeines zum Arbeiten mit OUtil" gegeben wurde.

Das Special Menu

=====

`Compiler Options`

Hier gibt es zwei Untermenüpunkte für die Liste, der Compiler Options für die bedingte Kompilierung, die im gleichnamigen StringGadget angezeigt wird.

\* Add an Entry

Fügt einen leeren Eintrag in der List hinter dem aktuellen Eintrag an.

\* Remove Entry

Entfernt den aktuellen Eintrag aus der Liste, wenn der letzte Eintrag aus der Liste entfernt ist wird "Unused Entry" angezeigt.

## 1.16 OUtil.guide/CompilerGadgets

Die Gadgets des CompilerSettings Fenster

\*\*\*\*\*

---



## Die CheckBoxgadgets

=====

`StachkChk`  
 `OvflChk`  
 `RangeChk`  
 `ReturnChk`  
 `NilChk`  
 `TypeChk`  
 `OddChk`

Diese Gadgets sind für die Compiler-Optionen die den Überprüfungscode einstellen, schauen sie bitte für deren Bedeutung in ihrem Amiga-Oberon Handbuch nach.

`SmallCode`  
 `SmallData`  
 `ClearVars`  
 `Debug`  
 `AutoReqPars`  
 `Garbage Collector`  
 `Language Extensions`  
 `MergeHunks`  
 `68881`

Diese Gadgets beeinflussen die Code-Erzeugung vom Amiga Oberon Compiler, schauen sie bitte für deren Bedeutung in ihrem Amiga-Oberon Handbuch nach.

`AskSymbol`

Diese Option entspricht dem Menüpunkt "New Symbol" vom OEd, schauen sie bitte für ihre Bedeutung in ihrem Amiga-Oberon Handbuch nach.

## Die Cyclegadgets

=====

`CPU`

Dies ist das einzige CycleGadget, mit diesem kann man den Prozessor einstellen für den der Compiler das ausführbare Programm erzeugt.

## Die Stringgadgets

=====

Auch hier gibt es nur ein Gadget von, und zwar das Compiler Options Gadget hier stehen die Optionen für die bedingte Kompilierung diese werden dem Compiler übergeben.

Für dieses Stringgadget gilt auch das was im "Allgemeines zum Arbeiten mit OUtil" über Stringgadgets gesagt wurde.

## 1.17 OUtil.guide/CompilerSonst

Was es sonst noch über das CompilerSettings Fenster zu sagen gibt

\*\*\*\*\*

Mit dem Pfeiltasten rauf/runter kann man den aktuellen Eintrag im Compileroptions StringGadget ändern ohne das das Gadget aktiv sein muss.

## 1.18 OUtil.guide/DebuggerSettings

Das CompilerSettings Fenster

\*\*\*\*\*

Menüs

Die Menüs vom DebuggerSettings Fenster

Gadgets

Die Gadgets vom DebuggerSettings Fenster

## 1.19 OUtil.guide/DebuggerMenu

Die Menüs des DebuggerSettings Fenster

\*\*\*\*\*

Die Menus entsprechen der Beschreibung die in "Allgemeines zum Arbeiten mit OUtil" gegeben wurde.

## 1.20 OUtil.guide/DebuggerGadgets

Die Gadgets des DebuggerSettings Fenster

\*\*\*\*\*

Die CheckBoxgadgets

=====

`Screen`

Gibt an ob beim Start von ODebug ein eigener Screen geöffnet werden soll.

`Interlace`

Gibt an ob beim Start von ODebug der evtl. zu öffnende Screen sich im Interlace öffnen soll. Wobei dieses nur funktioniert wenn man keine ScreenMode innerhalb vom OEd eingestellt hat.

`Popup Windows`

`Sort Records`

`Close Pointer`

`Open Sources`

Die Optionen entsprechen den Menüpunkten von ODebug und deren

Bedeutung kann man im Handbuch nachschlagen.

Die Integergadgets

=====

`XPos`

`YPos`

`Width`

`Height`

Diese Gadgets geben die Position und Ausmaße der Fenster an die von ODebug geöffnet werden.

`Depth`

Die Anzahl der Bitplanes mit der ODebug seinen Screen öffnet.

## 1.21 OUtil.guide/EditorSettings

Das EditorSettings Fenster

\*\*\*\*\*

Menüs

Die Menüs vom EditorSettings Fenster

Gadgets

Die Gadgets vom EditorSettings Fenster

## 1.22 OUtil.guide/EditorMenu

Die Menüs des EditorSettings Fenster

\*\*\*\*\*

Die Menüs entsprechen der Beschreibung die in "Allgemeines zum Arbeiten mit OUtil" gegeben wurde.

## 1.23 OUtil.guide/EditorGadgets

Die Gadgets des EditorSettings Fenster

\*\*\*\*\*

Die CheckBoxgadgets

=====

`Screen`

Gibt an ob beim Start von OEd ein eigener Screen geöffnet werden soll.

`'Interlace'`

Gibt an ob beim Start von OEd der evtl. zu öffnende Screen sich im Interlace öffnen soll. Dieses Flag für von OEd ignoriert wenn man seinen ScreenModus vom OEd aus eingestellt hat.

`'AutoUpperCase'`

Schaltet im OEd die automatische Großschreibung von Standardbezeichnern ein.

Wenn im Stringgadget "Editor Name" nicht "OEd" vorkommt werden die Flags die mit den CheckBoxGadgets angezeigt werden nicht an den Editor weitergegeben.

## Die Integergadgets

=====

`'XPos'``'YPos'``'Width'``'Height'`

Diese Gadgets geben die Position und Ausmaße der Fenster an die vom OEd geöffnet werden.

`'Depth'`

Die Anzahl der Bitplanes mit der OEd seinen Screen öffnet.

`'MaxUndo'`

Die Maximale anzahl von Undo/Redo Schritten die OEd speichert und die er dann auch zurücknehmen kann.

Wenn im Stringgadget "Editor Name" nicht "OEd" vorkommt werden die Werte die in den Intergadgets stehen nicht an den Editor weitergegeben.

## Die Stringgadgets

=====

`'Editor Name'`

Hier steht der Aufruf des Editors wie er gestartet wird, als Voreinstellung ist `'Run >NIL: OBERON:OEd'` eingetragen. An diesen namen werden dann die evtl. zu ladenen Module angehängt. Man kann auch dem Editor auch Optionen übergeben. Wenn man z.B: den CygnusEd als seinen Lieblingseditor auserkoren hat kann man hier `'CED'` oder `'Ed -sticky'` eintragen.

`'Load Skript'`

Hier steht ein kleines ARexx Skript was seinen Editor veranlasst das aktuelle Modul zu laden. Dieses Skript muss in Anführungszeichen stehen. Hier ist für den OEd voreingestellt:

```
"Address 'OEd'; 'newwindow', 'load %s'"
```

An der stelle wo das %s im String steht wird vor dem Ausführen der Name des Moduls eingesetzt. Ein Beispiel Skript für den CED könnte so aussehen:

```
"Address 'rexx_ced'; 'open new'; 'open %s'"
```

## 1.24 OUtil.guide/GeneralSettings

Das General Settings Fenster

\*\*\*\*\*

Menüs

Die Menüs vom GeneralSettings Fenster

Gadgets

Die Gadgets vom GeneralSettings Fenster

## 1.25 OUtil.guide/GeneralMenu

Die Menüs des GeneralSettings Fenster

\*\*\*\*\*

Die Menüs entsprechen der Beschreibung die in "Allgemeines zum Arbeiten mit OUtil" gegeben wurde.

## 1.26 OUtil.guide/GeneralGadgets

Die Gadgets des GeneralSettings Fenster

\*\*\*\*\*

Die CheckBoxGadgets

=====

'Icons'

Gibt an ob OEd, Oberon, OLink und OUtil beim Speichern Ihrer Dateien Icons erstellen sollen.

'Start Editor'

Gibt an das der Editor den man in dem Editor Settings Window eingestellt hat beim Start von OUtil automatisch gestartet wird und das die Module die man nicht markiert hat, das diese dann geladen werden.

'Res. Manager'

Gibt an ob der Residentmanager gestartet werden soll, wenn dieser schon aktiviert im Speicher rumfliegt, wird er nicht nochmal gestartet.

'Oberon 3.00'

Diese Gadget sollte man aktivieren wenn man das Amiga Oberon Packet 3.0 besitzt, damit diese richtig aufgerufen wird. Wenn sie die Version 2.1

des Packetes besitzen, dann sollten sie hier den Haken nicht setzen.

Die Integergadgets

=====

'Wait Time'

Gibt die Zeit an wie lange OUtil warten soll bis es sein Window öffnet. Die ist dafür gedacht um damit zu warten bis der Editor geladen ist un der dann seinen Screen aufgemacht hat, damit OUtil sein Fenster auf dem Editor Screen öffnen kann (wenn es ein PublicScreen ist und man den Screenamen in dem Editor Settings Window eingestellt hat). Die Zeit wird in Ticks angeben (dies kommt daher weil die Dos.Delay() Funktion benutzt wird. Im Europäischen Raum ist es normalerweise so das 50 Ticks eine Sekunde sind.

Die Stringgadgets

=====

'PopKey'

Enthält den PopKey mitdem man das OUtil II MainWindow nach vorne holen kann. Dies kann von überall geschehen. Wenn die Tastenkombination gedrückt wurde, wird das Window aktiviert und der Screen worauf sich das Window befindet nach vorne geholt. Das Format dieses Stings entspricht der der commodities.library, wie die aktuellen Tastenbezeichnung sind kann man am besten aus dem Handbuch zur Workbench entnehmen.

## 1.27 OUtil.guide/Der ARexx Port

Der ARexx Port und was man damit macht

\*\*\*\*\*

Der ARexx Port umfasst zur Zeit 49 Befehle, und wird noch ausgebaut. Der Portname von OUtil ist "OUTIL", aber wenn jetzt OUtil mehrmals gestartet wird, so das OUtil mehrmals parallel läuft, so ändert sich auf der ARexx Portname. Und zwar wie folgt:

Das erste OUtil bekommt den Namen "OUTIL", das zweite "OUTIL.2" und das dritte "OUTIL.3" usw. Wenn jetzt allerdings z.b. OUtil Nummer 2 geschlossen wird. So bekommt dann das nächste OUtil was gestartet wird die nummer des geschlossen OUtil's im Beispiel also "OUTIL.2".

In der Beschreibung wird der Befehl angeben und das was dahintersteht ist die Template für die Parameter des Befehl,der Aufbau der Templates entspricht dem der Shell Programme der Workbench und kann im Shell Handbuch nachgelesen werden. Ein ", " als Template bedeutet das keine Argumente ausgewertet werden.

BEFEHL : MainWindow ,

FUNKTION: öffnet das MainWindow

ERGEBNIS: Nichts

BEFEHL : EditorSettings ,

FUNKTION: Öffnet das Editor Settings Window

ERGEBNIS: Nichts

BEFEHL : CompilerSettings ,

FUNKTION: Öffnet das Compiler Settings Window

ERGEBNIS: Nichts

BEFEHL : DebuggerSettings ,

FUNKTION: Öffnet das Debugger Settings Window

ERGEBNIS: Nichts

BEFEHL : GeneralSettings ,

FUNKTION: Öffnet das General Settings Window

ERGEBNIS: Nichts

BEFEHL : CloseMainWindow ,

FUNKTION: Schliesst das MainWindow, wobei OUtil noch im Speicherbleibt und so auch mit OpenMainWindow wieder geöffnet werden kann.

ERGEBNIS: Nichts

BEFEHL : CloseEditor ,

FUNKTION: Schliessen des Editor Settings Window

ERGEBNIS: Nichts

BEFEHL : CloseCompiler ,

FUNKTION: Schliessen des Compiler Settings Window

ERGEBNIS: Nichts

BEFEHL : CloseDebugger ,

FUNKTION: Schliessen des Debugger Settings Window

ERGEBNIS: Nichts

---

BEFEHL : CloseGeneral ,

FUNKTION: Schliessen des General Settings Window

ERGEBNIS: Nichts

BEFEHL : Quit ,

FUNKTION: Beenden von OUtil, schliesst alle Fenster und entfernt es aus dem Speicher.

BEFEHL : Compile MODULE

FUNKTION: Startet den Compiler, für weitere Beschreibung siehe "MainWindow/ButtonGadgets/Compile"

EINGABE : MODULE = Quelltext der Kompiliert werden soll, wenn kein Quelltext angegeben wird, wird der aktive Quelltext genommen der im StringGadget steht. Auch hier wird ein TXT/ bzw. ein .mod angefügt falls es nicht vorhanden ist.

ERGEBNIS: Nichts

BEFEHL : Link ,

FUNKTION: Startet den Linker, für weitere Beschreibung siehe "MainWindow/ButtonGadgets/Link"

ERGEBNIS: Nichts

BEFEHL : Start ,

FUNKTION: Startet das Programm, für weitere Beschreibung siehe "MainWindow/ButtonGadgets/Start"

ERGEBNIS: Nichts

BEFEHL : Debug ,

FUNKTION: Startet den Debugger, für weitere Beschreibung siehe "MainWindow/ButtonGadgets/Debug"

ERGEBNIS: Nichts

BEFEHL : Load MODULE

FUNKTION: Lädt den Quelltext, für weitere Beschreibung siehe "MainWindow/ButtonGadgets/Load"

EINGABE : MODULE = Quelltext der geladen werden soll, wenn

---



kein Quelltext angegeben wird, wird der aktive Quelltext genommen der im StringGadget steht. Auch hier wird ein TXT/ bzw. ein .mod angefügt falls es nicht vorhanden ist.

ERGEBNIS: Nichts

BEFEHL : ModToDef /N

FUNKTION: Erstellt eine Definitionsdatei, für weitere Beschreibung siehe "MainWindow/ButtonGadgets/ModToDef"

ERGEBNIS: Nichts

BEFEHL : Make ALL/S

FUNKTION: Startet das Make Utility, für weitere Beschreibungen siehe "MainWindow/ButtonGadgets/Make".

EINGABE : Startet das Make Utility mit der Option ALL, für weitere Beschreibungen siehe "MainWindow/Menu/MakeAll...".

ERGEBNIS: Nichts

BEFEHL : RefreshEditorWindow ,

FUNKTION: Das Editor Settingswindow wird Refreshed d.h. wenn man Optionen geändert hat und diese im Window angezeigt werden sollen falls es geöffnet war muss dieser Befehl aufgerufen werden.

ERGEBNIS: Nichts

BEFEHL : RefreshCompilerWindow ,

FUNKTION: Das Compiler Settingswindow wird Refreshed d.h. wenn man Optionen geändert hat und diese im Window angezeigt werden sollen falls es geöffnet war muss dieser Befehl aufgerufen werden.

ERGEBNIS: Nichts

BEFEHL : RefreshDebuggerWindow ,

FUNKTION: Das Debbuger Settingswindow wird Refreshed d.h. wenn man Optionen geändert hat und diese im Window angezeigt werden sollen falls es geöffnet war muss dieser Befehl aufgerufen werden.

ERGEBNIS: Nichts

---

BEFEHL : RefreshGeneralWindow ,

FUNKTION: Das General Settingswindow wird Refreshed d.h. wenn man Optionen geändert hat und diese im Window angezeigt werden sollen falls es geöffnet war muss dieser Befehl aufgerufen werden.

ERGEBNIS: Nichts

BEFEHLE: StackChk, OvflChk, RangeChk, CaseChk, ReturnChk, NilChk, TypeChk, OddChk, AutoRegPars, ClearVars, AskSymbol, SmallCode, SmallData, AutoUpperCase, PopupWindows, SortRecords, ClosePointers, OpenSources, Icons, StartEditor, ResManager

TEMPLATE: ON/S,OFF/S,T=TOGGLE/S

FUNKTION: Mit diesen Befehlen kann man die Optionen in OUtil setzen die Befehle entsprechen den Namen der Gadgets.

EINGABE : ON = Schaltet die Option ein  
OFF = Schaltet die Option aus  
TOGGLE = Wechselt die Option von aus nach ein bzw. von ein nach aus

ERGEBNIS: ON oder OFF jenachdem welchen Zustand die Option vor dem Befehl hatte.

BEFEHL : EditorScreen ON/S,OFF/S,T=TOGGLE/S

FUNKTION: Einstellen der Option Screen im Editor Settings Window

EINGABE : ON = Schaltet die Option ein  
OFF = Schaltet die Option aus  
TOGGLE = Wechselt die Option von aus nach ein bzw. von ein nach aus

ERGEBNIS: ON oder OFF jenachdem welchen Zustand die Option vor dem Befehl hatte.

BEFEHL : EditorInterlace ON/S,OFF/S,T=TOGGLE/S

FUNKTION: Einstellen der Option Interlace im Editor Settings Window

EINGABE : ON = Schaltet die Option ein  
OFF = Schaltet die Option aus  
TOGGLE = Wechselt die Option von aus nach ein bzw. von ein nach aus

---

ERGEBNIS: ON oder OFF jenachdem welchen Zustand die Option vor dem Befehl hatte.

BEFEHL : DebuggerScreen ON/S,OFF/S,T=TOGGLE/S

FUNKTION: Einstellen der Option Screen im Debugger Settings Window

EINGABE : ON = Schaltet die Option ein  
OFF = Schaltet die Option aus  
TOGGLE = Wechselt die Option von aus nach ein bzw. von ein nach aus

ERGEBNIS: ON oder OFF jenachdem welches Zustand die Option vor dem Befehl hatte.

BEFEHL : DebuggerInterlace ON/S,OFF/S,T=TOGGLE/S

FUNKTION: Einstellen der Option Interlace im Debugger Settings Window

EINGABE : ON = Schaltet die Option ein  
OFF = Schaltet die Option aus  
TOGGLE = Wechselt die Option von aus nach ein bzw. von ein nach aus

ERGEBNIS: ON oder OFF jenachdem welchen Zustand die Option vor dem Befehl hatte.

BEFEHL : DebugOption ON/S,OFF/S,T=TOGGLE/S

FUNKTION: Einstellen der Option Debug im Compiler Settings Window. Ich muss leider den Namen ändern weil "Debug" schon für den Start des Debuggers verwendet wurde.

EINGABE : ON = Schaltet die Option ein  
OFF = Schaltet die Option aus  
TOGGLE = Wechselt die Option von aus nach ein bzw. von ein nach aus

ERGEBNIS: ON oder OFF jenachdem welchen Zustand die Option vor dem Befehl hatte.

BEFEHL : Processor 68000/S,68010/S,68020/S,68030/S,68881/S

FUNKTION: Einstellen der Processor Version für die das Programm kompiliert werden soll.

EINGABE : 68000 = für MC 68000  
68010 = für MC 68010  
68020 = für MC 68020  
68030 = für MC 68030

---

68881 = für MC 68881

ERGEBNIS: Nichts

## 1.28 OUtil.guide/Geschichte

Geschichte

\*\*\*\*\*

Geschichte der OUtil Releases:

ProgVersion	Interne Version	Autor	Datum
-------------	-----------------	-------	-------

-----  
Ver: 2.2           IntRev: 37.306           [awn] 03-Aug-1992

Tja, mit der Historyführung fange ich jetzt erst an, weil mir das vorher zuviel Arbeit war und ich jetzt das ganze über ARexx fast automatisiert habe (nur die Änderungen muss ich noch selbst tippen)

Beim Processor Gadget im Compiler Settings Window ist es nun möglich auch mit gedrückter Shifttaste abwärts zu Cyclen.  
Das gesamte GadgetSystem wurde ein ganze Menge vereinfacht, dies ist für die Tastaturabfrage so das die Tasten jetzt automatisch rausgesucht werden.

-----  
Ver: 2.2           IntRev: 37.307           [awn] 03-Aug-1992

Das Cycle Gadget Flackert jetzt nicht mehr wenn man mit der Shift-Taste runtercycled.

-----  
Ver: 2.2           IntRev: 37.308           [awn] 03-Aug-1992

Den Programmcode ein bisschen überarbeitet so das das Programm um sage und schreibe 992 bytes kürzer ist ;-)

-----  
Ver: 2.2           IntRev: 37.309           [awn] 04-Aug-1992

Wenn jetzt ein Window gegen unerlaubtes betätigen der Gadget geschützt wird (beim About Requester, beim File Requester und beim Starten vom Compiler, linker etc.), wird jetzt auch der WaitPointer der WB gesetzt.

-----  
Ver: 2.2           IntRev: 37.310           [awn] 04-Aug-1992

Beim Aboutrequester wird jetzt auch der Name des ARexx Ports mit angegeben (mal wieder was für das UISG)

-----  
Ver: 2.2           IntRev: 37.311           [awn] 05-Aug-1992

Das verschliessen des Windows wurde geändert, es entspricht jetzt dem Beispiel aus dem RKM:Librares 3rd Edition

-----  
Ver: 2.2           IntRev: 37.312           [awn] 06-Aug-1992

MenuPunkt "Exec ARexx" in Project Menu mit aufgenommen, jetzt ist

---

es möglich auch ARexx Skripts von OUtil aus Starten zu lassen  
(wieder was fürs UISG).

---

Ver: 2.2      IntRev: 37.313      [awn] 12-Aug-1992

Das einsetzen des Namens beim TextLaden wird nun über SPrintf gemacht,

---

Ver: 2.2      IntRev: 37.314      [awn] 12-Aug-1992

Es wird jetzt auch ein Backup von Programm gemacht wenn man bei  
den OLink Paramtern noch Objekte mit angeben hat.  
Außerdem ist es jetzt möglich 230 Zeichen statt 50 in der Linker List  
und 200 statt 50 Zeichen in der ParamterList einzugeben.

---

Ver: 2.2      IntRev: 37.315      [awn] 14-Aug-1992

Das Programm der Anleitung angepasst :) Im Projekt Menu wurde jetzt noch  
der Menüpunkt StopRexx eingebaut. Wenn man diesen anwählt wird das  
Programm "HI" gestartet und alles ARexx Skripts die Laufen gestoppt.

Diese Version wurde in FRAS gepostet.

---

Ver: 2.2a      IntRev: 37.316      [awn] 16-Aug-1992

Wenn nun ein Window durch einen zu großen Font nicht mehr auf den Screen  
paßt wird nun auf Topaz 8 runtergeschaltet. Allerdings wirkt das nicht ←  
wenn  
der ScreenFont so groß ist das das Window trotz Topaz 8 größer wie der  
Screen ist. Aber das wird im normalen Einsatz hoffentlich nicht erreicht.

---

Ver: 2.2a      IntRev: 37.317      [awn] 20-Aug-1992

BUG: Fehler in dem ListView welches die Einträge von einem StringGadget  
angezeigt hat. Wenn MungWall im hintergrund lief, hing sich der Rechner  
bei Schliessen des Windows auf....jetzt nicht mehr. Nun ist das Programm  
Enforcer und Mungwall getestet.

---

Ver: 2.2a      IntRev: 37.318      [awn] 20-Aug-1992

BUG: Bei der Speicherung der SmallData und SmallCode Option im Speicher  
und somit auch im Config File wurde die beiden Optionen vertauscht..  
jetzt nicht mehr...  
Die Liste der Einträge der StringGadget öffnet sich jetzt so groß das  
6 Einträge drin Platzfinden.

---

Ver: 2.2b      IntRev: 37.319      [awn] 24-Oct-1992

Durch einen Plattencrash ist mir irgendwann ein Modul abhandengekommen  
so das ich dieses neuschreiben musste. Gleichzeitig habe ich das  
Listenhändling auf ein AmigaOberon 3.0 kompatibles Modul umgestellt.

---

Ver: 2.2b      IntRev: 37.320      [awn] 09-Nov-1992

OUtil stürzt beim Config laden nun nicht mehr ab (ist es durch das  
neue Listen Händling). Aber Irgendwie klappt es noch nicht so ganz mit  
der Auswahl aus dem ListView Gadget.

---

---

Ver: 2.2b    IntRev: 37.321    [awn] 15-Nov-1992  
Hehe, mein IntuiSupport Modul hatte gar keine Funktionen zum verwalten eines ListView Gadgets :). Ich hatte das vorher wohl anders gemacht. Das bearbeiteten der Liste klappt nun. Nur der CallBackHook in den StringGadgets wirft mir ein paar Enforcer hits raus (aber erst seit dem neuen Enforcer von Michael Sinz).

---

Ver: 2.2b    IntRev: 37.322    [awn] 15-Nov-1992  
Die EnforcerHits sind nun nicht mehr das, man sollte halt nicht den Inhalt eines Gadgets in einem Edit Hook mit Procedures ändern die nicht dafür geeignet sind.

---

Ver: 2.2b    IntRev: 37.323    [awn] 20-Nov-1992  
Allgemein Interne Änderungen im Gadgethandling, nun ist es besser für die Zukunft gerüstet.

---

Ver: 2.2b    IntRev: 37.324    [awn] 21-Nov-1992  
Durch diverse Aufräumarbeiten im GadgetHandling ist das Programm nun etwas kürzer geworden.

---

Ver: 2.2b    IntRev: 37.325    [awn] 21-Nov-1992  
Das Window wurde jetzt für den Gadget aufbau in Segmente eingeteilt so das ein einfacheres Positionieren der Gadgets möglich ist.

---

Ver: 2.2b    IntRev: 37.326    [awn] 21-Nov-1992  
Das Gadget für das aufklappen der Liste, neben den StringGadgets hat jetzt ein Image, ich nenne es das ListImage. Naja, bei Windows 3.x ist so ein Image dafür üblich und mir viel kein besseres ein.

---

Ver: 2.2c    IntRev: 37.327    [awn] 27-Nov-1992  
Kleinere Fixes. Eigentlich unbedeutend, aber ich brauche diesen Eintrag um auf 2.2c zu erhöhen :)

---

Ver: 2.3    IntRev: 37.328    [awn] 30-Nov-1992  
Neue Option, wenn man im LinkerList StringGadget vor dem Programmnamen ein "@" setzt (ohne Leerstelle), dann wird nach erfolgreichem Linken bzw. Maken ein Batchdatei gestartet. Die Batchdatei hat folgenden namen xxxx.mkf, xxxx steht für den Programmnamen.

---

Ver: 2.3    IntRev: 37.329    [awn] 30-Nov-1992  
Es werden jetzt verschiedene Abfragen getätigt, so das man erst einen schritt nach dem anderen machen kann.  
Compile -> Link -> MakeFile -> Start, oder  
Make -> MakeFile -> Start.  
Wenn bei bei einer vorherigen Stufe ein Fehler auftrat, dann wird ein Requester geöffnet der Aufschluss gibt in welcher Stufe der Fehler auftrat und ob man trotzallem die gewählte Funktion ausführen will.

---

---

Ver: 2.3      IntRev: 37.330      [awn] 09-Jan-1993  
An AmigaOberon 3.0 angepasst. Zwar wird noch nichts von Oberon-2  
benutzt. Aber die neuen CompilerOptionen werden unterstützt. So  
das man damit Arbeiten kann.

---

Ver: 2.3      IntRev: 37.331      [awn] 10-Jan-1993  
AmigaOberon 3.0 Switch eingebaut, dafür ist der OEd 2.13 Switch  
entfernt worden. So kann man OUtil auch wieder mit älteren  
Compilerversionen verwenden. Wenn jetzt im Modulnamen ein  
/ vorkommt. So wird jetzt nicht mehr TXT/ davor gesetzt.

---

Ver: 2.3      IntRev: 37.332      [awn] 22-Jan-1993  
An AmigaOS 3.0 angepasst, so das dort die Menus jetzt im Release 3  
Look sind.

---

Ver: 2.3      IntRev: 37.333      [awn] 29-Jan-1993  
Die Images für die Gadgets neben den Stringgadgets, wurden bei mehr  
wie 2 Bitplanes falsch erstellt, dies ist jetzt behoben.

---

Ver: 2.3      IntRev: 37.334      [awn] 05-Feb-1993  
Zwei Bugfixes: OUtil stürzte ab wenn man es über die Shell startete  
und man parameter übergab. Auch wurde beim Start über die Shell  
nur der Editor gestartet wenn man OUtil gerunt hat. Diese beiden  
Fehler sind behoben

---

Ver: 2.3:      IntRev: 37.335      [awn] 22-Feb-1993  
Es wird jetzt ab Kickstart 3.0 der Zeichensatz des Bildschirmes  
als Zeichensatz für die Gadgets benutzt.

---

Revision: 37.336 Version: 2.3      [awn] 28-Mar-1993

Es wurde eine Linker Option garnicht vom OUtil unterstützt. Jetzt  
kann man im Compiler Window noch angeben ob alle Hunks zu einem  
großen Hunk zusammenn gefügt. Dies wurd bei "SmallCode" zwar jetzt  
schon gemacht, aber nun kann man expliziet angeben. Das EditorWindow  
wurde mal wieder etwas anders layoutet, es braucht jetzt nicht  
mehr soviel Platz in der Breite.

---

Revision: 37.337 Version: 2.3      [awn] 28-Mar-1993

Wenn der Editor der "OEd" ist, dann wird nach Beendigung des  
Compilierungsvorgangs die Fehlerdatei über ARexx neu eingelesen.  
Allerdings sollte zur Erkennung des OEds, auch OEd im Editoraufruf  
vorkommen.

---

Revision: 37.338 Version: 2.3      [awn] 28-Mar-1993

Beim Start vom OUtil und wenn man das CompilerWindow mit "Use"  
verläßt, dann werden zwei neue Dateien angelegt. "Oberon.prefs"

---

und "OLink.prefs", diese Dateien sind ein bis jetzt eine nicht Dokumentierte Funktion von "Oberon" und "OLink", diese beiden lesen diese Datei nämlich ein und setzten Ihre Optionen aus dieser Datei, die die aktuellen Stand der Optionen hat die auch in OUtil eingestellt sind. Diese Datei wird in dem Verzeichniss angelegt woraus OUtil gestartet wurde. Aber Achtung, die Optionen in den Dateien haben Absoluten Vorrang über den Shell Parameter bzw. ToolTypes.

---

Revision: 37.339 Version: 2.3 [awn] 08-Apr-1993

Und V39 wird jetzt mit SetVBuf() die Pufferart verändert so, das das erstellen der Dateien Oberon.prefs und OLinks.prefs unter V39 schneller geschehen müßte.

---

Revision: 37.340 Version: 2.3 [awn] 08-Apr-1993

Wenn jetzt kein Modul in der Modul-Liste eingetragen ist, aber dafür in der Link-Liste etwas steht dann wird das Compiliert was in der Link Liste steht.

---

Revision: 37.341 Version: 2.3 [awn] 16-May-1993

Komplett neue MakeFile Variante, das MakeFile wird jetzt nicht mehr nach erfolgreichem Linken oder Maken ausgeführt sondern, anstatt. In dem MakeFile sollte ein %s vorkommen, und nur eins, dieses wird dann durch den entsprechenden Linker oder Make aufruf ersetzt.

---

Revision: 37.342 Version: 2.3 [awn] 16-May-1993

Wenn in dem Module Namen ein "." oder ein "-" vorkommt, dann wird kein ".mod" mehr an den Namen gehangen.

---

Revision: 37.343 Version: 2.3 [awn] 17-May-1993

Man kann jetzt zwei verschiedene Dateien für das Linken und das Maken erstellen. Die für das Linken endet mit ".lkf" und die für das maken mit ".mkf". Wenn beim Make die #?.mkf Datei nicht da ist, dann wird die #?.lkf Datei genommen.

Im Special Menü sind jetzt die Menu Punkte 'Load LinkFile' und 'Load MakeFile' hinzugekommen. Wenn man diese anwählt, dann wird das MakeFile zum gerade aktuellen Programm in den Editor geladen.

## 1.29 OUtil.guide/ToDo

Was noch?

\*\*\*\*\*

---



Tja, hier steht jetzt also drin was ich noch vorhabe, also folgendes stelle ich mir vor ist noch ganz nützlich oder praktisch wenn ich es einbauen würde.

1. Die Compileroptionen von OUtil an OEd über ARexx übermitteln.
2. Den ARexx Befehlssatz ausbauen, bzw. komplett neu erstellen, dies wird dann wohl mit der ARexxBox geschehen.
3. Für Wünsche an dem Programm bin ich sehr offen eingestellt.

### 1.30 OUtil.guide/Sonstiges

Sonstiges  
\*\*\*\*\*

Ein Modul war nicht von mir und dieses war 'HotKey.mod' vom Thomas Igracki

### 1.31 OUtil.guide/Index

Adresse	Adresse des Autors
Bug reports	Adresse des Autors
Compiler Settings Fenster CompilerSettings	
Copyright	Copyright
DebuggerSettings Fenster DebuggerSettings	
Distribution	Copyright
E-Mail	Adresse des Autors
EditorSettings Fenster EditorSettings	
EditorSettings Fenster GeneralSettings	
Fähigkeiten in Stichpunkten	

---

Einleitung	
Geschichte	Geschichte
History	Geschichte
InterNet Adresse	Adresse des Autors
Kritik	Adresse des Autors
MainWindow Fenster	MainWindow
MainWindow Menüs	MainMenu
MainWindow Project Menü	MainMenu
MainWindow Settings Menü	MainMenu
MainWindow Special Menü	MainMenu
OUtil	Fenster
OUtil starten	Starten
Postadresse	Adresse des Autors
Rechtliche Dinge	Copyright
Shell Start	Starten
Sonstiges	Sonstiges
Spenden	Adresse des Autors
Was noch?	ToDo
Workbench Start	Starten

---