

WComm

COLLABORATORS

	<i>TITLE :</i> WComm	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		November 23, 2024
<i>SIGNATURE</i>		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	WComm	1
1.1	main	1
1.2	was ist wcomm	1
1.3	system voraussetzungen	4
1.4	rechtliches	4
1.5	installation und start	5
1.6	cli-start	6
1.7	wbench-start	9
1.8	hotkeys	10
1.9	liste der funktionen	12
1.10	allgemein	19
1.11	os 2.04 gadgets	19
1.12	das haupt fenster	21
1.13	das ausschluß fenster	23
1.14	das voreinsteller fenster	24
1.15	das uhren fenster	26
1.16	das nachrichten fenster	28
1.17	das paletten fenster	32
1.18	die uhr	33
1.19	fehlerbehandlung	34
1.20	technisches	35
1.21	liste der dateien	35
1.22	adresse	36

Chapter 1

WComm

1.1 main

Dokumentation

WComm Version 1.80

WComm das Fenster-Manipulations-Tool für den Amiga

Index:

Was ist WComm	Allgemeine Beschreibung Features
System Voraussetzungen	Welches OS usw.?
Rechtliches	Bitte Lesen !
Installation und Start	Wichtig
CLI-Start	Syntax und Parameter
WBench-Start	ToolTypes
HotKeys	Mögliche Kombinationen usw.
Liste der Funktionen	Alle 50 (!) ausführlich
Allgemein	Generelles
OS 2.04 Gadgets	Kurzbeschreibung der Benutzung
Das Haupt Fenster	
Das Ausschluß Fenster	
Das Voreinsteller Fenster	
Das Uhren Fenster	
Das Nachrichten Fenster	
Das Paletten Fenster	
Die Uhr	
Fehlerbehandlung	Wichtig
Technisches	Nicht nur für Programmierer
Liste der Dateien	Bitte prüfen
Adresse	Für alle Geschenke etc.

1.2 was ist wcomm

Was ist WComm:

WComm ist ein Standard-Kommodity, daß es dem Benutzer ermöglicht, 43 Fenstermanipulationsfunktionen mittels Hotkey (Tastatur-Macro) vorzunehmen.

Fenster lassen sich

- verschieben,
- vergrößern,
- verkleinern,
- auf bestimmte Voreinstellungen setzen,
- zurücksetzen,
- schließen,
- zoomen,
- teilen,
- nach vorne oder hinten bringen,
- kaskadieren,

Bildschirme und Fenster lassen sich durchblättern.

WComm besitzt außerdem die Möglichkeit den Benutzer auf bestimmte Ereignisse oder Termine aufmerksam zu machen. Dazu kann der Benutzer Daten im Nachrichtenfenster eingeben, wenn das eingestellte Datum und/oder die eingestellte Uhrzeit erreicht ist, öffnet WComm einen Reqlüster mit der entsprechenden Nachricht und/oder startet ein Programm (EXECUTE, RUN oder AREXX).

WComm besitzt auch eine komfortable Uhr mit verschiedenen Datums- und Uhrenanzeigetypen. Sie besitzt eine Alarmfunktion und ist fast vollkommen benutzerdefinierbar (Breite, Position, Anzeigart).

WComm kann Fenster von bestimmten Funktionen ausschließen oder nur bestimmte Fenster zulassen.

WComm kann Voreinstellungswerte für jedes Fenster verwalten und mit einem Tastaturkommando auf diese Werte (Position, Größe) setzen.

WComm hat ein Palettenfenster, mit dem die WBenchfarben auf einfache Art geändert werden können.

Jedes Tastaturkommando (Hotkey) kann nach den Wünschen des Benutzers abgeändert werden.

WComm gibt es in einer englischen (WComm) und einer deutschen (WComm_G) Version.

WComm kann mittels Hotkey eine Shell öffnen.

WComm kann "Fensterleichen" schließen (vom System vergeßene Fenster).

WComm ist ein Kommodity und voll durch Exchange kontrollierbar.

WComm ist absolut Zeichensatzunabhängig. Es paßt alle Fenster und Gadgets dem gerade aktuellen Zeichensatz an. Auch die Kaskadierungs- und Verschiebefunktionen benutzen die Höhe der Fenster Titelleisten als Vorgabewert.

Man sieht WComm kann eine Menge. Da erstaunt auch die Größe von ~80 kb nicht. Aber man kann durch die Benutzung von WComm eine Reihe anderer Programme ersetzen, so z.B.

IntuiHelp etc,
Uhrenprogramme,
Kronographenprogramme,
Palettenprogramme,
Tiler- Kaskadierungsprogramme,
Fensterkiller etc..

WComm hat eine komplette Gadtoolsbenutzeroberfläche und ermöglicht so alle Funktionen auf den entsprechenden Fenstern mittels Maus- oder Tastaturbedienung.

Alle Fenster und Funktionen sind asynchron, d.h. die Veränderungen wirken sich sofort, auch bei geöffneten Fenstern, aus.

WComm wird durch verschiedene Konfigurationsdateien gesteuert. Für jeden Teilbereich existiert eine andere Konfigurationsdatei. So wird für die Ausschlußliste, für die Voreinstellungsliste und für die Nachrichtenliste jeweils eine andere Datei verwendet.

Alle Dateien besitzen zwei voreingestellte Namen, so daß der Benutzer nicht das Problem mit nicht zu findenden Datei bekommt.

Außerdem besitzt WComm noch eine Konfigurationsdatei, in der die Tastaurmakros (Hotkeys), die Fensterpositionen, die Uhrenvoreinstellungen usw. abgespeichert sind.

WComm kann aber auch ohne jede Konfigurationsdatei gestartet und betrieben werden. Hierbei wird WComm den Benutzer auf die fehlenden Dateien aufmerksam machen.

Alle Konfigurationsdateien können auf den entsprechenden Fenstern abgeändert, editiert, neu geladen und abgespeichert werden.

Um auch hier den Komfort so hoch wie möglich zu halten, können die meisten Konfigurationsdateien auch mit einem Texteditor editiert werden. Siehe nähere Informationen bei den entsprechenden Absätzen.

Alle HotKeys können einzeln und unabhängig voneinander abgeändert und danach wieder abgespeichert werden, indem man das Hauptfenster benutzt. Der voreingestellte HotKey für das Öffnen des Hauptfensters ist `'Control o'`. Dieser HotKey kann natürlich auch später abgeändert werden.

Weiteres:

WComm unterstützt schon einige Eigenschaften von OS 3.0. Dabei kann es aber selbst unterscheiden, ob es auf einem System mit OS 2.xx oder 3.0 läuft.

Das Palettenfenster z.B. kann bis zu 256 Farben verwalten und anzeigen.

Das Erscheinungsbild der Fenster unterscheidet sich unter OS 3.0 und 2.xx in einigen Punkten:

Das PalettenGadget hat keinen Indikator der aktuellen Farbe, die aktuelle Farbe wird hier invertiert dargestellt.

Die ListViewGadgets zeigen kein TextGadget, das den aktuellen Eintrag dar-

stellt, an, sondern der aktuelle Eintrag wird invertiert dargestellt.

1.3 system voraussetzungen

System Voraussetzungen:

Da WComm ein Standardkommodity ist, benötigt es mindestens OS 2.04. Es wurde unter 2.04 und 3.0 getestet und läuft einwandfrei unter beiden Betriebssystemversionen.

Der Speicherverbrauch ist sehr abhängig von der Größe der Listen, die WComm für die verschiedenen Funktionen benötigt. Speichermangel sollte bei Amigas mit ≥ 1 MByte Memory nicht auftreten. Um dennoch Speicher zu sparen, kann man Elemente, wie z.B. Nachrichten melden oder Ausschlußlistenfunktionen inaktivieren. Auch können Hotkeys, die inaktiviert werden, einige Bytes an Speicher einsparen, da für jedes aktivierte Hotkey eine zusätzliche Struktur vom System angefordert und verwaltet werden muß.

WComm verwaltet seinen Speicher dynamisch, d.h. Speicher wird nur dann angefordert, wenn er benötigt wird. Er wird sofort zurückgegeben, wenn der Speicher nicht mehr benötigt wird.

1.4 rechtliches

Rechtliches:

Ich habe mich entschlossen dieses Programm und alle beigefügten Dateien und Scripts, sowie aller Anleitungen, als FREEWARE zu deklarieren.

D.h.:

Jeder kann WComm benutzen.

Jeder darf WComm weitergeben, wenn:

1. Alle unten aufgeführten Programme mit weitergegeben werden,
2. Keinerlei Veränderungen der unten aufgeführten Programmen (am Ende der Anleitung) vorgenommen wurden,
3. Bei Nachkompilierungen der QuellCodes muß ein Hinweis auf den Originalprogrammierer beigefügt sein, bei Herausgabe dieser, muß eine Kopie an mich gesandt werden.
4. WComm darf nicht verkauft werden.

Außnahme: Es darf eine geringe Kopiergebühr erhoben werden, die aber 5 DM bzw. den entsprechenden Gegenwert in anderen

Währungen nicht übersteigt.

5. WComm darf nur als Teil eines kommerziellen Produktes herausgegeben werden, mit ausdrücklicher schriftlicher Genehmigung des Autors (Adresse siehe unten).

Zu widerhandlungen der oben aufgeführten Konditionen werden zivilrechtlich verfolgt.

Ferner wird für die Funktion des Programmes, sowie möglicher Folgen der Benutzung des Programmes sowie der u. a. Dateien, keinerlei Gewährleistung übernommen.

!!! Alle Risiken trägt der Benutzer allein !!!

1.5 installation und start

Installation und Start:

Installation:

Die Installation geht sehr einfach von statten. Zunächst einmal müssen alle Konfigurationsdateien, die von WComm benötigt werden, in das ENVARC: - Verzeichnis kopiert werden.

```
1_> Copy #?.prefs ENVARC:
```

Danach müssen sich im ENVARC: folgende Konfigurationsdateien befinden:

```
XWindow.prefs,  
ZWindow.prefs  
WComm.prefs  
Notify.prefs
```

Will man die entsprechenden Icons ebenfalls kopieren:

```
1_> Copy #?.prefs.info ENVARC:
```

Nun stellt sich die Frage, ob man WComm ständig oder nur gelegentlich benutzen möchte. Auch ist es ein Unterschied, ob man WComm von der WBench oder vom CLI aus starten möchte.

Will man WComm ständig benutzen, kann man einfach das Icon von WComm (WComm_G) in das WBStartup-Verzeichnis der WBench ziehen. (Siehe auch Kapitel über WBench-Start).

Alternativ kann auch eine Zeile in die User-Startup-Datei einfügen:

```
Run <Pfad>WComm_G >NIL: <NIL:
```

WComm unterstützt auch eine Kommandozeilenargumente, die man beim Start mitangeben kann. (Siehe Kapitel CLI-Start).

Will man WComm nur gelegentlich benutzen, kann man es einfach mit dem Icon in das Commodities -Verzeichnis ziehen oder mit

```
1._> Copy WComm_G C:
```

in das C:-Verzeichnis kopieren.

Start:

Da WComm ein Standardkommodity ist, wird es beim zweiten Aufruf automatisch beendet.

Ist WComm gestartet worden bleibt es im Hintergrund und wartet auf ankommende Nachrichten. In dieser Zeit wird keinerlei CPU-Zeit verbraucht, außer der Zeit für das Setzen der WBenchtiteluhr.

WComm kann auch mit dem Exchange-Kommodity kontrolliert werden. Will man das Hauptfenster von WComm öffnen und hat den Hotkey dafür vergessen, kann man dies auch von Exchange aus. Ebenfalls läßt sich WComm von Exchange aus Aktivieren, Inaktivieren und Beenden.

1.6 cli-start

CLI-Start:

Wird WComm vom CLI aus gestartet werden folgende Kommandozeilenargumente mit folgender Schablone (Syntax) unterstützt:

```
WComm [Config] [CX_POPUP=P/S] [WPrefs=W/K] [EPrefs=E/K] [CX_Priority=C/K/N]
      [Notify=N/K]
```

Config

Wird Config gefolgt mit dem Namen einer Konfigurationsdatei angegeben, versucht WComm diese Datei zu laden. Wird Config nicht angegeben, wird versucht, die voreingestellte Konfigurationsdatei zu laden. Diese befindet sich im ENV: Verzeichnis und heißt WComm.prefs. Ist sie dort nicht zu finden versucht WComm diese Datei aus dem ENVARC:-Verzeichnis zu lesen. Kann auch sie nicht geladen werden, zeigt WComm eine entsprechende Fehlernachricht an, und ermöglicht dem Benutzer das Programm abzubrechen oder mit den voreingestellten Werten fortzusetzen.

Diese Konfigurationsdatei besteht aus binären Daten und kann deshalb nicht mit einem Texteditor abgeändert werden. Bitte nicht versuchen, sonst riskiert man einen Systemabsturz !

CX_POPUP

Dies ist ein Schalter. Wird CX_POPUP angegeben, öffnet WComm das Hauptfenster beim Start.

Voreinstellung: Hauptfenster nicht öffnen.

WPrefs

Wird WPrefs gefolgt von dem Namen einer Fensterlistendatei angegeben, so versucht WComm dies zu laden. Wird WPrefs nicht angegeben so sucht WComm diese Datei im ENV: - Verzeichnis mit dem Namen 'ZWindow.prefs'. Kann WComm sie nicht finden, sucht er im ENVARC:-Verzeichnis.

Die Fensterlistendatei enthält Voreinstellungen von Fenstern, mit deren Hilfe WComm diese Fenster auf diese Werte setzen kann. Siehe auch Kapitel Voreinsteller Fenster.

Diese Datei besteht aus reinen ASCII-Daten und kann so auch mit jedem Texteditor editiert werden. Dabei muß man das korrekte Format der Datei beachten.

Die Daten müssen folgender Syntax entsprechen:

Jede Zeile entspricht einer Fenstervoreinstellung,

der Titel darf keine Leerzeichen beinhalten,

AmigaDos Suchmuster sind im Titel erlaubt und werden voll unterstützt,

Zwischen den Angaben muß ein Leerzeichen stehen.

Kommentare sind nicht erlaubt.

Syntax: <Fenstername> <X> <Y> <W> <H>

Fenstername: Der Name mit oder ohne AmigaDOS Suchmustern.

Die Benutzung von Suchmustern ermöglicht eine Erkennung von Fenstertiteln, die sich mit der Zeit verändern. So z.B. verändert sich die Titelzeile eines WBenchfensters, wenn man etwas hinainkopiert oder löscht.

Beispiel:

```
RAM#?
      erkennt

      Ram Disk 100% voll 0 % frei ...
RAMDISK ...
RaMDiSk ...
```

Es wird also auch keine Unterscheidung zwischen Groß- und Kleinschreibung vorgenommen.

X: Die linke Position des Fensters.
Y: Die obere Position des Fensters.
W: Die Breite des Fensters.
H: Die Höhe des Fensters.

Die in dieser Datei stehenden Werte können mit der Funktion

'Setze nach Preferences' (siehe Liste der Funktionen) abgerufen werden. Entspricht das aktuelle Fenster dann dem Titelmuster, wird versucht das Fenster auf die angegebene Position zu verschieben und die Größe auf die angegebenen Werte zu ändern.

Verschoben werden allerdings nur Fenster mit einer DragBar, und Vergrößert bzw. Verkleinert werden nur Fenster mit einem SizeGadget.

Mit dieser Datei kann man also für jedes Fenster eine andere Voreinstellung abspeichern.

Es ist hiermit auch möglich eine Übergruppe zu definieren. Wenn man einige Fenstertitel mit den gleichen ersten Buchstaben hat, und jedes eine andere Voreinstellung benutzen soll, z.B.:

man hat 'Shell1', 'Shell2' und 'Shell', kann man einen Eintrag in die Datei mit Shell1 Shell2 und Shell#? eingeben. Öffnet man ein Fenster, daß z.B. Shell4 heißt, benutzt es die Voreinstellung von Shell#?, während die Shell1 die Werte für Shell1 benutzt.

EPrefs

Wird EPrefs gefolgt von dem Namen einer Ausschlußdatei angegeben, versucht WComm dies Ausschlußdatei zu laden. Schlägt dies fehl, versucht WComm die Datei 'XWindow.prefs' im ENV:-Verzeichnis zu laden. Kann auch sie nicht gefunden werden, sucht WComm sie im ENVCAR:-Verzeichnis.

Die Fenster, die in dieser Datei angegeben sind, werden von einigen Funktionen ausgeschlossen. So kann z.B. ein Uhrenfenster von der Kaskadierung, dem Vor- und Zurückblättern usw. ausgeschlossen werden.

Auch diese Datei besteht aus reinen ASCII-Daten und kann mit einem Editor abgeändert werden. Natürlich kann sie auch von dem Ausschlußfenster aus abgespeichert werden.

Dabei muß folgendes Format benutzt werden:
(Siehe Kapitel 'Liste der Funktionen' und 'Ausschluß Fenster')

Syntax: <Fenster Titel>

In jeder Zeile steht ein Fenstertitel. Er darf keine Leerzeichen enthalten, aber AmigaDOS Suchmuster.

CX_Priority

Die hinter CX_Priority angegebene Zahl, entspricht der Kommodity- ↔
Priorität,
die WComm benutzen soll. Voreingestellt ist 0. Läuft ein anderes ↔
Kommodity

gleichzeitig, einen gleichen Hotkey benutzt, bekommt nur das Kommodity ↔
mit
der höherenn Priorität, die Nachricht. Das mit der geringeren ↔
Priorität
bleibt weiter im Hintergrund.

Man sollte keine Prioritäten über 5 oder unter -5 benutzen.

Notify

Wird Notify gefolgt von dem Namen einer Nachrichtendatei angegeben, versucht WComm dies nachrichtendatei zu laden. Schlägt dies fehl, versucht WComm die Datei 'Notify.prefs' im ENV:-Verzeichnis zu laden. Kann auch sie nicht gefunden werden, sucht WComm sie im ENVARC:-Verzeichnis.

Diese Nachrichtendatei enthält alle Daten der Nachrichten, auf die WComm den Benutzer aufmerksam machen soll.

Diese Datei besteht aus reinen ASCII-Daten und kann auch von einem ↔
Editor
aus abgeändert werden. Das Format ist allerdings so kompliziert, daß ↔
ich
in dieser Version von WComm, den Benutzer raten möchte, dies beqüm vom Nachrichten Fenster aus vorzunehmen.

Aus diesem Grund werde ich auch die Synatx nicht bekanntgeben.

Für Profis, die dies aber dennoch möchten:

Schreibt an meine Adreese und schickt mir eine Leerdiskette und einen frankierten und an sich selbst adressierten Rückumschlag und ich sende euch die neuste Version von WComm zusammen mit der Syntaxbeschreibung zu.

1.7 wbench-start

WBench-Start:

Wenn man WComm von der WBench aus starten will, kann man anstelle der Cli-Argumente auch ToolTypes benutzen. Die folgenden ToolTypes werden von WComm unterstützt.

```
CONFIG
CX_POPUP
WPREFS
EPREFS
CX_PRIORITY
NOTIFY
```

Die Bedeutung der Argumente ist dieselbe, wie die, der Cli-Argumente. Allerdings muß man beachten, daß sie alle 'großgeschrieben' werden müssen oder ansonsten einfach ignoriert werden.

Die Syntax: (Die <>'s bitte nicht eingeben)

CONFIG=<Name der Konfigurationsdatei>

CX_POPUP=<YES|NO>

Man kann hier nur YES oder NO angeben.

WPREFS=<Name der Fensterlistendatei>

EPREFS=<Name der Ausschlußdatei>

CX_PRIORITY=<#> # Muß eine Nummer sein.

NOTIFY=<Name der Nachrichtendatei>

Das System erkennt und benutzt auch noch folgende ToolTypes:

TOOLPRI=<#>	Das ist die Priorität, die die WBench dem aufgerufenen Programm zuteilt. Bitte nicht mit der Commodity-Priorität verwechseln.
CLI	Dieses ToolType hat keine weiteren Angaben. Ist CLI angegeben öffnet die WBench sein 'command execute' Fenster und ermöglicht so eine Eingabe von CLI-Argumenten. Dann startet die WBench das Program als CLI-Programm.
DONOTWAIT	Dieses ToolType muß beim automatischen Start vom WBStartup Verzeichnis aus angegeben werden. Dies ist ähnlich dem CLI-Kommando Run und ermöglicht so ein starten des Programmes im Hintergrund. Wird dieses Argument nicht angegeben, wartet die WBench auf die Beendigung des Programmes, bevor die Startup weitergeführt wird.

1.8 hotkeys

HotKeys:

Die schon erwähnten Hotkeys müssen bestimmten Konventionen folgen. Jedes Hotkey besteht aus einem Qualifier und einem KeyCode.

Qualifier sind Tasten, wie Shift, Alt Control etc. KeyCodes sind alle anderen Tasten.

Ein Hotkey kann auch nur aus einem KeyCode bestehen, aber nicht aus nur einem Qualifier.

Für jede Funktion des Programmes muß eine andere HotKey-Kombination benutzt werden.

Es folgt nun eine Liste möglicher Hotkeys:

Mögliche Qualifier sind:

```

lshift      : Linke Shift
rshift      : Rechte Shift
shift       : Irgendeine Shift
lalt        : Linke Alt
ralt        : Rechte Alt
alt         : Irgendeine Alt
lcommand    : Linke Amiga (Commodore)
rcommand    : Rechte Amiga (Vorsicht bei Konflikten mit Menü-
              shortcuts)
command     : Irgendeine Amiga
control     : Control Taste

lbutton     : drücken der linken Maustaste
rbutton     : drücken der rechten Maustaste
midbutton   : drücken der mittleren Maustaste (falls man
              eine Maus mit 3 Tasten besitzen sollte)
diskinserted : Diskette einlegen
diskremove  : Diskette herausnehmen

```

Mögliche Keycodes sind:

```

a,b ... z   : jede alphanumerische
0,1 ... 9   : Taste
üäö?       : auch Umlaute
+-.,*/|\?... : oder jedes andere Zeichen
f1,f2 ... f10 : Die Funktions-Tasten
space       : Leer Taste
backspace   : Backspace Taste
up,down,left,right : Die Pfeiltasten
esc         : Die Escape Taste
del         : Die Delete Taste
numericpad  : Achtung: Es gibt einen Unterschied
              : zwischen Tasten auf der normalen Tastatur
              : und den Tasten auf dem Numerischen Ziffern-
              : block. Soll eine Taste des Ziffernblocks
              : angegeben werden, muß man 'numericpad #'
              : eingeben, wobei # eine Taste des Ziffern-
              : blocks sein muß. Z.B. 'numericpad 8' ist
              : nur die Taste '8' des Ziffernblocks und
              : nicht die Taste '8' der normalen Tastatur.
return      : Die Return Taste
esc         : Die Escape Taste

```

Beispiele:

```
shift control lbutton
```

Wird die Shift- zusammen mit der Control- Taste gedrückt und dann die linke Maustaste betätigt.

```
f10
```

Die Funktionstaste f10.

```
ralt numericpad 9
```

Die rechte Alt-Taste mit der 9 des Ziffernblocks.

1.9 liste der funktionen

Liste der Funktionen:

Hier folgt nun eine Liste aller Funktionen von WComm und deren voreingestellten HotKeys. Bei jeder neuen Version von WComm wird diese Liste erweitert.

Wie das Hauptfenster funktioniert, wird später noch erklärt.

Einige Funktionen z.B. die Kaskadierungsfunktionen benutzen einen vertikalen und horizontalen Wert für ihre Bewegungsfunktionen. Die Werte hängen aber vom gerade benutzten Zeichensatz der WBench ab. Da ich nicht weiß welcher Font gerade benutzt wird, spreche ich bei diesen Werten von Linker bzw. Oberer Offset.

POPUP: (Voreingestellt: 'control o')

Dieser Hotkey öffnet das Haupt Fenster von WComm. Auf diesem Fenster kann man alle Hotkeys editieren, aktivieren und deaktivieren. Die Konfigurationsdatei kann abgespeichert werden und eine neu geladen werden.

Dieses Fenster öffnet sich auch, wenn man das Exchange-Programm zum Anzeigen von WComm benutzt.

ENDE: (Voreingestellt: 'control q')

Dieser Hotkey beendet WComm sofort ohne Nachfrage usw. Ob dies funktioniert hat, kann man mit dem Exchange Programm kontrollieren.

Schließe Fenster: (Voreingestellt: 'control esc')

Dieser Hotkey versucht das momentan aktive Fenster zu schließen, indem es dem Fenster einen InputEvent mit CLOSEWINDOW flag sendet.

Dies funktioniert natürlich nur bei einem Fenster, mit Schließ-Gadget.

Maximal Fenster: (Voreingestellt: 'control f1')

Dies vergrößert ein Fenster zu seinem Maximum, daß in der Fenster-

struktur in den MaxWidth und MaxHeight Variablen definiert ist. Dies funktioniert natürlich nur bei Fenster, die ein SizeGadget haben.

Minimal Fenster: (Voreingestellt: `control f2`)

Dies verkleinert ein Fenster zu seinem Minimum, daß in der Fensterstruktur in den MinWidth und MinHeight Variablen definiert ist. Geht nur bei Fenstern mit SizeGadget.

Zurücksetzen: (Voreingestellt: `control f3`)

Wenn WComm zum Ersten mal irgendetwas an einem Fenster ändert, speichert WComm seine Ursprungswerte. Mit dieser Funktion wird der ursprüngliche Platz und die ursprüngliche Größe eines manipulierten Fensters wiederhergestellt. Dies funktioniert nur bei Fenstern, die schon mit WComm geändert wurden.

Zoom Window: (Voreingestellt: `control f5`)

Zoomen eines Fensters. Dies ist dasselbe, als wenn man auf das ZipGadget (Links neben dem DepthGadget) klickt. Die System Zip-Funktion speichert den Wert vor dem 'Zippen'. Bei einem erneuten 'Zippen', befindet sich das Fenster wieder auf dem Platz und mit der gleichen Größe, wie vor dem ersten 'Zippen'.

Verkleinere Breite: (Voreingestellt: `shift alt left`)

Dies verkleinert das Fenster horizontal. D.H. die Breite des Fenster wird um Linker Offset kleiner.

Verkleinere Höhe: (Voreingestellt: `shift alt up`)

Dies verkleinert das Fenster vertikal. D.h. die Höhe des Fenster wird um Oberer Offset verkleinert.

Verkleinere Beides: (Voreingestellt: `shift alt backspace`)

Die Breite wird um Linker Offset und die Höhe um Oberer Offset verkleinert.

Vergrößere Breite: (Voreingestellt: `shift alt right`)

Die Breite des Fenster wird um Linker Offset vergrößert.

Vergrößere Höhe: (Voreingestellt: `shift alt down`)

Dies vergrößert die Höhe des Fenster um Oberer Offset.

Vergrößere Beides: (Voreingestellt: `shift alt space`)

Dies vergrößert die Breite des Fensters um Linker Offset und die Höhe um Oberer Offset.

Bewegen zu links oben: (Voreingestellt: `lalt numericpad 7`)

Dies vergrößert/verkleinert das Fenster auf die Hälfte der Bildschirmbreite und -höhe und platziert es in der linken oberen Ecke des Bildschirms.

Bewegen zu rechts oben: (Voreingestellt: ``lalt numericpad 9``)

Dies vergrößert/verkleinert das Fenster auf die Hälfte der Bildschirmbreite und -höhe und platziert es in der rechten oberen Ecke des Bildschirms.

Bewegen zu links unten: (Voreingestellt: ``lalt numericpad 1``)

Dies vergrößert/verkleinert das Fenster auf die Hälfte der Bildschirmbreite und -höhe und platziert es in der linken unteren Ecke des Bildschirms.

Bewegen zu rechts unten: (Voreingestellt: ``lalt numericpad 3``)

Dies vergrößert/verkleinert das Fenster auf die Hälfte der Bildschirmbreite und -höhe und platziert es in der rechten unteren Ecke des Bildschirms.

Bewegen zum Zentrum: (Voreingestellt: ``lalt numericpad 5``)

Dies vergrößert/verkleinert das Fenster auf die Hälfte der Bildschirmbreite und -höhe und platziert es in der Mitte des Bildschirms.

Bewegen zum linken Zentrum: (Voreingestellt: ``lalt numericpad 4``)

Dies vergrößert/verkleinert das Fenster auf die Hälfte der Bildschirmbreite und -höhe und platziert es an dem linken Rand des Bildschirms vertikal zentriert.

Bewegen zum rechten Zentrum: (Voreingestellt: ``lalt numericpad 6``)

Dies vergrößert/verkleinert das Fenster auf die Hälfte der Bildschirmbreite und -höhe und platziert es an dem rechten Rand des Bildschirms vertikal zentriert.

Bewegen zum oberen Zentrum: (Voreingestellt: ``lalt numericpad 8``)

Dies vergrößert/verkleinert das Fenster auf die Hälfte der Bildschirmbreite und -höhe und platziert es an dem oberen Rand des Bildschirms horizontal zentriert.

Bewegen zum unteren Zentrum: (Voreingestellt: ``lalt numericpad 4``)

Dies vergrößert/verkleinert das Fenster auf die Hälfte der Bildschirmbreite und -höhe und platziert es an dem unteren Rand des Bildschirms horizontal zentriert.

Bewegen nach oben: (Voreingestellt: ``shift control up``)

Dies schiebt das Fenster um Oberer Offset nach oben.

Bewegen nach links: (Voreingestellt: ``shift control left``)

Dies schiebt das Fenster um Linker Offset nach links.

Bewegen nach rechts: (Voreingestellt: `shift control right`)

Dies schiebt das Fenster um Linker Offset nach rechts.

Bewegen nach unten: (Voreingestellt: `shift control down`)

Dies schiebt das Fenster um Oberer Offset nach unten.

Setze nach Preferences: (Voreingestellt: `control f4`)

Dies setzt ein Fenster zu den in der Voreinstellerdatei angegeben Werten zurück. Findet WComm keinen entsprechenden Eintrag in dieser Datei, passiert nichts.

Vergrößere bis rechts: (Voreingestellt: `control f6`)

Dies vergrößert die Breite des Fenster bis zur rechten Bildschirmgrenze.

Vergrößere bis unten: (Voreingestellt: `shift control f6`)

Dies vergrößert die Höhe des Fenster bis zur unteren Bildschirmgrenze.

Vergrößere zu Grenzen: (Voreingestellt: `control f7`)

Dies vergrößert die Breite des Fenster bis zur rechten Bildschirmgrenze und die Höhe zur unteren Bildschirmgrenze.

Fenster aufteilen (Tiling): (Voreingestellt: `control f8`)

Wenn man diese Funktion aufruft, versucht WComm alle Fenster des aktuellen Bildschirms so zu verschieben, dass es keine Überlappungszonen der Fenster mehr gibt.

Z.B. hat man 4 Shell Fenster auf dem WBench Screen, werden die Fenster auf die Hälfte der Breite und Höhe des Screens vergrößert/verkleinert und dann in die vier Ecken des Bildschirms verschoben. Hat man stattdessen 8 Fenster werden sie auf ein Viertel der Breite und Höhe gebracht. Die Positionen werden entsprechend angepasst.

Diese Funktion kann bis zu 16 (!) Fenster überlappungsfrei positionieren.

Ausprobiert ?

Man sieht, dass alle ungeraden Zahlen von Fenstern zuerst die oberen Reihen füllen. Hat man mehr als 16 Fenster werden die anderen einfach über die breits verschobenen platziert.

Tiling bringt ausserdem das gerade platzierte Fenster in den Vordergrund. Dies ist kein Standard und wird bei anderen Programmen dieser Art meistens nicht gemacht.

Ein weiterer Aspekt ist, dass wenn man viele Fenster (Shell oder Verzeichnisfenster) benutzt, muss man unter OS 2.xx eine ganze Zeit warten, bis das gerade verschobene Fenster wiederhergestellt worden ist. In dieser
↔
Zeit

muss eine Tiling Funktion normalerweise warten, bis der Refresh Vorgang ab
geschlossen ist. Stellt man sich vor man hat 10 Fenster, dann kann das
schon
eine geraume Zeit dauern (zum Mindest bei einem Standard Amiga).
Aus diesem Grund habe ich mich entschlossen, die Geschwindigkeit ein
bisschen
zu erhöhen und setzte Kurzfristig das NOCAREREFRESH Bit der
Fensterstruktur.

Nach Beendigung der Funktion werden die Fensterbits wiederhergestellt.

Alle Fenster, die mit den Auschlußdateieinträgen übereinstimmen, werden
von
dieser Funktion ausgeschlossen.

Kaskading von links nach rechts: (Voreingestellt: 'shift control f8')

Hier werden alle Fenster des aktuellen Bildschirms in Kaskadenart
positioniert. Angefangen von links oben nach rechts oben. Dabei
wird jedes Fenster um Linker Offset mehr nach rechts verschoben.

Alle Fenster, die mit den Auschlußdateieinträgen übereinstimmen, werden
von
dieser Funktion ausgeschlossen.

Kaskade von oben nach unten: (Voreingestellt: 'control f9')

Hier werden alle Fenster des aktuellen Bildschirms in Kaskadenart
positioniert. Angefangen von links oben nach links unten. Dabei
wird jedes Fenster um Oberer Offset unter das vorherige verschoben.

Alle Fenster, die mit den Auschlußdateieinträgen übereinstimmen, werden
von
dieser Funktion ausgeschlossen.

Kaskade horizontal&vertikal: (Voreingestellt: 'shift control f9')

Hier werden alle Fenster des aktuellen Bildschirms in Kaskadenart
positioniert. Angefangen von links oben nach rechts unten. Dabei
wird jedes Fenster um Linker Offset mehr nach links und um Oberer
Offset mehr nach unten verschoben.

Alle Fenster, die mit den Auschlußdateieinträgen übereinstimmen, werden
von
dieser Funktion ausgeschlossen.

Kaskade von unten nach oben: (Voreingestellt: 'control f10')

Hier werden alle Fenster des aktuellen Bildschirms in Kaskadenart
positioniert. Angefangen von rechts unten nach rechts oben. Dabei
wird jedes Fenster um Oberer Offset mehr über das vorherige ver-
schoben.

Alle Fenster, die mit den Auschlußdateieinträgen übereinstimmen, werden
von

dieser Funktion ausgeschlossen.

Akt. vorherigen Fenster: (Voreingestellt: ``lcommand left``)

Diese Funktion aktiviert das vorherige Fenster in der Fensterliste des aktuellen Bildschirms. Dabei werden nur Fenster berücksichtigt, die nicht in der Fenster Ausschlussdatei angegeben sind.

Akt. nächstes Fenster: (Voreingestellt: ``lcommand right``)

Diese Funktion aktiviert das nächste Fenster in der Fensterliste des aktuellen Bildschirms. Dabei werden nur Fenster berücksichtigt, die nicht in der Fenster Ausschlussdatei angegeben sind.

Vorh. Fenster nach vorn: (Voreingestellt: ``shift lcommand left``)

Diese Funktion aktiviert das vorherige Fenster in der Fensterliste des aktuellen Bildschirms und bringt es in den Vordergrund. Dabei werden nur Fenster berücksichtigt, die nicht in der Fenster Ausschlussdatei angegeben sind.

Nächs. fenster nach vorn: (Voreingestellt: ``shift lcommand right``)

Diese Funktion aktiviert das nächste Fenster in der Fensterliste des aktuellen Bildschirms und bringt es in den Vordergrund. Dabei werden nur Fenster berücksichtigt, die nicht in der Fenster Ausschlussdatei angegeben sind.

Fenster nach vorn: (Voreingestellt: ``lcommand return``)

Dies bringt das aktuelle Fenster in den Vordergrund.

Fenster nach hinten: (Voreingestellt: ``lcommand esc``)

Dies bringt das aktuelle Fenster in den Hintergrund.

Nächster Screen: (Voreingestellt: ``lcommand up``)

Dies bringt den vordersten Bildschirm ganz nach hinten.

Kill Window: (Voreingestellt: ``shift control esc``)

Diese Funktion schließt das aktuelle Fenster nach einer Sicherheitsabfrage.

!!! Vorsicht !!!

Bei dieser Funktion werden Fenster geschlossen, die von anderen Programmen hinterlassen wurden, die z.B. durch einen Programmfehler beendet wurden ↔
oder
Ähnliches.

Wenn man Fenster von laufenden Programmen schließt, kann dies zu einem Systemabsturz führen. Also sollte diese Funktionen nur in Notfällen ↔
benutzt
werden.

Öffne Shell: (Voreingestellt: `shift control s`)

Diese Funktion öffnet eine NewShell. Dabei benutzt die Shell den Pfad, der in der Startup-Sequenz angegeben wurde. Auch der Systemstack wird benutzt.

Öffne Ausschluß Fenster: (Voreingestellt: `shift control x`)

Öffnet das Ausschluß Fenster ohne das Hauptfenster zu öffnen.

Öffne Voreinsteller Fenster: (Voreingestellt: `shift control l`)

Öffnet das Voreinsteller Fenster ohne das Hauptfenster zu öffnen.

Öffne Uhren Fenster: (Voreingestellt: `shift control c`)

Öffnet das Uhren Fenster ohne das Hauptfenster zu öffnen.

Öffne Paletten Fenster: (Voreingestellt: `shift control p`)

Öffnet das Paletten Fenster ohne das Hauptfenster zu öffnen.

Öffne Nachrichten Fenster: (Voreingestellt: `shift control n`)

Öffnet das Nachrichten Fenster ohne das Hauptfenster zu öffnen.

Zusammenfassung:

Das Verschieben von Fenstern funktioniert nur bei Fenstern mit einer DRAG_BAR (lässt sich herausfinden, indem man versucht mit der Maus das Fenster zu verschieben).

Das Vergrössern und Verkleinern funktioniert nur bei Fenstern die ein GrössenGadget haben (unten rechts).

Die Zoom (Zip) Funktion funktioniert nur bei Fenstern mit ZipGadget.

In den Vordergrund/Hintergrund bringen funktioniert nur bei Nicht-Backdrop-Fenstern (Ein Backdrop Fenster ist z.B. das WorkBench Fenster der WBench 1.3).

Die Funktionen zum Aktivieren des nächsten und vorherigen Fensters und die Tiling und Kaskadierungsfunktionen funktionieren nur bei Fenster, die nicht in der Ausschlussliste stehen.

Die Setzte nach Preferences Funktion funktioniert nur bei Fenstern, die in der Fensterlistendatei stehen.

Die Reset Funktion funktioniert nur, wenn das Fenster vorher von WComm manipuliert worden ist (ausgenommen Kaskadieren und Tiling Manipulationen) ↔

Tiling und Kaskadierung Funktionen versuchen Fenster zu verschieben und in ihrer Grösse zu verändern. Hat ein Fenster kei GrössenGadget, wird das Fenster nur verschoben. Backdrop Fenster werden nicht berücksichtigt.

Die Voreinsteller- und die Ausschlussliste können ein Fenstertitel bei Angabe der ersten unverwechselbaren Buchstaben angeben. Man kann auch `Super` Gruppen angeben.

1.10 allgemein

Allgemein:

Bevor ich die Fenster näher erkläre, möchte ich noch etwas allgemeines zu der Benutzung der Benutzeroberfläche sagen.

Alle Funktionen können über Gadgets gestartet werden.

Für fast alle Gadgets existiert eine Taste, die man betätigen kann anstatt das Gadget mit der Maus zu aktivieren. Diese Taste ist unterstrichen dargestellt. Für einige wichtige Gadgets existiert eine solche Keyboardunterstützung nicht. Dies ist besonders bei wichtigen Funktionen implementiert, damit nicht aus Versehen wichtige Funktionen abgerufen werden können.

Die unterstrichenen Tasten unterscheiden nur dann zwischen Groß- und Kleinschreibung, wenn es beide Versionen mit unterschiedlichen Bedeutungen gibt.

Auf das Schließgadget des Fensters zu klicken ist genauso, als wenn man das Beenden Gadget benutzt.

Alle Funktionen sind jederzeit benutzbar. Jedes Fenster arbeitet asynchron. D.H. Man kann eine Einstellungen verändern und dies wird sofort sichtbar.

Für jedes Fenster existiert ein Hotkey, mit dem man ohne Umwege auf dieses Fenster gelangen kann. Ist ein Fenster schon geöffnet aber im Hintergrund, kann man das Fenster mit erneutem Betätigen des Hotkeys aktivieren und nach Vorne bringen.

1.11 os 2.04 gadgets

OS 2.04 Gadgets:

Da unter OS 2.04 eine Reihe neuer Gadgets eingeführt wurden, werde ich hier eine kurze Beschreibung geben.

BUTTON_GADGET:

Knopf Gadget - Wird durch einmaligen Druck mit der linken Maustaste invers dargestellt. Läßt man die Taste über dem Gadget los, wird die jeweilige Funktion gestartet.

Drücken der unterstrichenen Taste bewirkt das Gleiche.

CHECKBOX_GADGET:

Haken-Box - Dieses Gadget besitzt zwei Zustände aktiviert und nicht-aktiviert. Ist es aktiviert, ist ein kleiner Haken sichtbar, sonst nicht.

Drücken der unterstrichenen Taste bewirkt das Gleiche wie Anklicken mit Maus.

STRING_GADGET:

Texteingabefeld - Ermöglicht die Editierung des im Gadget stehenden Textes. Durch Drücken der RETURN - Taste, wird der Editiervorgang abgeschlossen.

Das Drücken der unterstrichenen Taste entspricht einem Aktivieren dieses Gadgets mit der Maus.

CYCLE_GADGET:

Hier wird eine Auswahl aus mehreren getroffen. Die momentane Auswahl wird angezeigt. Durch Klicken mit der linken Maustaste wird die nächste Auswahlmöglichkeit angezeigt.

Benutzt man die unterstrichene Taste, hat das denselben Effekt, wie ein Mausklick. Drückt man hingegen die unterstrichene Taste zusammen mit der Shift-Taste, wird nicht zur nächsten Auswahl, sondern zur vorherigen Auswahl gewechselt.

MX_Gadget:

Auswahl Gadget - Hier hat man die Alternative zwischen den dargestellten Möglichkeiten. Die eingedrückt dargestellte Möglichkeit ist aktiv. Ein Aktivieren einer anderen Auswahl, deaktiviert die erste.

SLIDER_GADGET:

Rollbalken - Der Körper (meist in schwarz) zeigt das Verhältnis der Anzahl aller Elemente zu den derzeitig dargestellten Elementen an. Durch Klicken unterhalb oder oberhalb des Körpers aber innerhalb der Umrandung, läßt sich ein Ausschnitt weiter nach oben (links) oder nach unten (rechts) rollen. Hält man die Maustaste gedrückt, während der Mauszeiger bewegt wird, kann man eine kontinuierliche Bewegung erreichen.

Auch hier hat die unterstrichene Taste, zusammen mit der Umschalttaste, eine Bewegung des Rollbalkens nach oben (links) zur Folge.

LISTVIEW_GADGET

Listengadget - Zeigt eine Liste an. Aktivierung mittels Mausklick.

Kein Tastaturäquivalent.

1.12 das haupt fenster

Das Haupt Fenster:

Wenn man den HotKey für das POPUP des Haupt Fensters benutzt, (Voreingestellt `control o`), öffnet sich das Haupt Fenster. Dies kann ebenfalls mit dem Exchange Programm durchgeführt werden, wenn man WComm auswählt und dann Show anklickt. Dieses Fenster wird auch beim Start geöffnet, wenn man CX_POPUP als CLI-Argument oder ToolType angibt.

Auf dem Haupt Fenster kann man die Konfigurationsdatei laden und abspeichern. Man kann alle anderen Fenster aufrufen und die Hotkeys abändern.

LISTVIEW Gadget

Das ListenGadget zeigt eine Liste aller Funktionen von WComm an. Man kann die Pfeil Gadgets benutzen, um in der Liste nach oben bzw. unten zu rollen. Klickt man auf eine dieser Zeilen, wird im Jetzige Textgadget die momentan benutzte HotKey-Kombination für die jeweilige Funktion angezeigt.

Edit (`e`)
Neu (`w`)

Klickt man auf Edit wird das Neu StringGadget aktiviert, in das man dann eine neu HotKey-Kombination eingeben kann, die dann die alte, bisher benutzte, ersetzen soll.

WComm versucht nun diesen neuen HotKey vom System anzufordern.

Gelingt dies, wird die neue Kombination im Jetzige Textgadget angezeigt und das StringGadget wird wieder inaktiviert.

Wird hingegen eine fehlerhafte Kombination eingegeben, oder kann WComm aus irgendeinen anderen Grund dieses HotKey nicht benutzen, so wird der voreingestellte HotKey benutzt. Auch dieser wird dann im TextGadget dargestellt.

Unter dem ListView Gadget sind noch zwei weitere Gadgets. Ist ein Eintrag des ListView Gadgets aktiviert, kann man auf Deaktiv drücken, um den HotKey zu deaktivieren. Aktivieren läßt sich ein Hotkey nur, wenn er deaktiviert ist.

Die Gadgets auf der rechten Seite des Fensters:

Öffnen (`o`)
Auswahl (`s`)
Sichern (`v`)
Sich Als (`a`)

Das Öffnen Gadget lädt die voreingestellte Konfigurationsdatei mit dem Namen `env:WComm.prefs`. Das Auswahl Gadget öffnet den Asl-Filereqüster, mit dem man dann eine andere Konfigurationsdatei

auswählen kann. Voreingestellt ist hier: `envarc:WComm.prefs`.

Mit dem Sichern Gadget werden die momentan benutzten HotKeys als Konfigurationsdatei unter den Namen `env:WComm.prefs` abgespeichert. Das SichAls Gadget öffnet wieder den AslFilereqüster mit voreingestelltem Namen `envarc:WComm.prefs`.

Zurück ('r')

Normal ('f')

Das Zurück Gadget benutzt den letzten HotKey, bevor er durch den Benutzer geändert wurde, während das Normal Gadget alle HotKeys zu den Voreingestellten Werten zurücksetzt und diese dann anschließend wieder aktiviert.

Es folgt nun eine Liste der Gadgets mit denen man auf die anderen Fenster von WComm gelangt.

Öffne Ausschluß Fenster ('\$\\times\$')

Dies öffnet das Ausschlussfenster, das später noch beschrieben wird.

Öffne Voreinsteller Fenster ('.')

Dies Gadget öffnet das Voreinstellerfenster. Auch dieses Fenster wird noch beschrieben.

Öffne Uhren Fenster (';')

Dies Gadget öffnet das Uhren Fenster. Später mehr dazu.

Öffne Paletten Fenster ('p')

Dies öffnet das Paletten Fenster, das später noch beschrieben wird.

Die nun folgenden zwei Gadget besitzt jedes Gadget in der ein oder anderen Form. Jedes dieser zwei Gadgets schließt das aktuelle Fenster.

Verbergen ('h')

Beenden ('q')

Mit Verbergen kann man das Fenster schließen und WComm bleibt weiterhin im Hintergrund aktiv.

Im Gegensatz dazu wird mit dem Beenden Gadget das ganze Programm beendet. Hat man die Einstellungen verändert und klickt man auf Beenden, öffnet sich ein Reqüster, der fragt, ob man die Konfigurationen vorher abspeichern will oder nicht. Hat man die Konfiguration nicht verändert fragt der Reqüster lediglich, ob man das Programm auch wirklich beenden will.

Der Hotkey für das Beenden von WComm entspricht im Wesentlichen dem Beenden Gadget, mit der Ausnahme, daß kein Reqüster dargestellt wird, egal ob irgendeine Einstellung verändert wurde oder nicht.

1.13 das ausschluß fenster

Das Ausschluß Fenster:

In diesem Fenster kann man die Ausschlussliste abändern, erweitern, speichern und laden.

Das Fenster kann sowohl vom Einsteller Fenster oder per Hotkey geöffnet werden.

Fenstertitel, die mit den in der Liste stehenden Fenstertiteln übereinstimmen, werden von bestimmten Funktionen wie z.B.

 Tilen, Kaskadieren, Vor-, Zurückblättern etc.

ausgeschlossen.

Die Listeneinträge bestehen aus AmigaDOS Suchmustern. Dabei muß man beachten, daß keine doppelten Einträge verwendet werden, da die Liste von WComm immer von oben nach unten auf Übereinstimmungen untersucht wird. Bei zwei gleichnamigen Einträgen wird also immer nur die obere benutzt.

Die Gadgets des Fensters:

LISTVIEW:

Dieses ListenGadget zeigt alle momentanen Ausschlussfenster an. Wählt man einen Eintrag aus, so wird er in das StringGadget unter dem ListenGadget kopiert und gleichzeitig wird das Gadget aktiviert. Wenn man eine Veränderung nun mit Return abschliesst, wird der alte Name durch die neu Eingabe ersetzt und in die Liste aufgenommen.

NEU:

Hiermit wird ein neuer Eintrag am Ende der Liste eingefügt, der unnamed genannt wird. Zugleich wird er in das StringGadget kopiert, das dann auch gleich aktiviert wird. Es steht also sofort zur Editierung bereit.

WEG:

Hiermit wird der gerade dargestellte Eintrag von der Liste entfernt.

DEAKTIV:

Man kann einen Eintrag hiermit deaktivieren. Wird ein Eintrag deaktiviert, erscheint am Anfang des Namens ein '>>'. Man kann nur Einträge deaktivieren, die vorher aktiviert sind. Auch die Aktivierung bzw. Deaktivierung eines Eintrages wird in der Ausschlußdatei mit abgespeichert.

AKTIV:

Dies (re)aktiviert einen vorher deaktivierten Eintrag. Das '>>' verschwindet. Es lassen sich nur Einträge aktivieren, die vorher deaktiviert wurden.

OEFFNE:

WComm versucht die voreingestellte Ausschlussdatei zu laden. ('env:XWindow.prefs', 'envarc:XWindow.prefs'). Kann sie nicht geladen werden, bekommt der Benutzer eine Fehlernachricht.

AUSWAHL:

Dies öffnet den FileReqüster, um eine Datei zum Laden auszuwählen.

SAVE:

Dies speichert die aktuelle Ausschußliste unter dem voreingestellten Namen 'env:XWindow.prefs' ab.

SICHALS:

Dies öffnet den FileReqüster, um einen Dateinamen zum Speichern auszuwählen. Voreingestellt ist hier envarc:XWindow.prefs. Man kann natürlich auch einen anderen Namen wählen.

BENUTZE:

Dies schliesst das Fenster.

QUIT:

Dies schliesst das Fenster und fragt nach, ob man WComm beenden will.

1.14 das voreinsteller fenster

Das Voreinsteller Fenster:

In diesem Fenster kann man die Voreinstellungsliste erweitern, speichern und laden usw.

Man kann hier Fenstertitel mit Koordinaten und Größenangaben angeben, die mittels der 'Setze nach Preferences' Funktion benutzt werden.

Das Fenster kann sowohl vom Einsteller Fenster als auch per Hotkey geöffnet werden.

Die Gadgets des Fensters:

LISTVIEW:

Dieses ListenGadget zeigt alle momentan bekannten Voreinstellungen an.

Wählt man einen Eintrag aus, so wird er in das StringGadget unter dem ListenGadget kopiert und gleichzeitig wird das Gadget aktiviert. Die Linke -,Obere Ecke-, die Breite- und Höhe- Werte werden in die IntegerGadgets kopiert und dargestellt.

Wenn man nun eine Änderung mit Return abschliesst, wird die Änderung übernommen.

Die Fenstertitel dürfen AmigaDOS Suchmuster enthalten, aber keine Leerzeichen.

NEU:

Hiermit wird ein neuer Eintrag am Ende der Liste eingefügt, der unnamed genannt wird. Zugleich wird er in das StringGadget kopiert, das dann auch gleich aktiviert wird. Es steht also sofort zur Editierung bereit.

WEG:

Hiermit wird der gerade dargestellte Eintrag von der Liste entfernt. Auch hier gilt: zwei gleichnamige Einträge sollten vermieden werden.

DEAKTIV:

Man kann einen Eintrag hiermit deaktivieren. Wird ein Eintrag deaktiviert, erscheint am Anfang des Namens ein '>>'.
Man kann nur Einträge deaktivieren, die vorher aktiv sind.

AKTIV:

Dies (re)aktiviert einen vorher deaktivierten Eintrag. Das '>>' verschwindet. Es lassen sich nur Einträge aktivieren, die vorher deaktiviert wurden.

ÖFFNEN:

WComm versucht die voreingestellte Voreinstellerdatei zu laden. ('env:ZWindow.prefs', 'env:ZWindow.prefs'). Kann sie nicht geladen werden, bekommt der Benutzer eine Nachricht.

AUSWAHL:

Dies öffnet den FileReqüster, um eine Datei zum Laden auszuwählen.

SICHERN:

Dies speichert die aktuelle Liste unter dem voreingestellten Namen 'env:ZWindow.prefs'

SICHALS:

Dies öffnet den FileReqüster, um einen Dateinamen zum Speichern auszuwählen. Voreingestellt ist hier 'env:ZWindow.prefs'.
Man kann natürlich auch einen anderen Namen wählen.

LIECKE, T ECKE, WEITE, HOEHE:

Diese Gadgets zeigen die Werte für den aktuellen Fenstereintrag dar.

LI ECKE:

Die linke Position des Fensters.

T ECKE:

Die obere Position des Fensters.

WEITE:

Die Weite des Fensters.

HOEHE:

Die Höhe des Fensters.

Alle Eingaben werden sofort übernommen.

USE:

Dies schließt das Fenster.

QUIT:

Dies schließt das Fenster und fragt nach, ob man WComm beenden will.

1.15 das uhren fenster

Das Uhren Fenster:

Das Fenster kann sowohl vom Hauptfenster oder per Hotkey geöffnet werden.

Mit diesem Fenster kann man das Aussehen der Bildschirmuhr verändern oder sie schließen. Voreinstellungsmäßig ist die Uhr angeschaltet und zeigt kein Datum an und die Zeit wird in Stunden:Minuten angezeigt.

Bemerkung: Will man die Nachrichtenfunktionen (siehe NACHrichten Fenster) benutzen, muß die Uhr eingeschaltet sein.

Alle Änderungen werden in der folgenden Sekunde auf die Titeluhr übertragen.

Nun zur Erklärung der einzelnen Gadgets:

Uhren Typ ('T' oder 't')

Die Stellung dieses CycleGadgets bestimmt, ob die Uhr geöffnet werden soll oder nicht. Voreinstellung: ein.

Zeit Format ('f' oder 'F')

Hiermit kann man die Darstellung der Zeit bestimmen.
Als Auswahl hat man hier:

ST:MI	- Stunden:Minuten (VOREINSTELLUNG)
ST:MI:SE	- Stunden:Minuten:Sekunden
24ST:MI	- 24-Stunden:Minuten
24ST:MI:SE	- 24-Stunden:Minuten:Sekunden

Datum Typ ('y' oder 'Y')

Hiermit kann man die Darstellung des Datums bestimmen.
Mögliche Darstellungsformen:

Kein Datum	- es wird kein Datum angezeigt (VOREINSTELLUNG)
WT Tag-Mon-Jahr	- Wochentag Tag-Monat-Jahr
KurzWT Tag-Mon-Jahr	- kurzer Wochentag Tag-Monat-Jahr
Tag-Mon-Jahr	- kein Wochentag Tag-Monat-Jahr

Weite ('w' oder 'W')

Dies ist die Breite des Fensters. Man kann hier einen Wert zwischen 150 und 450 bestimmen. Voreingestellt: 150

Linke Ecke ('l' oder 'L')

Dies ist die linke Position der Uhr.
Voreingestellt: 0

Obere Ecke ('o' oder 'O')

Dies ist die obere Position der Uhr.
Voreingestellt: 0

Alarm ein ('n' oder 'N')

Die Uhr hat die Möglichkeit zu einer bestimmten Uhrzeit einen Alarm auszulösen. Der Bildschirm blinkt dann kurz auf und kann so den Benutzer auf eine bestimmte Uhrzeit aufmerksam machen. Dieses Gadget entscheidet, ob die Alarmfunktion eingeschaltet ist oder nicht.
Voreingestellt: aus

Beep jede Std ('p' oder 'P')

Zusätzlich zur Alarmfunktion kann WComm auch jede Stunde den Bildschirm kurz aufblitzen lassen.
Voreingestellt: aus

Al Std ('a' oder 'A')

Dies ist die Stunde, an der der Alarm ausgelöst werden soll.
Voreingestellt: 0

Al Min ('m' oder 'M')

Dies ist die Minute, an der der Alarm ausgelöst werden soll.

Voreingestellt: 0

Benutzen ('u' oder 'U')

Die aktuellen Werte werden benutzt.

Abbruch ('c' oder 'C')

Die aktuellen Werte werden nicht benutzt, stattdessen werden die Werte benutzt, die vor dem Aufruf dieses Fensters aktuell waren.

Alle Einstellungen dieses Fenster werden in die Konfigurationsdatei mit abgespeichert.

Wird die Uhr durch betätigen des Schließgadgets geschlossen, hat dies auf das geöffnete Fenster keinen Einfluß. Wird das Fenster aber erneut aufgerufen, ist der Uhrentyp auf 'keine Uhr' eingestellt.

1.16 das nachrichten fenster

Das Nachrichten Fenster:

Dieses Fenster kann auch vom Haupt Fenster oder per Hotkey gestartet werden.

Alle Funktionen dieses Fensters beschäftigen sich mit der Nachrichtenliste. Sie kann hier geändert, geladen und abgespeichert werden.

Die Nachrichtenliste beinhaltet einen Titel, einen Nachrichtentext, das Datum und die Uhrzeit der Nachricht und ein optionales Kommando, daß auf dreierlei Arten aufgerufen werden kann (EXECUTE, RUN und AREXX).

Ein Nachrichtenrequester kann zu einer bestimmten Zeit geöffnet werden und eine Nachricht mitteilen. Optional kann dieser Requester auch ein vorher eingegebenes Kommando ausführen. Der Zeitpunkt des Requesters kann mit den Stunden, Minuten, Monat oder Tag Slidergadgets eingestellt werden. Jedes einzelne dieser Angaben kann dabei unabhängig voneinander aktiviert und deaktiviert werden. Eines dieser Angaben muß aber immer aktiviert sein, sonst wird es einfach ignoriert. Mit diesem Konzept ist es möglich, eine Nachricht jede Stunde oder nur einmal anzeigen zu lassen.

Gadget Beschreibung:

LISTVIEW Gadget:

Dieses Gadget stellt die Titel der Requester dar. Dieser Titel wird dann in der Titelzeile der Requester angezeigt.

NEU:

Dieses Gadget fügt einen neuen leeren Eintrag an das Ende der Liste an.

WEG:

Dieses Gadget entfernt den aktuellen Eintrag aus der Liste.

AKTIV:

Dieses Gadget aktiviert den aktuellen Eintrag. Das geht nur bei zuvor deaktivierten Einträgen.

INAKTIV:

Dieses Gadget deaktiviert den aktuellen Eintrag. Dies funktioniert nur bei aktiven Einträgen.

NAME:

Dieses StringGadget enthält den Namen des aktuellen Eintrages.

NACHRICHT:

Dieses StringGadget enthält den Text des Nachrichtenrequesters. Dabei können mehrere Zeilen mittels einem '\n' angegeben werden. Die Nachricht darf bis zu 256 Zeichen lang sein.

KOMMANDO:

Soll ein Kommando ausgeführt werden, kann man es hier angeben. Um dieses Kommando benutzen zu können, muß aber noch die Aktion CheckBox aktiviert sein.

REQUESTER ZEIT/DATUM:

Wenn diese CheckBox aktiviert ist, wird der Requester angezeigt.

STD und 1:

Dies ist die Stundenangabe, die für das Erscheinen des Requesters benutzt werden soll. Dieser Wert wird nur benutzt, wenn die '1' CheckBox aktiviert ist.

MIN und 2:

Dies ist die Minutenangabe, die für das Erscheinen des Requesters benutzt werden soll. Dieser Wert wird nur dann benutzt, wenn die '2' CheckBox aktiviert ist.

MON und 3:

Dies ist die Monatsangabe, die für das Erscheinen des Requesters benutzt werden soll. Dieser Wert wird nur dann benutzt, wenn die '3' CheckBox aktiviert ist.

TAG und 4:

Dies ist die Tagesangabe, die für das Erscheinen des Requesters benutzt werden soll. Dieser Wert wird nur dann benutzt, wenn die '4' CheckBox

aktiviert ist.

BEMERKUNG: Es muß zumindest einer dieser Werte benutzt werden.

START:

Dieses CycleGadget entscheidet, wie das Kommando gestartet werden soll. Dabei gibt es drei Arten:

EXECUTE

Dies startet das angegebene Kommando als ein Programm oder als CLI Kommando. Dabei wartet WComm, bis das Kommando ausgeführt worden ist. Alle evtl. Ausgaben werden zu einem dann erscheinenden CLI Fenster kopiert. Werden keine Ausgaben gemacht, bleibt das Fenster geschlossen~(AUTO SHELL).

BACKGROUND

Dies startet ein Kommando als Hintergrundprozeß. Somit ist keinerlei Ein- und Ausgabe mehr möglich.

AREXX

Dies startet RX mit der angegebenen Kommandozeile.

Beachten: Alle drei dieser Möglichkeiten, benutzen den Pfad und Stack der BootShell.

EINM:

Wenn diese CheckBox aktiviert ist, wird die Nachricht nur einmal angezeigt. Man kann auch Nachrichten benutzen, die z.B. jede Stunde angezeigt werden. Wenn man dieses nicht wünscht, kann man Einm aktivieren. Erst bei einem erneuten Start von WComm wird dieser Requester erneut aktiviert (Siehe auch Beispiele).

AKTION:

Wenn diese CheckBox aktiviert ist, wird das Kommando, daß in dem KommandoStringgadget steht, beim Aufruf der Nachricht gestartet.

ÖFFNE:

Dies lädt die voreingestellte Nachrichtendatei mit dem Namen 'env:notify.prefs'. kann sie nicht gefunden werden, wird versucht die 'envarc:notify.prefs' Datei zu laden.

WÄHLE:

Dies öffnet den FileRequester, und ermöglicht so daß Laden einer anderen Nachrichtendatei.

SICHERN:

Dies speichert die Nachrichtendatei unter den voreingestellten Namen ab.

SICHALS:

Dies öffnet den FileRequester und ermöglicht so die Daten unter einem anderen Namen abzuspeichern.

BENUTZE:

Dies benutzt die Veränderungen und schließt das Fenster.

ENDE:

Dies fragt den Benutzer, ob er WComm beenden möchte.

BEISPIELE:

=====

Da die Nachrichtenverwaltung etwas komplizierter ist, möchte ich hier einige mehr oder weniger sinnvolle Beispiele zeigen.

1. WComm soll zu Weihnachten eine Nachricht ausgeben. Dabei soll kein Kommando ausgeführt werden:

Name: !!! FROHE WEIHNACHTEN !!!

Nachri: Ich wünsche Frohe Weihnachten|und hoffe, daß du viele
Geschenke|bekommen wirst

Mon: 12 3: aktiviert

Tag: 24 4: aktiviert

Einm.: aktiviert

Hier sieht man den Sinn des Einm. Gadgets. Wenn es nicht aktiviert ist, kommt dieser Requester genau einmal pro Minute. Aktiviert, kommt er erst nächstes Jahr Weihnachten wieder.

2. WComm soll eine Nachricht jede Stunde zum Ende der ersten viertel Stunde anzeigen, daß es Zeit zu einer Pause ist.

Name: !!! Pausen Zeit !!!

Nachri: Okay Junge|Jetzt ists zeit für eine Tasse Kaffe| oder
lieber Tee?

Min: 15 2: aktiviert

3. WComm soll automatisch jeden 3. Tag eines jeden Monats ein Backup starten. Man möchte zuerst eine Nachricht erhalten, die dieses einem mitteilt, damit man alle anderen Computeraktivitäten stoppen kann. Ich gehe hier davon aus, daß man ein Programm mit dem Namen Backup im aktuellen Suchpfad der Shell hat.

Name: !!! Backup Zeit !!!

Message: Okay Junge !| Jetzt ists an der Zeit, ein| neues Backup zu starten

Kommand: Backup DH0:

Tag: 3 4: aktiviert

Einm: aktiviert

Aktion: aktiviert

Man kann natürlich auch alle Geburtstage der Freunde eingeben und sich rechtzeitig bescheid geben lassen, um noch nötige Geschenke zu kaufen etc. Auf diese Art wird man nie wieder einen Geburtstag vergessen.

1.17 das paletten fenster

Das Paletten Fenster:

Das Fenster kann sowohl vom Haupt Fenster als auch per Hotkey geöffnet werden.

Alle auf diesem Fenster durchzuführenden Funktionen beschäftigen sich mit der Modifizierung der WBenchfarben. Da unter OS 3.0, die ich auch benutze, eine Anzahl von bis zu 256 WBench Farben möglich ist, fielen Palettenprogramme gleicher Art leider aus. So habe ich mich entschlossen, auch ein Palettenfenster einzubaün.

Dieses Palettenfenster bietet alle Standardfunktionen von ähnlichen Programmen, die wohl weitestens bekannt sein dürften. Besonderheit ist bei diesem Fenster, daß es automatisch erkennt, wie viele Farben gerade aktiviert sind. Die Anzahl der unterstützten Farben reicht von 2 bis 256.

Die Funktionen dieses Fenster sind schnell erklärt und wahrscheinlich kennt jeder ähnliche Programme. Dennoch hier eine Auflistung der Funktionen und der entsprechenden Gadgets:

PALETTEN Gadget (RETURN)

Dies stellt die gerade aktivierte Farbe dar. Bei OS 2.04 ist die aktuelle Farbe links neben den anklickbaren Bereichen nocheinmal dargestellt. Alle Veränderungen wirken sich immer nur auf die aktuelle Farbe aus.

RED ('r' oder 'R')

GREEN ('g' oder 'G')

BLUE ('b' oder 'B')

Diese SliderGadgets verändern den Rot-, Blau- oder Günanteil der aktuellen Farbe. Zunächst habe ich die Intervalle (Level) hier bei den üblichen 16 Grundfarben belassen. Bei neüren Versionen werde ich aber alle 256 Grundfarben als Intervalle zulassen. (zur Kompati-

bilität mit OS 2.04).

Die kleinen Buchstaben vergrößern den Farbanteil, während die großen den Farbanteil verringern.

Kopie ('o')

Wird dieses Gadget betätigt, verändert sich der Mauszeiger zu einem TO. Dies zeigt an, daß keine anderen Eingaben, als ein Klick auf das Palette Gadget erlaubt sind. Wird das Paletten Gadget angeklickt, wird die vorherige aktuelle Farbe zur jetzigen kopiert.

Austausch ('a')

Diese Funktion ist ähnlich der vorherigen, nur daß hier die Farben ausgetauscht werden.

Undo ('d')

Wenn das Paletten Fenster zum Ersten mal geöffnet wird, läßt WComm die aktuellen Farben ein. Wird Undo betätigt, werden alle Farben auf diese Werte zurückgesetzt, also auf die Werte vor dem ersten Aufruf des Paletten Fensters.

Zurück ('r')

Wurde die aktuelle Farbe verändert, wird sie auf den vorherigen Wert zurückgesetzt.
Dies funktioniert für jede Farbe einzeln und unabhängig.

Benutze ('u')

Benutzt die neun Farbeinstellungen und schließt das Fenster.

Abbruch ('c')

Benutzt die vorherigen Einstellungen und schließt das Fenster.

Bemerkungen: Werden die aktuellen Farben verändert, werden sie nur im ViewPort des Screens zwischengespeichert. Werden Programme wie z.B. die Palette der WBench aufgerufen werden die Farben auf die ursprünglichen Werte zurückgesetzt. Dies ist kein BUG, sondern Absicht.

1.18 die uhr

Die Uhr:

Alle Einstellungen der Uhr können mit dem Uhren Fenster abgeändert werden.

Die Uhr kann mittels Aktivierung des CloseGadgets geschlossen werden.
Um es wieder zu öffnen, muß man das Uhren Fenster benutzen.

Die Uhr kann nicht mit der Maus verschoben werden. Will man die Position verändern, kann man dies auf dem Uhren Fenster tun.

Die Darstellung der Uhr wird einmal pro Sekunde erneuert. In der Zwischenzeit wird im Hintergrund gewartet, so daß keine wertvolle CPU-Zeit vergeudet wird.

1.19 fehlerbehandlung

Fehlerbehandlung:

WComm wurde komplett unter OS 3.0 geschrieben. Dabei wurde Enforcer benutzt. Ich hoffe alle Enforcer-Hits behoben zu haben. Da aber auch bei bester Kontrolle einige Fehler passieren können, wurde eine Fehlerbehandlung in WComm miteinbezogen.

Die meisten auftretenden Fehler dürften von Fehlbedienungen herrühren. Auch sind mögliche weitere Gründe falsche Konfigurationsdateien und ungewollte Beeinflussungen mit anderen nicht so system-konformen Programme möglich.

ALLGEMEIN:

Bei jedem auftretenden Fehlern, versucht WComm den Benutzer einzubeziehen. D.h. der Grund für das Aufkommen des Fehlers wird dem Benutzer immer mitgeteilt. Bei manchen Fehlern kann der Benutzer entscheiden, ob er das Programm beenden möchte oder nicht. Bei ernsteren auftretenden Fehlern kann WComm unter Umständen nicht fortfahren und das Programm wird automatisch beendet.

LEICHTERE FEHLER:

Bei leichteren nicht ernsten Fehlern, wird der Benutzer nur mittels einem Requäter auf den Fehler aufmerksam gemacht. Nach Bestätigen des Requäters wird WComm fortgesetzt.

ERNSTERE FEHLER:

Ist der aufgetretene Fehler ernster, läßt der Requäter zwei Auswahlmöglichkeiten zu. Das Programm kann fortgesetzt werden oder aber beendet.

Ich möchte dem Benutzer empfehlen, eine solche Mitteilung möglichst mit einem Abbruch zu beantworten, da ein korrektes Arbeiten nach einem solchen Fehler nicht immer möglich ist.

FATALE FEHLER:

Tritt hingegen ein fataler Fehler auf, so kann WComm nicht fortgesetzt werden und der Benutzer wird nur informiert. Nach Bestätigen des Requäters wird WComm beendet.

UNBEKANNTE FEHLER:

Tritt ein unbekannter Fehler auf, sollte man zuerst nachschaun, ob WComm

korrekt installiert ist und man die folgenden Bibliotheken im Libs:-Verzeichnis hat:

```
asl.library
commodities.library
```

Kann der Fehler auch dann nicht gefunden werden, sollte man sich diese Anleitung nocheinmal gut durchlesen. Vielleicht entdeckt man hier die Ursache für das Problem. Kann man die Fehlerursache auch hier nicht finden, so kann man mir auch schreiben. Ich werde versuchen das Problem zu lösen. Aber dazu bitte eine korrekte Schilderung des Fehlers, evtl. anderer laufender Hintergrundprogramme, Computertyp, Kickstartversion, Erweiterungen, Speicher und die Version von WComm angeben.

Damit wird die Fehlersuche vereinfacht.

1.20 technisches

Technisches:

WComm ist ein Standard Kommodity. Es besitzt alle Fähigkeiten und Funktionen, die für ein Commodity vorgeschrieben sind. Mit dem Tool Exchange kann man das Verhalten von WComm voll kontrollieren.

Geschrieben und entwickelt wurde WComm mitfolgenden Programmen:

```
AztecC Compiler Version 5.2a (@ MANX),
TurboText (@ OXXI),
SnoopDos (@ Eddi Carroll)
GadToolsBox (@ Jaba Development),
Enforcer (@ Michael Sinz),
DMake (@ Matt Dillon)
```

Meinen herzlichen Dank an die Programmierer dieser hervorragenden Programme.

Meinen Dank auch an Mark Wesiheit, der mir bei der Fehlerbeseitigung geholfen hat. ↔

Anstelle der mitgelieferten Startup-Routinen, benutze ich einen eigenen Startup, ↔
der alle 1.3 Features durch die noch besseren 2.0 (37++)ersetzt. Dies spart ↔
einen

Haufen KBytes an Programm Code ein.

Dennoch ist WComm relativ groß geworden. Die Größe aller Listen ist nur vom ver- ↔

fügbaren Speicher abhängig. Alle Strukturen werden dynamisch verwaltet.

1.21 liste der dateien

Liste der Dateien:

```
WComm           :   Englischs Programm
WComm.info      :
WComm_G         :   Deutsches Programm
WComm_G.info    :
WComm.doc       :   Die englische Dokumentation
WComm.doc.info  :
WComm.dok       :   Die deutsche Dokumentation
WComm.dok.info  :
WComm.guide     :   Englische AmigaGuide ® Datei
WComm.guide.info :
WComm_G.guide   :   Deutsche AmigaGuide ® Datei
WComm_G.guide.info :
History.log     :   Eine History-Datei
History.log.info :
WComm.prefs     :   Eine Beispiel Konfigurationsdatei
WComm.prefs.info :
ZWindow.prefs   :   Eine Beispiel Voreinstellerdatei
ZWindow.prefs.info :
XWindow.prefs   :   Eine Beispiel Ausschlussdatei
XWindow.prefs.info :
Notify.prefs    :   Eine Beispiel Nachrichtendatei
Notify.prefs.info :
```

1.22 adresse

Adresse:

Für alle Arten von Post, Bugs, Geschenke and Kritik ...

```
Hans-Peter Günther
Grüner Weg 5
W-3501 Fuldata 2
GERMANY
```

euer H.P.G