

WComm

COLLABORATORS

	<i>TITLE :</i> WComm		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 10, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	WComm	1
1.1	main	1
1.2	what is wcomm	2
1.3	system requirements	4
1.4	legal stuff	4
1.5	installation and start	5
1.6	cli-start	6
1.7	wbench-start	9
1.8	hotkeys	10
1.9	list of functions	11
1.10	general	17
1.11	os 2.04 gadgets	18
1.12	the main window	19
1.13	the exclusion window	20
1.14	the window prefs window	22
1.15	the clock prefs window	23
1.16	the palette window	25
1.17	the notify window	26
1.18	the clock window	30
1.19	error handling	31
1.20	technical	32
1.21	list of files	32
1.22	address	33

Chapter 1

WComm

1.1 main

Documentation

WComm Version 1.80

WComm the Window manipulation tool for the Amiga

Index:

What Is WComm

Overview and general description

System Requirements

which OS ?

Legal Stuff

Read this !

Installation and Start

Important

CLI-Start

Cli arguments and options

WBench-Start

Tooltypes

HotKeys

List of available combinations

List Of Functions

Short listing of all 50 (!)

General

OS 2.04 Gadgets

Short description

The Main Window

The Exclusion Window

The Window Prefs Window

The Clock Prefs Window

The Palette Window

The Notify Window

The Clock Window

Error Handling
 Important

Technical
 Not only interested for programmers

List Of Files
 Read This !

Address
 For all gifts, presents, money etc...

1.2 what is wcomm

What Is WComm:

WComm is a standard commodity which enables you to manipulate windows in different ways. WComm has 43 window manipulation functions. You are able to:

- move,
- enlarge,
- minimize windows,
- set windows to specified size and positions,
- reset,
- close,
- zip,
- tile,
- cascade,
- bring to front and back windows,
- you can change to every screen and window,
- etc..

WComm has also the ability to notify the user about special dates. Therefor you can insert the dates and times of this events in the notify window. When the specified time is come, a requester will be opened and an optional command can be started (EXECUTE, RUN or AREXX).

WComm has also a user friendly clock with different kinds of date and time formats. It has an alarm function and is fully user definable.

WComm can exclude several given windows from some functions or even include only some windows for this functions.

WComm can set a window to a special position and size that can be given by a settings list.

WComm has a palette window which enables the user to manipulate the wbench colours upto 256 colours.

Each hotkey can be changed to the users needs.

WComm can open a shell via hotkey which uses the path and stack which is defined in the startup-sequence.

WComm can kill windows which are left from other programs.

WComm is a commodity and full controllable with exchange.

WComm is full font independent. It sizes all windows and gadgets dependent to the size of the current used wbench font. Also the tile and cascade functions uses the height of the window title bars.

You will see, WComm has a lot of features. So will not be surprised when you hear WComm's size is about 80 kb. But you can spare a lot of other commodities if you are using WComm. Such as

- IntuiHelp,
- clocks,
- cron programs,
- palette programs,
- tilers and cascade programs,
- window zappers etc...

WComm has a full gadtools user interface and enables the user to call all functions with the mouse or even the keyboard.

All windows and functions are fully asynchron that means all modifications of the settings will get visual immediate.

WComm is controlled by several configuration files.

For each kind of functions exists a different configuration file. There is an exclude, a window settings and a notify configuration file.

All files have two default names which can be loaded automatically at startup.

Also WComm has aother configuration file which contains all values of the hotkeys, window positions, clock settings etc...

All configuration files can be modified, edited, loaded and saved from the several windows.

To get this more comfortable some configuration files can be edited directly by the user with an text editor (see the sections about the windows).

Each hotkey can be changed and saved from the main window. The default hotkey to open the main window is 'control o'.

More:

WComm just supports some features of the new OS 3.0. So the palette window is able to handle upto 256 different colours. WComm is able to recognize if running on an OS 2.04 or OS 3.0 system.

The design of some windows and gadget may be different under OS 3.0.

The palette gadget owns no more indicator. The image of the currently activated colour is displayed recessed.

The listview gadgets doesn't display a text gadget which indicates the currently selected entry. The selected entry is indicated with an inverse image.

1.3 system requirements

System Requirements:

WComm needs OS 2.04 or better to run, because it is a standard commodity. It is tested under OS 2.04 and 3.0 and runs (I hope so :-)) with no errors.

The memory usage of WComm is much dependent to the sizes of the several lists. But it should run with no problems on Amigas with ≥ 1 MB memory. To save although memory you can disable some functions, i.e. the notify, and exclude functions. Also each hotkey which is disabled spares some bytes, which will otherwise allocated by the system.

WComm handles all resources and structures dynamically. That means: memory will only be allocated when needed. If some structures will not be needed anymore the memory for this will be given back to the system.

1.4 legal stuff

Legal Stuff:

This program ~and all files come with it, also the docfiles, are released as FREEWARE.

this means:

everybody may use WComm,

everybody may copy WComm, if:

1. You copy all included files, see list at the end of the

- doc file.
2. No changes are made, this means the program and all included files must stay in the original form.
Exception: you may archive it; but in this case you must also archive the whole package.
 3. If you decompile it, you have to include a notice to the original author. If you also want to (re)distribute it in any new form, you have to send me a copy.
 4. You may not sell WComm.
Exception: you're allowed to take a small copy fee, which should not extend 5US\$.
 5. If you want to include WComm or one of the included files as a part of a commercial package, you must have written permission from the author (address see below).

I take no warranty or guaranty for the function of this program.
The whole risk of using WComm stays only by the user.

1.5 installation and start

Installation and Start:

Installation:

The installation is very easy. First you have to copy all needed configuration files of WComm to your `envarc: drawer`.

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

Then the following file should be located in the `envarc: drawer`:

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

If you want to copy their icons too, you can type:

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

You have to decide now if you want to run WComm at each startup automatically or just sometimes. And if you want to start WComm from `wbench` or `cli`.

If you want to use WComm automatically after each startup you can just drop its icon to your `wbstartup drawer`.
(see also section `WBench-Start`).

`drawer`. Check in this case if the `'DONOTWAIT'` tooltype is set. Otherwise the `WBench` would wait until the program finishes.

You can alternatively insert something like:

Run >NIL: <NIL: WComm <arguments>

into your User-Startup script.

Alternatively you can insert a line into your user-startup file i.e.:

Run <path>WComm >NIL: <NIL:

WComm understands also some cli arguments, you can pass to it at startup. (See sections CLI-Start).

If you want to use WComm just sometimes you could just drop its icon into your commodities drawer.

Alternatively you can copy WComm to your c: directory.

1_> Copy WComm c:

Start:

WComm is a standard commodity. So if started the second time it will be finished.

If WComm is started it stays in the background and waits for receiving messages. In this time no CPU time will be wasted, except the time for the title clock.

WComm can be controlled with the exchange tool. If you want to open WComm's main window and you have forgotten the hotkey you can use exchange. You can also enable, disable and quit WComm with exchange.

1.6 cli-start

CLI-Start:

When started from cli WComm supports the following arguments and options:

These arguments must follow this

Syntax:

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

Config

this is name of the configuration file which should WComm load at starttime. If you do not specify Config, WComm looks first for the file 'env:WComm.prefs' and then if not found for the 'envarc:WComm.prefs' file.

If WComm cannot find the given or default configuration files, you will get a requester which will inform you about this.

The configuration file is a binary file and should not be edited manually with a texteditor. Do not try this, or you will risk a system crash.

CX_POPUP

This is a switch. If you specify it, WComm will open its main window when started. The default is: do not open it.

WPrefs

This is the name of the window configuration file. If you do not specify WPrefs, WComm looks first for the file `'env:ZWindow.prefs'` and then if not found for the `'envarc:ZWindow.prefs'` file.

If WComm cannot find the given or default configuration files, you will get a requester which will inform you about this.

The window configuration file contains titles, positions and sizes for some windows. (See also section window settings window).

This file contains only ascii characters and can be edited with a texteditor. You have to use the following format:

Each line represents one window,
the window titles must not contain any whitespaces,
AmigaDOS patterns in window titles are supported,
after each value you have to insert a whitespace,
no comments are allowed.

Syntax: <window title> <X> <Y> <W> <H>

WindowTitle: The window title with optionally amigados wildcards.

The pattern support enables WComm to recognize windows which titles will be changed at any time such as wbench windows (when you copy or delete something).

Example:

```
RAM#?  
    will match  
  
Ram Disk 100% full 0 % free ...  
RAMDISK ...  
RaMDisK ...
```

WComm will make no case sensitive difference. So RamDisK matches

RAMDISK and ramdisk etc.

X: This is the leftedge the window should be moved to.
Y: This is the topedge the window should be moved to.
W: This is the width the window should be resized to.
H: This is the height the window should be resized to.

This values can be activated with the function 'Set to prefs' (see List Of Functions). If the active window matches an entry of this list, the window will be moved to the x and y positions and resized to the w and h values.

WComm can only move windows which contains a dragbar.

WComm can only resize windows which contains a sizegadget.

You can specify different settings for each window and save this to disk.

You can also specify a supergroup here. If you have some windows with the same first chars and you want to change each one different e.g.:

for

'Shell1', 'Shell2' and 'Shell#?', you can have an entry with Shell1 and Shell2 and Shell. If you now got a window with title 'Shell4' the Shell#? entry will be used, instead Shell1 will only use the 'Shell1' entry.

EPrefs

This is the name of the exclusion configuration file.
If you do not specify EPrefs, WComm looks first for the file 'env:XWindow.prefs' and then if not found for the 'envarc:XWindow.prefs' file.

If WComm cannot found the given or default configuration files, you will get a requester which will inform you about this.

The exclude configuration file contains titles of windows which should be excluded from some functions. So can e.g. exclude the clock so its window will be ignored by the tiling and cascading functions. (See also section Exclusion Window).

This file contains only ascii characters and can be edited with a texteditor. You have to use the following format:

Syntax: <window title>

Each line represents a new window. You may use AmigaDOS wildcards. There must be no whitespaces.

CX_Priority

This is commodity priority which WComm should use. The default

priority is 0. If you have another commodity running which uses a same hotkey combination, only the commodity with the higher priority will get the message. The one with the lower priority will still stay asleep.

Do not use higher priorities than 5 and lower priorities than -5.

Notify

This is the name of the notify configuration file. If you do not specify Notify, WComm looks first for the file `'env:Notify.prefs'` and then if not found for the `'envarc:Notify.prefs'` file.

If WComm cannot found the given or default configuration files, you will get a requester which will inform you about this.

The Notify configuration file contains the notify dates and times and the commands which should be started at the given dates.

This file contains only ascii characters and can be edited with a texteditor. Because of the complicated format I will suggest you to edit this file only from the notify window.

So I will not describe the syntax of this file.

1.7 wbench-start

WBench-Start:

If you want to start WComm from the wbench, you can instead of specifying any cli argument, set some tooltypes.

These are:

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

The arguments are equal to the cli arguments. You must take care that the arguments are case sensitive.

The syntax: (do not type the `<>'s`)

```
CONFIG=<name of the configuration file>
```

```
CX_POPUP=<YES|NO> this can only be either YES or NO.
```

```
WPREFS=<name of window configuration file>
```

EPREFS=<name of exclusion file>

CX_PRIORITY=<#> # must be a number

NOTIFY=<name of the notify file>

for the system you can also use the following tooltypes

TOOLPRI=<#> this is the priority the wbench should give to WComm, this is not the commodity priority.

CLI This has no argument. This asks the wbench to open its 'command execute' window which lets you specify some CLI arguments. Then the wbench will run WComm as a cli program.

DONOTWAIT If you put WComm into your WBStartup drawer you have to specify this. This is quite equal to the Run >NIL <NIL cli command. If omitted the wbench will wait until WComm is finished.

1.8 hotkeys

HotKeys:

For each function you can specify another hotkey combination. A hotkey is a combination of a qualifier and a keycode.

Qualifier means buttons such as shift, alt, control etc...

Keycodes means all other buttons.

A hotkey may consist of only a keycode but not only of a qualifier.

Here is a list of available hotkeys:

Available Qualifiers:

lshift	: left shift
rshift	: right shift
	WComm.doc Page 4
shift	: any shift
lalt	: left alt
ralt	: right alt
alt	: any alt
lcommand	: left amiga
rcommand	: right amiga (take care for menu-shortcuts)
command	: any amiga
control	: control key
lbutton	: pressing down the left mousebutton
rbutton	: pressing down the right mousebutton
midbutton	: pressing down the middle mousebutton (if you have some)

```
diskinserted      : inserting a disk
diskremove        : removing a disk
```

Available Keycodes:

```
a,b ... z        : any aphanumeric
0,1 ... 9        : key
üäö?            : also umlauts
+-.,*/|\?...    : and other special keys
f1,f2 ... f10    : the function keys
space            : space key
backspace        : backspace key
up,down,left,right : the cursor keys
esc              : the escape key
del              : the delete key
numericpad       : note: there is a differnce between the
                  : keys on the standart keyboard and the
                  : keys on the numericpad. If you want to
                  : specify a key on the numericpad, you have
                  : to type `numericpad #' where # is the
                  : normal key. E.g. `numericpad 8` will only
                  : fit the `8` on the numericpad and not the
                  : `8` on the standart keyboard.
return          : the return key
esc             : the escape key
```

Examples:

```
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 list of functions

List Of Functions:

Here is now a list of all supported functions and their default hotkey combinations. For each new version of WComm this list will be updated.

How to use the popup window and change the hotkeys will be described later in this manual.

There are some funtions especially the cascading and tiling functions which will use a left and top offset to move the windows. The left and top offset

values depends on your system font. So I cannot tell you which value of pixels will be used, because I don't know you currently used font. I will therefor refer to these values with LeftOffset and TopOffset.

POPOP: (default: `control o`)

This opens the popup window of the WComm program. On this window you change several settings. Load a new prefs file and save the current settings, quit out WComm and so on ...
Alternatively you can use the exchange tool by clicking on the show button.

EXIT: (default: `control q`)

This tells WComm to remove immediately from memory without any ask requester. You can check this by calling the exchange program.

Close Window: (default: `control esc`)

This hotkey tries to close the current activated window by sending it an InputEvent with CLOSEWINDOW. This does only function with windows with a CloseGadget.

Enlarge Window: (default: `control f1`)

This enlarges the current activated window to the MaxWidth and MaxHeight settings of the window or to the screen borders. This does not function with windows without any SizeGadget.

Minimize Window: (default: `control f2`)

This reduces the size of a window to its MinWidth and MinHeight settings. This does not function with windows without any SizeGadget.

Reset Window: (default: `control f3`)

When WComm first tries to manipulate a window (not with tiling and cascading), it will save its current settings so you are able to reset a window to the old place and size it has before WComm has changed anything. This does only function when the window is manipulated from WComm before.

Zip Window: (default: `control f5`)

Zooms a window. This does the same than click on the ZipGadget of a window. The system zip function saves the place and size of a window before you zip it. so zipping again brings the window to place and size before the first zipping.

Reduce Width: (default: `shift alt left`)

This reduces the width of a window by the LeftOffset

Reduce Height: (default: `shift alt up`)

This reduces the height of a window by TopOffset.

Reduce Both: (default: shift alt backspace`)

This reduces the width of window by LeftOffset and the height by TopOffset.

Enlarge Width: (default: shift alt right`)

This enlarges the width of a window by LeftOffset.

Enlarge Height: (default: shift alt down`)

This enlarges the height of a window by TopOffset.

Enlarge Both: (default: shift alt space`)

This enlarges the width of a window by 24 pixels and the height by 12 pixels.

Move Left Top: (default: `lalt numericpad 7`)

This resizes the width of the window to the half width and height of the screen and moves it to the left top of the screen.

Move Right Top: (default: `lalt numericpad 9`)

This resizes the width of the window to the half width and height of the screen and moves it to the right top of the screen.

Move Left Bottom: (default: `lalt numericpad 1`)

This resizes the width of the window to the half width and height of the screen and moves it to the left bottom of the screen.

Move Right Bottom: (default: `lalt numericpad 3`)

This resizes the width of the window to the half width and height and moves it to the right bottom of the screen.

Move To Center: (default: `lalt numericpad 5`)

This resizes the width of the window to the half width and height and moves it to center of the screen.

Move Left Screen: (default: `lalt numericpad 4`)

This resizes the width of the window to the half width and height moves moves it vertically centered to the left edge of the screen.

Move Right Screen: (default: `lalt numericpad 6`)

This resizes the width of the window to the half width and height and moves it vertically centered to the right edge of the screen.

Move Top Screen: (default: `lalt numericpad 8`)

This resizes the width of the window to the half width and height and moves it horizontally centered to the top edge of the screen.

Move Down Screen: (default: ``lalt numericpad 4``)

This resizes the width of the window to the half width and height and moves it horizontally centered to the bottom edge of the screen.

Move Up: (default: ``shift control up``)

This moves the window up by TopOffset.

Move Left: (default: ``shift control left``)

This moves the window left by LeftOffset.

Move Right: (default: ``shift control right``)

This moves the window right by LeftOffset.

Move Down: (default: ``shift control down``)

This moves the window down by TopOffset.

Set To Prefs: (default: ``control f4``)

This resets the window to the settings specified in the window list. If there is no matching title this does nothing.

Set To Right Limit: (default: ``control f6``)

This enlarges the width of the window to the right screen border.

Set To Bottom Limit: (default: ``shift control f6``)

This enlarges the height of the window to the bottom screen border.

Size To Limits: (default: ``control f7``)

This enlarges the width of the window to the right screen border and the height to the bottom border.

Tiling Windows: (default: ``control f8``)

By calling this functions, WComm tries to manipulates all windows on the currently active screen in a way that there are no more overlapped areas between the windows.

E.g. if you have 4 shell windows on the workbench screen and you call this function, the windows will be sized to the half of the screenwidth and height and placed in the upper left, upper right, lower left and lower right edge of the screen. If you have instead 8 shell windows there, they will be resized to 1/8 th of the screen height and width.

This functions handles upto 16 (!) window positions. I won't say too much, but ... , try and joy it.

And ?

You will see that all odd numbers of windows will fill up first the upper rows of windows and then the lower ones. If you have more than 16 windows the others will be managed in the same way and positioned above the older.

Tiling also brings the currently resized and repositioned window to the front automatically. This is not standard and will not be used by other tiling programs.

An other aspect is, if you use a lot of windows (shells or drawers) you can wait (under OS 2.xx) a lot of time until the refresh of the windows has been finished. In this time a tiling functions normally has to wait until this refresh process has been finished, before it can go on to the next window. Imagine you have 10 shell windows (or drawers ...) what time will be wasted when using a standard Amiga. So I decided to increase the speed a little (or more). Each processing window will be analyzed. If its a SMART_REFRESH window its flags will be changed so the tiling functions have no more to wait until the window has been refreshed. The tiling process is speeded up a lot. After the process the window gets its old flags again.

Cascading Left To Right: (default: 'shift control f8')

Here all windows will be cascaded positioned from the left upper corner of the screen to right. Each window is positioned LeftOffset more to the right.

Cascading Top To Bottom: (default: 'control f9')

Here all windows will be cascaded positioned from the left upper corner of the screen to lower left. Each window is positioned TopOffset under the previous.

Cascading Hor&Vertically: (default: 'shift control f9')

Here all windows will be cascaded positioned from the left upper corner of the screen to lower right. Each window is positioned LeftOffset more to the right and TopOffset under the previous.

Cascading Bottom To Top: (default: 'control f10')

Here all windows will be cascaded positioned from the lower right corner of the screen to upper right. Each window is positioned TopOffset above the previous.

Previous Window: (default: 'lcommand left')

This activates the previous window in the screen's window list. Here only windows will be used which's titles are not specified in the window exclusion list.

Next Window: (default: 'lcommand right')

This activates the next window in the screen's window list.

Here only windows will be used which's titles are not specified in the window exclusion list.

Prev Window To Front: (default: `shift lcommand left`)

This activates the previous window in the screen's window list and brings it to the front. Here only windows will be used which's titles are not specified in the window exclusion list.

Next Window To Front: (default: `shift lcommand right`)

This activates the next window in the screen's window list and brings it to the front. Here only windows will be used which's titles are not specified in the window exclusion list.

Window To Front: (default: `lcommand return`)

This brings the currently active window to the front.

Window To Back: (default: `lcommand esc`)

This brings the currently active window to the back.

Next Screen: (default: `lcommand up`)

This cycles the frontmost screen behind the last screen.

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

This is a funtion, that gives you the ability to kill windows which are left from other programs which had been terminated by a crash etc. The currently activated window will be closed.

Do not try to close windows from applications that are currently running. In this case you will get a crash.

Open Shell (default: `shift control s`)

This just opens a newshell. The path and stack of the boot shell will be used.

Open Exclude Window (default: `shift control x`)

Opens the exclude window without popping up the main window.

Open Settings Window (default: `shift control l`)

Opens the Windowlist window without popping up the main window.

Open Clock Prefs Window (default: `shift control c`)

Opens the Clockprefs window without popping up the main window.

Open Palette Window (default: `shift control p`)

Opens the Palette window without popping up the main window.

Open Notify Window (default: `shift control n`)

Opens the Notify window without popping up the main window.

Summary:

Moving functions will only affect windows which owns a DRAG_BAR.

Sizing functions will only affect windows which owns a SIZE_GADGET.

Zip function will only affect windows which owns a ZIP_GADGET.

FRONT/BACK actions will only affect windows which are not backdrop windows like the old 1.3 WBench window.

Next/Prev, Cascading and Tiling actions will only affect windows which are not included in the window exclusion list.

The SetToPrefs function will only affect windows which are included in the window settings list.

The Reset function will only affect windows which has been manipulated by WComm before (not the cascading and tiling functions) you call this function.

Tiling and Cascading will try to move and size windows. If one window doesn't contain a SizeGadget the window will only be moved. If a window is a backdrop window, it will not be affected.

The window settings list and window exclusion list files can specify any window title by specifying the first word by amigados pattern matching.

You are also able to create `super groups`.

1.10 general

General:

Before describing the windows I will first say something general about the usage of the user interface.

For almost each gadget exists a key you can press alternatively instead of using the mouse. This key is usually displayed underscored. For some important functions this keyboard equivalent doesn't exist.

This is for security not to overwrite existing projects etc. Also conflicts with hotkeys will not appear. (Note all hotkeys are user definable).

To click on a CloseGadgets means the same than use the quit or cancel gadgets.

All functions are available at any time. This means each window is asynchronous. So you can modify a setting and you will immediately see the success.

1.11 os 2.04 gadgets

OS 2.04 Gadgets:

BUTTON_GADGET:

This button will be displayed inversiv when you click on its image with the left mousebutton. If you release the button on its imaginary the function will be started.

CHECKBOX_GADGET:

This gadget have two states. If this gadget is activated the hook is visual otherwise not.

STRING_GADGET:

This gadget enables you to easily edit a short text which is displayed in the gadget box. After pressing the return key, the editing is finished.

If you press the underlined key the gadget becomes activated and you can suddenly start to edit the text.

CYCLE_GADGET:

This kind of gadget displays one selection from a list. By clicking on this gadget the next selection will be displayed.

To press the underlined key is exactly the same than clicking on this gadget. If you press the underlined key together with the SHIFT key, the previous selection will be displayed instead.

MX_Gadget:

Here you can select also a selection. But you can see all selections at the same time. If you activate one entry the previously selected one will be automatically deselected.

SLIDER_GADGET:

The body of the gadget (mostly in black) displays the relation between the total number of elements, to the currently displayed number of elements. If you click under or above the body but in the gadgetbox, you will scroll to the next (previous) page. If you click on the body and hold down the mousebutton while moving the mousepointer, you can get a continous moving.

Also here does the underlined key together with the SHIFT key a

scroll back.

LISTVIEW_GADGET:

This gadget displays a list. Activating by clicking with thre left mouse button.

No keyboard support.

1.12 the main window

The Main Window:

If you use the hotkey for popping up (default: control o), the main window will be opened. This window will also appear if you use the exchange tool by activating the show gadget or if you specify CX_POPUP at starttime.

On this window you can edit, load or save the configuration file. You can open all other windows and edit the hotkeys.

LISTVIEW Gadget:

The ListView gadget displays a list of all available functions of WComm. You can use the arrow gadgets to scroll up and down through the list. If you click on a line, the from this function used hotkey combination will be displayed in the Current Text gadget under the ListView gadget.

Edit:

New:

If you click on the Edit gadget, the New string gadget becomes unghosted and active. After specifying another hotkey code combination, the combination will be used to replace the old one.

If the combination's syntax is correct and can be allocated from the system, the string gadget will get ghosted again and the used hotkey will be displayed in the Current text gadget.

If you instead type in an incorrect combination, the program uses its defaults and you will get the currently used one in the text and string gadget.

Under the ListView gadget there are still two more gadgets. If you have selected a list entry you can click either on Enable or Disable to enable/disable the selected hotkey. Of course you can only use disable, if the hotkey is currently active and vice versa.

The gadgets on the right handside of the window are:

OPEN:

SELECT:

SAVE:

SAVEAS:

The Open gadget lets you reload the default configuration file. The

select gadget lets you use the asl filerequester to select one. The Save gadget lets you save the currently used hotkeys under the default name `env:WComm.prefs`. The SaveAs gadget lets you use the asl filerequester to select a name to save the config file to. Default file is `envarc:WComm.prefs`.

RESTORE
DEFAULT

The Restore gadget does an undo for the currently selected hotkey, while the default, uses the default settings for all hotkeys and (re)activates them.

Open Exclude Window:

This brings up the Exclusion window, which will be described later in this manual.

Open Settings Window:

This brings up the Window Settings window, which will be described later in this manual.

Open Clock Prefs Window:

This brings up the Clock Prefs window, which will be also described later.

Open Palette Window:

This brings up the palette window which will be described later in this manual.

HIDE:
QUIT:

With Hide you can close the window while WComm stays still active, instead Quit finishes the program. If you select Quit and you have changed the settings without saving you will get a requester which lets you chose if you want to save it first. Else you got a requester if you really want to quit WComm.

The EXIT hotkey means mainly the same but if selected no confirmation requester will be opened.

1.13 the exclusion window

The Exclusion Window:

On this window you are able to change the currently used exclusion list. Also you can edit, save and load another one.

This window can be opened from the main window or via hotkey.

Windowtitles which matches the titles in this list will be excluded by some functions like

tiling, cascading, next and previous activations etc.

The list entries consists of amigados patterns with wildcards. Therefore you must take care not to insert redundant title entries. Because in this case only the first entry will be used.

The gadgets on the window are:

LISTVIEW:

The listview gadget displays a list of all excluded window titles.

If you select an entry this title is copied into the string gadget and this gadget becomes active. Now you can edit the title.

If you press return the new title will replace the old one.

NEW:

With new you can create a new entry in the exclusion list. This entry is initially called unnamed. The stringgadget becomes active and you can edit the new entry immediately.

REM:

With this button you are able to remove an entry from the current list. Take care that you should not use a name twice.

Also whitespaces are not allowed.

If you have two entries with the same name and you select Rem, the upper one will be removed.

DISABLE:

You can also disable and enable an entry. If you click on disable the currently selected entry will get an '>>' in front of the name. You can only disable entries which are currently enabled.

ENABLE:

If you click on enable the currently selected entry will be enabled again. You can only enable entries which are disabled before.

OPEN:

This lets WComm try to load the default exclusion list ('env:XWindow.prefs', 'envarc:XWindow.prefs'). If these could not be found you will get an error message.

SELECT:

This opens a filerequester which lets you select another exclusion file to load.

SAVE:

This saves the actual list to env:XWindow.prefs

SAVEAS:

This opens the filerequester which displays the second default exclusion file envarc:XWindow.prefs. Certainly you can also use another name than the default.

USE:

This closes the exclusion window.

QUIT:

This closes the exclusion window and ask you to quit the WComm program.

1.14 the window prefs window

The Window Prefs Window:

On this window you are able to change the currently used window settings list. Also you can save and load another one.

You can specify here window titles and their default positions and sizes. This settings can be used with the 'Set To Prefs' function.

This window can be opened from the main window or via hotkey.

The gadgets on the window are:

LISTVIEW:

The listview gadget displays a list of all known titles.

If you select an entry the title will be copied into the string gadget and this gadget becomes active. The leftedge, topedge, width and height settings will be copied into the integer gadgets under the string gadget. Now you can edit all values.

NEW:

With new you can create a new entry in the settings list. This entry is initially named unnamed. The string gadget becomes active and you can edit the new entry immediately.

REM:

With this button you are able to remove an entry from the current list. Take care that you should not use a name twice. Also spaces

in the titles are not allowed.
If you have two entries with the same name and you select Rem, always the upper one will be removed.

DISABLE:

You can also disable and enable an entry. If you click on disable the currently selected entry will get an '>>' in front of the name. You can only disable entries which are currently enabled.

ENABLE:

If you click on enable the currently selected entry will be enabled again. You can only enable entries which are disabled before.

LeftEdge, TopEdge, Width, Height:

These integer gadgets accepts only numbers. They display the settings of the currently selected entry. To change a value, just type in another value and press return.

OPEN:

This lets WComm try to load the default window settings list ('env:ZWindow.prefs', 'envarc:ZWindow.prefs'). If these could not be found, you will get an error message.

SELECT:

This opens a filerequester which lets you select another settings file to load.

SAVE:

This saves the actual list to env:ZWindow.prefs

SAVEAS...:

This opens the filerequester which displays the second default settings file envarc:ZWindow.prefs. Certainly you can also use another name than the default.

USE:

This closes the window.

QUIT:

This closes the window and ask you to quit the WComm program.

1.15 the clock prefs window

The Clock Prefs Window:

This window can be opened from the main window or directly via hotkey.

On this window you can control the display and the settings of the clock window and modify them or even close the clock window.

The default is to open the clock window. It displays not date and the time as hours:minutes.

Now the description of the gadgets.

Clock Type ('T' or 't')

This controls whether the clock window should be opened or not.
Default: ON.

Time Format ('f' or 'F')

This controls the format of the time display. You may select one of these:

HH:MM	- Hours:Minutes (DEFAULT)
HH:MM:SS	- Hours:Minutes:Seconds
24H:MM	- 24-Hour clock Hours:Minutes
24H:MI:SE	- 24-Hour clock Hours:Minutes:Seconds

Date Type ('y' or 'Y')

This controls the date display format. Available format types:

No Date	- Display no date (default)
WD Day-Month-Year	- Weekday Day-Month-Year
ShoWD Day-Month-Year	- Short weekday Day-Month-Year
Day-Month-Year	- No weekday Day-Month-Year

Width ('w' or 'W')

This is the width of the clock window. You may choose a value from 150 to 450. default: 150.

LeftEdge ('l' or 'L')

This is the left position of the clock window.
Default: 0.

TopEdge ('o' or 'O')

This is the upper position of the clock window.
Default: 0.

Alarm On ('n' or 'N')

The clock has the ability to inform the user at a given time with an alarm (a screen flash). If this gadget is activated, the alarm is on.
Default: OFF.

Beep Each Hour ('p' or 'P')

Also WComm may beep each hour. If this gadget is activated this function is activated.
Default:OFF.

Al Hour ('a' or 'A')

This is the hour which should be used by the alarm function.
Default: 0.

Al Min ('m' or 'M')

This is the minute which should be used by the alarm function.
Default: 0.

Use ('u' or 'U')

The current settings will be used and the window will be closed.

Cancel ('c' or 'C' or ESC)

The current settings will not be used. Instead the values before opening the clock prefs window will be used.
The window will be closed.

All settings on this window can be saved into the configuration file. Each modification will change the display of the clock just by the next displaying second. If you close the clock window by using the CloseGadget, this will not affect the currently opened clock prefs window. If you close and reopen the clock prefs window you will notice that the clock type has been changed to 'No Clock'.

1.16 the palette window

The Palette Window:

This window can be opened via gadget from the main window or directly via hotkey.

All functions on this window handles the modifications of the wbench colours. Because you can use with OS 3.0 up to 256 colours for your wbench, you cannot use older palette programs. So this had been included.

The functions will be shortly described. I think everybody knows such programs. Nevertheless here are the gadgets description:

PALETTEN Gadget (RETURN)

This selects the current colour for all modification functions.
Under OS 2.04 you see an indicator left beneath the gadget.

RED ('r' or 'R')

GREEN ('g' or 'G')
BLUE ('b' or 'B')

These slider gadgets modify the red green and blue values of the current colour. Upto now I used only the usual 16 levels for compatibility with OS 2.04. Newer versions may use all new 256 value levels.

The lower case characters will increase the specific value while the upper ones will decrease it.

Copy ('o')

If you click on this gadget, the mousepointer will be changed into a TO. This means no activation except a click on the palette gadget is now allowed. If you click on this palette gadget the mousepointer gets normal again and the older colour will be copied to the current one.

Exchange ('e')

This does quite the same, but the second colour will be exchanged with the first one.

Undo ('d')

If the Palette window will be first opened, WComm reads all colours. If you modify the colours and click then on Undo, the old values will be restored.

Reset ('r')

This restores only the current colour. This functions with each colour independent.

Use ('u')

Use all modifications and close the window.

Cancel ('c')

Use the old values and close the window.

Note: If you change the colours of your wbench screen, they will be only written to the viewport of the screen. If you start a tool like the palette wbench tool you get the old values back, before any modification. This is not a BUG.

1.17 the notify window

The Notify Window:

This window can be opened via gadget from the main window or directly via hotkey.

All functions on this window handles the modifications of the notify list. You can also load and save the notify list.

The list consists of a title, a message text, the date of the notify and an optional command which should be started.

There are also three kinds of launching such a command. EXECUTE, RUN and AREXX.

Notification requester will appear on the given date and time. Optionally the requester can start a whole commandline at this time. The requester can also be disabled and only the command will be executed. The date which should be used for the notifies can be set with specifying the hour, minute, month and day. Each of these items may be activated or not. But one item must be activated otherwise this entry will be ignored. With this concept you are also able to start a given request each hour or only one at the given date. See also the examples section.

Gadget Description:

LISTVIEW Gadget:

This gadget displays the titles of all entries of the notify list. This title will be displayed in the title of the request.

NEW:

This gadget adds a new entry to the end of the current notify list. This entry is initially empty and disabled.

REM:

This gadget removes the currently displayed entry from the list.

ENABLE:

This gadget enables the currently selected entry. You can only activate entries which are just disabled.

DISABLE:

This gadget disables the currently selected entry. You can only disable entries which are still enabled.

REQUEST:

This string gadget contains the title of the notify requester.

MESSAGE:

This string gadget contains the text of the requester. You can specify here multiple lines with an '\n'. The string can be upto 256 characters in length.

COMMAND:

If you want to start an command at the given time of the current notify, you can insert here the command. To activate the command of the notify you must activate also the action checkbox.

REQUEST ON TIME/DATE:

If this checkbox is activated, a requester will be displayed otherwise not.

HOUR and 1:

This is the hour, when the notify should be invoked. The value will only be used if the hour checkbox ('1') is activated.

MIN and 2:

This is the minute, when the notify should be invoked. The value will only be used if the minute checkbox ('2') is activated.

MON and 3:

This is the month, when the notify should be invoked. The value will only be used if the month checkbox ('3') is activated.

DAY and 4:

This is the day, when the notify should be invoked. The value will only be used if the day checkbox ('4') is activated.

NOTE: At minimum one of this date and time checkboxes must be activated, otherwise the notify will be ignored.

LAUNCH:

This cycle gadget decides which type of command should be executed. There are three available kinds:

EXECUTE

This starts simply a program or cli command and waits until it has been finished. If the command produces any output this output will be copied to a new opened shell window. Otherwise this window will still stay closed (AUTO SHELL).

BACKGROUND

This starts a program or command in the background. All in and outputs will be sent to NIL:.

AREXX

This will invoke RX with the given arexx script.

Note: All types of commands will use the path and stack which had been set in the startup-seuqence.

ONCE:

If this checkbox is activated, the notify will be displayed only once. You are able to specify notifies which can popup each hour. If you do not want this, you can specify Once and the notify will be invoked only once. If Wcomm is starting the next time the request will be displayed again only once (See examples).

ACTION:

If this checkbox is activated, the notify invokes the given commandline otherwise not.

OPEN

This opens the default notify configuration file which is called 'env:notify.prefs'. If this is not found the 'envarc:notify.prefs' will be tried to load. If this cannot be loaded nothing happens.

SELECT:

This brings up the filerequester which lets you chose a new notify configuration file.

SAVE:

This saves the current notify list under the default names.

SAVE AS:

This bring up the filerequester to chose a name the notify list should be saved to.

USE:

This uses the notify setting and closes the window.

QUIT:

This ask the user to quit WComm and closes the window.

EXAMPLES:

=====

Because its a little morre difficult, so I will give you here some (less) useful examples.

1. You want that WComm notifies you at christmas. Only a requester should appear.:

Request: !!! Merry Christmas !!!

Message: I wish you a merry christmas|an hope you will get a lot|
of wonderful presents

Mon: 12 3: activated

Day: 24 4: activated

Once: activated

Here you will see the usage of the Once gadget. If not activated, you will get the requester each minute. So you will get the requester each 24-Dec each year, but only once on these days.

2. You want to get a message each first quarter of an hour to notify you its time for a break.

Request: !!! Break Time !!!

Message: Okay Boy |Its now the time for a cup of coffe| or do you prefer tea ?

Min: 15 2: activated

3. You want to start a backup of your drive dh0: automatically each month on the 3rd day of the month. You will first receive a message to stop all other computer activities. Assuming you have the program backup into your current dos searchpath.

Request: !!! Backup Time !!!

Message: Okay Boy !! No it is the time to start a| new backup

Command: Backup DH0:

Day: 3 4: activated

Once: activated

action: activated

You can also put in all dates of the birthdays of all friends and you will never forget these dates anymore. Set the date one day before their birthday and you will not forget to buy a present.

1.18 the clock window

The Clock Window:

All settings of the display of this clock can be modified by using the Clock Prefs window.

The Clock can be closed by pressing the CloseGadget. To reopen it again you must use the Clock Prefs window.

You cannot move the Clock with the mouse. If you want to change the position of the clock, you must use the Clock Prefs window too.

The display of the clock will be refreshes only 1 times per second. Between two refreshes the clock stays in the background and doesn't waste any CPU time.

1.19 error handling

Error Handling:

WComm has been created under OS 3.0. I used Enforcer and hope that there are no more enforcer hits left. But also with best control some bugs may be left. So I support some error handling functions.

Most errors should occur by errors of the usage. Also wrong configuration files and formats or other not so system conform programs may lead WComm to an error.

IN GENERAL:

For each error WComm tries to inform the user about the reason. By some errors the user will be only informed by some more serious errors the user gets the decision to break WComm or to proceed. If any very fatal error occurs WComm have to terminate. But in all cases the user should get full informations about any reason.

SOFT ERRORS:

With soft errors the user will be only informed. After answering the requester, WComm proceeds.

SERIOUS ERRORS:

With more serious errors the user can decide if he wish to proceed or to stop WComm.

I will suggest you: always finish WComm or you will get the risk that WComm cannot function well.

FATAL ERRORS:

With fatal errors WComm is not able to proceed. So when the user answers the requester, WComm have to terminate.

UNKNOWN ERRORS:

If any unknown error occurs, you should check whether WComm is correctly installed and if you have also the following libraries in your libs drawer:

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

If you cannot find the error, you should first use the manual to look for further details. If you cannot find the answer, you can write me a letter. I will do my best to answer all questions as good as possible. But do not forget to tell me correctly the error, all running background processes, your type of computer, your OS version, your expansions, memory and the version of WComm.

So it will be much easier to find the reason.

1.20 technical

Technical:

WComm is a standard commodity. It has all abilities and functions a commodity should have. Exchange gives you full control about the behaviour of WComm.

WComm was written and created with the help of the following programs:

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

My best thanks to the programmers of these very good programs.

My best thanks also to Mark Weisheit for giving me suggestions and help me to remove some errors.

Instead of using the delivered standard startup routines from Aztec I used other ones that do not use the old 1.3 features which has been replaced with new OS 2.04 (37++) ones. This saves a lot of code size. Although WComm is got relatively big. The size of all used structures is only dependent to the free memory. All structures will be handled dynamically.

1.21 list of files

List Of Files:

```
WComm           : english binary
WComm.info      :
WComm_G         : german binary
WComm_G.info    :
WComm.doc       : english documentation
WComm.doc.info  :
WComm.dok       : german documentation
WComm.dok.info  :
```

```
WComm.guide      : english AmigaGuide ® file
WComm.guide.info :
WComm_G.guide    : german AmigaGuide ® file
WComm_G.guide.info :
History.log      : history log file
History.log.info :
WComm.prefs     : example configuration file
ZWindow.prefs   : example window list file
XWindow.prefs   : example exclusion list
Notify.prefs    : example notify file
WComm.prefs.info :
ZWindow.prefs.info :
XWindow.prefs.info :
Notify.prefs.info :
```

1.22 address

Address:

For all mail and messages, bugs, gifts, and our critics ...

Hans-Peter Guenther
Gruener Weg 5
W-3501 Fuldata 2
GERMANY

your H.P.G