KCommodity

COLLABORATORS

	TITLE : KCommodity		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		November 24, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

KCommodity

1.1 KCommodity

Welcome to KCommodity v2.5. So you finally made it into the documentation. I hope you like this program as much as I do and many other users. For any requests and suggestions just contact me under the address given in that document.

Within this manual I will refer to the original program texts, that means not to the localized version of KCommodity.

INSTALLER SOFTWARE IS PROVIDED "AS-IS" AND SUBJECT TO CHANGE;
NO WARRANTIES ARE MADE. ALL USE IS AT YOUR OWN RISK. NO LIABILITY
OR RESPONSIBILITY IS ASSUMED.

--- The programs ---

--- Advanced features ---

1.2 KCommodity/Overview

Overview
****

1.3 KCommodity/Copyright

Copyright
=========

You are allowed to copy and spread this program but only if you do not make any commercial profit out of the program itself and all supported files. The source of KCommodity is considered teachware and the author claims the right to be the only one to produce and distribute new versions of KCommodity. Public Domain vendors are not allowed to sell KCommodity for more than the price of a regular disk, which should not be higher than 5 Dollars.

If you do not have the sourcecode yet, you may send me a disk (including appropriate stamps/money for postage and packing). I will copy the sourcecode on that disk and return it to you.

1.4 KCommodity/Guarantee

Guarantee
=========

The author does not give any guarantee that the program KCommodity works perfectly. The program was tested and it is used every day, anyway nobody is perfect and that is why I can not take the responsibilty for any damages occured during the usage of KCommodity.

1.5 KCommodity/Evolution

```
Evolution
=========
```

KCommodity was developed out of the need to reduce the amount of smaller programs, which should ease the daily work with the computer. Instead I wanted to create an effecient program, which should implement the functions of these utilities and that my program might go one step further. At first KCommodity only consisted of a window activator, which gave me the possibility to activate windows via keystroke or via mouse movement. Growing knowledge of the new Operating System and the eagerness to minimize the number of running utilities there were more functions to be implemented into KCommodity. So KCommodity became bigger and bigger but it‘s size, in comparison to it‘s functions, was held small.

Now I‘ve come to a point that KCommodity almost completely suits my needs and that it replaces other programs of it‘s kind. I hope that you like the supported functions as much as I do and I wait for any suggestions to implement new, powerful functions.

1.6 KCommodity/Parts

```
Introduction on KCommodity v2.5
===============================
```

KCommodity v2.5 comes as three files (well, there are 4 files, but the last one is a shared library, which can not be launched by the user). The first program KCommodity (per default located within ‘SYS:WBStartUp/’) is the main part. This program controls all the functions available.

The second program is the Preferences editor also called KCommodity, which is located within ‘SYS:Prefs/’. This program can be launched without having to run the main KCommodity first. It is the interface to KCommodity for setting all the preferences you like. If you start the Preferences editor, you will not see any changes made, while KCommodity itself is not running. All you can do is to change your preferences.

The third program is called ‘KCXPrinter’ and is normaly located in ‘SYS:Utilities/’. This program is KCommodity‘s printer driver for dumping screens/windows to a suited printer or to an IFF-ILBM file. It can only be launched, when the main program is running. If not the printer driver will tell you and the program will terminate.

1.7 KCommodity/Ups

Notes for updating users
========================

The preferences file changed again for that release. Please use the supplied program ‘PrefsConvert’ in order to change existing preferences to suit the new preferences format. Here is an example how to call the program.

```
                   PrefsConvert ENVARC:KCX/KCX.PRFES
```

This would convert the existing preferences file within ‘ENVARC:KCX/’ for the new format.

1.8 KCommodity/UpDates

How to obtain updates
=====================

KCommodity will be available on ADS, FRAS, FTP, and Fred Fish disks. The versions found on these sources are not restricted, these are complete and fully working versions of KCommodity. If you already registered yourself for an older version, please send in some money, covering copying costs and postage and packing. I am not a millionaire to be able to do updates myself. I will announce any new versions on the corresponding networks (Fido, ZNet, Usenet).

To register please fill out the ‘Register.Txt’ form supplied on the distribution disk and send it to the address given in that document. Previous users will find out that KCommodity got somehow more expensive than older versions. This is due to copy costs and postage and packing. On many regsitrations I was forced to pay some more than the DM 20...

1.9 KCommodity/Address

How to contact the author
=========================

If you have any questions, suggestions, bug reports or anything else you want to tell me, you can contact me under one of the addresses.

Either use normal snail mail:

Kai Iske
Brucknerstrasse 18

```
     6450 Hanau 1
     Germany
     Tel.: +49-(0)6181-850181

   or reach me using electronical mail

     USENET:   kai@iske.adsp.sub.org
     FIDO:     Kai Iske, 2:249/40.11
     ZNET:     SNAKE%HTH@AMNET.ZER
```

1.10 KCommodity/Thanks

```
Whom I must say 'thank you'
===========================

   I would like to thank the following people

`Dirk Federlein'
     For betatesting

`Michael Herbst'
     For betatesting

`Martin Berndt'
     For betatesting and some bytes that were saved, and for the
     TagScreens source

`Olaf Barthel'
     Who answered some questions and helped me with some problems

`Frank Mariak'
     For betatesting and the suggestions for the efficient localization

`Christoph Reichert'
     My FIDO-Boss for the support

`My girlfriend'
     Who accepts what I`ve done/doing

`The translators'
     Benoit Mortier, Ruud van der Veen, Terje Bergstrom, Michael
     Korsback, Juan Gomez, David Santamaria, Wolfgang Paul and Franz
     Hemmer.

   ...and all the others I forgot
```

1.11 KCommodity/Installation

```
Installing KCommodity
*********************
```

Here some topics about the installation procedure

1.12 KCommodity/CBMInstall

```
Installing with Installer
=========================
```

First, simply double-click on the ‘Install-KCommdodity’ icon. You will have to place Commodore‘s Installer in the ‘C:’ directory. There is not too much to say about how to install KCommodity using Installer. Simply follow the steps. Help is available almost everywhere.

1.13 KCommodity/ToolTypes

```
Using ToolTypes
===============
```

KCommodity‘s main program knows some ToolTypes which can either be located within the .info file or may be issued on the command line. The syntax should be of the form ToolType=Value. If you start the program from a command line you can leave out the ‘=’ and replace it by a space.

Here are the ToolTypes the program recognizes.

‘CX_PRIORITY=n’

Sets the priority for the program‘s section which controls the HotKeys. If you issue a priority > 0 it may be that HotKeys initialized by other programs may not be reached anymore if KCommodity uses the same. Default: It‘s neutral 0.

‘CX_POPKEY=s’

Lets you define a new HotKey for opening KCommodity‘s main window. The Value for this ToolType will be overridden by the definition loaded from the Preferences file. Default: LCOMMAND HELP.

‘CX_POPUP=YES|NO’

You can choose whether the main window should be opened on startup or not. Issuing YES will open the window whilst NO will keep it away. Default: YES.

‘PORTNAME=s’

You can select a different name for the program‘s ARexx-Port. Please remember that this name‘s length should not exceed 16 chars. Normally this will be KComm.1.

'TOOLPRI=n'

This one is used to set the priority for the program itself. This may be usefull while running other programs of KCommodity's kind, since they might conflict. Just set the priority to a slightly higher value and all the problems should be gone. Default: Again the neutral 0.

'HANDLERPRI=n'

This ToolTypes lets you define the priority for the InputHandler. This Handler is used to control KCommodity's functions like window activator etc.pp. If you use other programs like KCommodity it may happen that these conflict. Just supply a higher priority and problems should be done. Remember that the minimum priority for such a Handler is 51. Default: 55.

'SETTINGS=s'

You can have a different Preferences file to be loaded on startup. If KCommodity is not successful in finding the specified file it will try to open the default 'KCX.PREFS' within the current directory. If this attempt also fails it will look for the file within 'ENV:KCX/'. If that one could not be found, too, KCommodity will use it's built-in defaults.

'PREFSPATH=s'

Per default, the prefs programm of KCommodity will be found in 'SYS:Prefs/'. If you prefer a different location for that program, you have to tell KCommodity to look for it in that place. This is done using that ToolType. You have to supply the full path including filename.

'PRINTERPATH=s'

Normally, KCommodity will look for the printer driver within 'SYS:Utilities/'. You can override the default path, using this ToolType. You have to supply the complete pathname and filename.

'NOICON'

This simple ToolType tells KCommodity not to generate the AppIcon on the Workbench. See AppIcon and AppMenuItem, for more information on AppIcons and AppMenuItems.

'NOMENU'

If do not want to have the AppMenuItem, too, use this ToolType. (see AppIcon and AppMenuItem).

'ICONX=n'

Using this ToolType you may specify the AppIcon's X-Position. Default setting is that the Workbench should find a suitable location.

'ICONY=n'

Same as above, with only one exeception. This ToolType controls the AppIcon's Y-Position on the Workbench Screen.

'NOREQ'

Per default, KCommodity will first ask you, whether to quit.

Setting this ToolType will cause the program not to display the requester.

For the prefs program there is a ToolType, too. So here it is:

'SETTINGS=s'
This one has the same function as the main program's *SETTINGS* ToolType.

1.14 KCommodity/ShellParm

Using Shell parameters
======================

You can supply parameters to KCommodity's main program when starting from a CLI / SHELL. The name and function of each parameter is the same as already described in the previous section (see Using ToolTypes).

For the prefs program there is a parameter, too. See Using ToolTypes, for a complete list of available ToolTypes.

1.15 KCommodity/AppStuff

AppIcon and AppMenuItem
=======================

If not chosen differently, when started, KCommodity will display an AppIcon on the Workbench screen. This icon does not have anything to do with *normal* Worbench icons. These stand for programs which can be started by a double-click on the corresponding icon which is defined within the program's .info file. An AppIcon is a so called Applications Icon. This type of icon is generated by programs from which the user may communicate with the program. For KCommodity the AppIcon only has one function: A double-click on the icon will open up the main window.

Furthermore KCommodity will generate an AppMenuItem which is appended to the *Tools* menu of the Workbench. Each program can add one (or more) entries to that menu. The method of selecting one of these entries is the same as for "normal" menus. KCommodity will add a menu entry called *KCX Preferences* from which you can open the program's main window.

1.16 KCommodity/StartKCX

Starting KCommodity
****

KCommodity can be started either from Workbench or from CLI / SHELL.

NOTE to users of MagicMenu. Please make sure that you launch Magic Menu, before KCommodity! While starting it from ‘WBStartUp’ you should set the STARTPRI ToolType to a somewhat higher value than that of MagicMenu. This is because MagicMenu also patches the OpenScreen functions, which would conflict with KCommodity‘s window-reactivation and screen cycling.

Starting from Workbench	Starting the program from Workbench
Starting from CLI / SHELL	Starting the program from a CLI / SHELL

1.17 KCommodity/StartWB

Starting from Workbench
=======================

If you start KCommodity from Workbench, you may enter some preferences into the .info file of the program, which will be loaded on startup. Within the .info file there should be another ToolType set except for these discussed before (see Using ToolTypes). This ToolType will prevent the Workbench from waiting until KCommodity stops, if the program was launched during boot time (assuming that the program is located within the ‘SYS:WBStartUp’ drawer). This ToolType is called DONOTWAIT.

1.18 KCommodity/StartShell

Starting from a CLI / SHELL
===========================

Of course you can start KCommodity from your CLI / SHELL, but you have to know that KCommodity does not detach itself from the CLI / SHELL so you will have to use RUN / RUNBACK(1) to prevent KCommodity from blocking the environment of your CLI / SHELL. When you start KCommodity you can supply the same ToolTypes within the commandline as described earlier (see Using ToolTypes). Instead of using one form for supplying parameters within the .info file you have two on the commandline. The first is to have a ‘=’ between the keyword and the value and the second is to have a space between them. Remember that parameters which include spaces should be enclosed in brackets.

e.g. "CX_POPKEY=LALT HELP"

or

CX_POPKEY "LALT HELP"

Additionally you have the possibility to retrieve a little Helptext.

The syntax is as follows:

```
KCommodity Help
```

This command will print out a little table showing all available ToolTypes. Starting KCommodity like this

```
KCommodity ?
```

will show you a list of all available ToolTypes giving you the possibility to type in the requested keywords and start the program.

---------- Footnotes ----------

(1) The latest version of RunBack can be found on Fred Fish Disk 429

1.19 KCommodity/MainPrg

The Main Program
****

Well, there is not too much to say about the main program. When launched, it will simply do nothing, except for watching/performing all the actions specified within the Preferences editor (see The Preferences Program, for more information about setting the prefs).

On demand KCommodity may also launch the Preferences editor either via HotKey, AppIcon or via the AppMenuItem (see AppIcon and AppMenuItem, for more about that).

Additionally you can launch the Graphics dump by using a HotKey.

There are at least one or two functions/windows which may be opened by the main program, which are built into this part of KCommodity.

NOTE to users of MagicMenu. Please make sure that you launch Magic Menu, before KCommodity! While starting it from ‘WBStartUp’ you should set the STARTPRI ToolType to a somewhat higher value than that of MagicMenu. This is because MagicMenu also patches the OpenScreen functions, which would conflict with KCommodity‘s window-reactivation and screen cycling.

Here is a list of functions KCommodity offers per default, which can not be altered:

- Mapping left Amiga N/M In addition to window-reactivation the keys left Amiga in conjunction with M or N are treated somewhat special, in order to allow complete window reactivation.

- WildStar KCommodity will automatically set the WildStar bit. This means, that you may use the '*' within the Shell as a real Wildcard. KCommodity will go one step further by giving you the possibility to use the '*' Wildcard alone as a joker for "all-files", which can not be done by simply setting a flag.

The windows of KCommodity`s main program are:

Formatter
Bill window
Select screens

1.20 KCommodity/FormatWin

Formatter
=========

KCommodity has the ability to check every disk inserted whether the disk is formatted or not. You can tell KCommodity to ask whether the disk just inserted should be formatted. This is done within the Formatter window. KCommodity will recognize all drives connected to your computer and you can tell the program which disk drives to watch, that means which drives should be checked for unformatted disks. So if you insert an unformatted disk into a drive which is checked by KCommodity, the program will pop up a requester asking whether you wish to format the disk or not. If you do want to format, this window will pop up.

Alternatively it can also be opened by using a HotKey (see Formatter) or by using the gadget within the Preferences editor (see The Preferences Program).

NOTE: Due to the different filesystem, KCommodity will not recognize MS-DOS disks inserted into a PCx: drive.

Here are the gadgets for this window. Some of them are only available to the Formatter by Olaf 'Olsen' Barthel (1). These gadgets will be marked as those only applying to this program. Additionally two of the gadgets are only available, when running under OS 3.0. These gadgets are only accepted by the original Commodore formatter.

NOTE: If you supply a flag to a formatter it does not understand, you will not be able to format any disk!

If you want to make use of KCommodity`s formatting facility, you have to copy the formatter program to `SYS:System'.

Drive
Label
Trashcan
Quick
FFS
Directory-Cache
Intern. Mode
No Verify
Install
Eject

```
  Watch disks
  DF0 - DF3
  Format
  Cancel

   ---------- Footnotes ----------

   (1)  Which can be found on Fish Disk 535
```

1.21 KCommodity/Drive

```
Drive
-----

   This gadget reflects the drive name selected for the format
operation. If the window was opened interactively (say, by KCommodity
after the requester), the program will automatically select the drive
the unformatted disk was inserted in.
```

1.22 KCommodity/Label

```
Label
-----

   This one`s simple. This gadget holds the new diskname.
```

1.23 KCommodity/Trashcan

```
Trashcan
--------

   This gadget tells the format program to generate the Trashcan on the
disk formatted. This option is available to all of the formatters.
```

1.24 KCommodity/Quick

```
Quick
-----

   If you want to format an already formatted disk, you may use this
switch to just clear the disk. This option may be used with all
formatters.
```

1.25 KCommodity/FFS

FFS

Beginning with OS 2.0, floppy disks can be formatted using the FastFileSystem. This flag tells the formatter to use this filing system to format the disk. It can be used with all of the programs.

1.26 KCommodity/DirCache

Directory-Cache

Starting with OS 3.0, there was a new option for the filesystem called Directory Caching. This will dramatically increase directory load times, but it can only be used with OS 3.0 and up and that means it can only be used with the OS 3.0 formatter command.

1.27 KCommodity/InterMode

Intern. Mode

This option is available only under OS 3.0 and up (and for that formatter program, of course). It is usefull for transfering disks between different localized systems that means AmigaDOS will correctly recognize german Umlauts within filenames, so that you do not have to write them uppercase, if they are uppercase within the name.

1.28 KCommodity/NoVerify

No Verify

Using this option you may turn off the verify option of Olaf Barthel‘s formatter. As you can beleive this option only works with this formatter. Turning off verify will reduce format time, but you can never be sure, whether the disk has been formatted correctly.

1.29 KCommodity/Install

```
Install
-------
```

In order to boot from a disk it has to be installed first. Olaf Barthel‘s formatter is able to install a disk, after formatting it. So simply turn on this gadget to have a bootable disk after formatting. As said, this option may only be used with Olaf‘s formatter.

1.30 KCommodity/Eject

```
Eject
-----
```

Some disk drives understand the eject command, which causes the disk drive to automatically eject the disk. Olaf Barthel‘s formatter supports this flag, in order to minimize user "support". This flag may only be used with his formatter.

1.31 KCommodity/WatchDisks

```
Watch disks
-----------
```

Turning on this gadget will cause KCommodity to watch all selected disks (see DF0: - DF3:). If this gadget is turned off, KCommodity will not bother you with any questions concerning formatting a disk.

1.32 KCommodity/Drives03

```
DF0: - DF3:
-----------
```

Maybe not all of these gadgets will be available (unghosted). Only those gadgets are available, which apply to a drive connected to your system. If one of these gadgets is turned on, KCommodity will check every disk inserted into that drive (see Watch disks).

1.33 KCommodity/Format

```
Format
------
```

Clicking onto this gadget will start the format operation. A small console window opens up and the formatter will be loaded.

1.34 KCommodity/FormCancel

Cancel

Simple, cancel the format.

1.35 KCommodity/BillWin

Bill window
===========

This window may either be opened by the Preferences editor (see The Preferences Program) or by hitting a HotKey (see HotKey window).

This window gives you information about the actual / past phone call. The actual system time *Time*, the online time *Online*, and the costs arising / arisen from a call *Cost*, will be shown. Underneith there are five gadgets you can select the time zone and tariff with. If you change the settings during a phone call KCommodity will automatically recalculate the costs so that the program *always* works with the correct values and stores them to the log file. Underneith these gadgets there is another display box showing the costs for one hour for the given time zone and tariff. At the bottom border of the window there is another gadget called *Log call*. If it is activated data about the current phone call will be written to the log file after the modem hung up. If the gadget is deselected this data will not be stored to the file. This data will be used by * Display total sum* (see Bill preferences, for more about this) to calculate the total costs for all of your phone calls. This gadget might be usefull to filter out some calls, say if you have been called by another person.

This window is equipped with a Zoom Gadget. Using this gadget it is possible to easily switch between two sizes for the window, so that it does not use too much space on the Workbench (or the Public Screen). The window‘s position (normal sized or "zoomed") will be written to the Preferences file. If the window is zoomed (small), the window‘s title bar will show the online time, costs, timezone, tariff, and whether the call will be logged or not, so that you always have an overview of data. You can also read, whether data about the call should be saved.

When looking at the window you might recognize two more gadgets labeled *Start* and *Stop*. These may be used to manually start the cost calculating, say while you are making a voice call. While you are in manual mode KCommodity will ignore any Carrier Detects and losses, that means you may do transmissions and then continue to talk voice. On the other side, you can not stop cost calculation by using *Stop* if the calculation was started due to a Carrier Detect.

1.36 KCommodity/SelScrWin

Select screen window
====================

This window will *only* be opened when you use it‘s HotKey. See Select screen. The default definition for this HotKey is LSHIFT ESC.

Imagine you have several screens open and wish to get to a specific screen quickly. Instead of flipping through all the screens manually you may open the *Select screen* window. Within this window you will find a list of all available screens currently known to the system.

If you try to open the window, KCommodity will first try to determine the active screen‘s type. If it is a Public screen, the window will pop up on this one. If not, the default Public screen will be used and brought to front. If you cancel the operation the non-public screen will be brought back to front.

Furthermore you will see the number of screens and the name of the default public screen.

The gadgets within this window are as follows:

List
Update list
Activate
Cancel

1.37 KCommodity/SelList

List

Within this listview gadget you will see a list of all screens currently open. Public screens will come first, their names will be preceded by a *(PUB)* string to indicate their type. If KCommodidty can not determine the name of a screen there will be an entry of *? NO NAME ?*.

1.38 KCommodity/SelUpdate

Update list

If you are not sure whether screens have been removed/added to the system, you may click onto this gadget to tell KCommodity to rescan the screen list. The new list will then be shown within the listview gadget (see List).

1.39 KCommodity/SelAct

Activate

This gadget will bring the selected screen to front.

WARNING: If you try to bring a screen to front which is no longer opened, the system will crash (I will secure that probably).

1.40 KCommodity/SelCancel

Cancel

This gadget will simply cancel the operation. If the window was not opened upon a Public screen, the previous screen will be brought back to front.

1.41 KCommodity/PrefsPrg

The Preferences Program
****

Assuming, that you used the default installation for installing KCommodity on your HardDisk/BootDisk I will now explain, how to use the Preferences program to set up KCommodity to your needs. This program may be started either by KCommodity or by simply double-clicking onto the icon. Per default the program is located within 'SYS:Prefs/'. This program is the main interface to KCommodity, as it controls all the settings.

Within the windows you will find gadgets (how come ?). All of the gadgets may alternatively controlled via keyboard. To find out which key to use, simply take a look at the gadget`s text. There is a character underlined, which is the shortcut to use.

The lower 6 gadgets are used to control KCommodity. These are:

- Load Prefs Having selected this gadget, a FileRequester will open up asking for a file to load new settings from, which should be used by KCommodity from now on. Using this feature you may store several settings to your (Hard)Disk. Have a look at section *Save Prefs* to see which data is written to that file.

- Save Prefs As mentioned under *Load Prefs* a FileRequester will open up. You can now specify a file the current settings should be written to. The following data is written to that file:

    * All settings of all windows

    * The position / size of all windows

- Remove This gadget will tell KCommodity (the main part) to remove itself. This gadget is unavailable if KCommodity does not reside in memory.

- Enable Clicking onto this gadget will invert the status after *Disable*. All of the program‘s functions will be reenabled.

- Disable Selecting this gadget will turn off almost all of the program‘s functions. Touched are the following features:

    * MMB-Shift

    * Map "Umlauts"

    * MenuWrap

    * String2Menu

    * KeyStroke clicker

    * ESC Window Close

    * Full dragging

    * Enter ASCII

    * Screen / mouse blanker

    * The screen / window cycling

    * The window activator

    * The mouse accelerator

    * LeftyMouse

    * Mouse locking

    * The function keys

    * *All* HotKeys

```
     Please remember that KCommodity`s windows can not be opened via
     HotKeys anymore. You will have to use either the AppIcon,
     AppMenuItem or use the Exchange program to open the program`s main
     window and from there you can open the other windows. Of course
     you can reenable KCommodity using the Exchange program.

   - Quit Will quit the Preferences Editor

   The main window has a menu attached.

   * Save settings...  Will save the current settings under the name
     last used, without opening a FileRequester.

   * About...  Will open the About-Requester.

   The upper 15 gadgets lead to windows which will be described now:


  Settings
  Miscellaneous
  Clock definitions
  Blanking
  Cycle settings
  Tag Screens
  Mouse
  HotKeys
  Graphics dump
  Function keys
  Lock window
  Formatter
  Bill preferences
  Alarm modes
  Bill

--- Windows, no gadgets lead to ---

  Qualifier window
  Task window
```

1.42 KCommodity/Settings

```
The settings window
===================

   This window only controls simple functions, which do not require any
further input. You can just turn on/off some features of KCommodity
here.

   Here are the features you can control through the Settings window


  MMB-Shift
  Write environment
  Map Umlauts
  Drive NoClick
  WB to front
  MenuWrap
  No DisplayBeep
  Avoid Flicker
  String2Menu
```

1.43 KCommodity/MMB-Shift

MMB-Shift

Is abbreviation for "Middle-Mouse-Button Shift". This feature aids you in multiselecting icons on the Workbench. While you are multiselecting there is no need to press the SHIFT key anymore, simply hit your middle mousebutton.

1.44 KCommodity/WriteEnv

Write environment

If you want to offer the current system time to another application, which does not supply a time display, but is able to read environment variables then you can tell KCommodity to generate environment variables for you, which contain the actual system time and date.

This gadget de- / activates the creation of environment variables for the actual system date. Variables will be generated for the day, date, and time. These variables take the names DAY, DATE, and TIME.

1.45 KCommodity/MapUmlauts

Map "Umlauts"

With a selected gadget all Umlauts will be automatically mapped to their pendants (ae, oe, ue). Of course this function works case-sensitive, so that KCommodity will generate Ae, Oe, and Ue if one of the keys was hit in conjunction with SHIFT. The ‘SZ’ will be mapped, too, so that there may not be any problems anymore in writing German texts not containing German Umlauts.

1.46 KCommodity/NoClick

```
Drive NoClick
-------------

   Using this gadget you may turn on / off the noisy drive-clicking.
```

1.47 KCommodity/WBFront

```
WB to front
-----------

   If this gadget is set, the Workbench screen will be brought to front
if you try to open KCommodity`s main window.
```

1.48 KCommodity/MenuWrap

```
MenuWrap
--------

   This gadget controls the MenuWrap feature of KCommodity. If the
gadget is set, and you try to access a window`s menu, the mouse pointer
will immediately jump into the menubar. When you are moving through the
menu and hit one of the menu`s horizontal borders, the mouse will
automatically jump to the other side of the menu, keeping it within the
menu boundaries. This function does not affect the vertical movement of
the mouse.

   *NOTE:* MagicMenu does something strange when doing it`s menu
thingies. I was not able to get this function to work with MagicMenu.
```

1.49 KCommodity/NoDisplayBeep

```
No DisplayBeep
--------------

   This gadget toggles the Intuition function DisplayBeep(). When
turned on, KCommodity will prevent any function from calling
DisplayBeep(), in order to turn off any annoying sounds or flashes.
```

1.50 KCommodity/AvoidFlicker

Avoid Flicker

This gadget may be of use for users of a Commodore FlickerFixer or an Amiga 3000. This FlickerFixer creates a flickering first line on the monitor. Using this feature you are able to turn off this flickering. One note to users of this function: You can not move a screen to vertical position 0 anymore. Otherwise the screen would be completely black.

1.51 KCommodity/String2Menu

String2Menu

Previous users of OS 1.3 may still remember the ability to reach menu shortcuts from within String Gadgets of a e.g. CygnusED window (like the Search Requester). Under OS 2.0 this has changed. There was no way to use this nice feature anymore. This changes, when you turn on String2Menu and back are the good old times. From any String Gadget in any window you may now reach a window‘s menu via it‘s shortcuts.

1.52 KCommodity/Misc

Miscellaneous window
====================

Within this window you will find settings which require more specific inputs.

Key-Click
Click volume
ESC WindowClose
ESC Qualifier
Full drag
Dragging qualifier
User-Shell command
Enter ASCII
ASCII Qualifier
Exploding windows
From/To mouse
Explode frames
Implode frames
Size frames
Move frames

1.53 KCommodity/KeyClick

Key-Click

You probably know this feature from professional keyboards. These keyboards produce a clicking sound when a key is hit. This feature is simulated by KCommodity. You can set the volume for producing the sound to have an acceptable simulation.

This gadget is used to de- / activate this feature.

1.54 KCommodity/ClickVol

Click volume

Using this gadget you can alter the volume for the clicking sound of Key-Click. Values between 1 and 64 are accepted. Remember that 1 is the lowest volume and 64 is the highest.

1.55 KCommodity/ESCWindowClose

ESC WindowClose

This gadget de- / activates the emulation of the beloved function of good PC programs. A hit on the ESC key in conjunction with a qualifier defined by ESC Qualifier will close the window underneith the mouse. (see ESC Qualifier, for more information) Prerequisits for this function are that the window does not show a requester and that a Close Gadget is tied to the window.

1.56 KCommodity/ESCCloseQual

ESC Qualifier

To the right of this gadget you find the shortcut definition which activates the ESC Window Close function. (see ESC WindowClose, for information about that feature) If you want to change the shortcut, simply hit on that gadget. A window will pop up giving you the possibility to change the key-combination. Simply hit the keys you want to use in conjunction with ESCAPE. If you click on OK the definition will be in use. Please remember that you have to click underneith the Text Gadget, otherwise the left mousebutton will not be recognized.

1.57 KCommodity/FullDrag

Full drag

Using KCommodity there is no need to click on a window‘s titlebar in order to move that window. Just click inside the window and perform the movement as if you clicked on the titlebar.

This gadget controls the Full-Drag function. The qualifier defined under Dragging Qualifier will not be checked if this gadget is not selected. See Dragging qualifier, for more information about defining it‘s key.

1.58 KCommodity/DragQual

Dragging qualifier

As with ESC Qualifier you can define the qualifier for the Full Drag function (see Full drag). If you hit this qualifier in conjunction with the left mousebutton you can move a window without having to click onto the window‘s titlebar.

1.59 KCommodity/UserShell

User-Shell command

You can enter a command string for a Shell command here. This command will be executed, if you hit the appropriate HotKey. Just enter a command with it‘s parameter here as if typed into a CLI / SHELL. Here is a small example:

```
                     NewSHELL CON:0/11//200/Shell
```

would start a new Shell process. It‘s window would be 200 pixels high and it would be positioned underneith the Workbench‘s titlebar.

Remember that you enclose the ’CON:...’ string in quotes, if it contains any blanks. Just like you would do it from a CLI / SHELL.

The Shell will be launched when you hit the appropriate HotKey. The default definition for this HotKey is ALT SHIFT S. See UserShell, for more information.

1.60 KCommodity/EnterASCII

Enter ASCII

This is a nice feature for users, who can remember ASCII codes of all characters available on the Amiga. Just like on PCs you can enter a character, by entering it‘s ASCII code on the numeric keypad, while holding down the LEFT ALT key. In KCommodity you can change the qualifier for use with Enter ASCII. This is done with the ASCII Qualifier gadget. (see ASCII Qualifier).

1.61 KCommodity/ASCIIQual

ASCII Qualifier

This gadget is used to set the qualifier to be hit in order to enter a character‘s ASCII code (see Enter ASCII). You can either use the LEFT ALT key or the CONTROL key. Simply click on the gadget and it will change it‘s state.

1.62 KCommodity/ExpWins

Exploding windows

This is more a hack than a "useful" feature. With an active Exploding Windows function on opening a window frames will be drawed from the center of the screen to the window‘s new position. It seems like the window is smoothly growing to it‘s full size. When closing the window the motion will go the other way around. First the window will be closed and then the frames will be drawed back to the center of the screen. This function does not have any effect on backdrop windows. It does not have to be the screen‘s center. Alternatively you can tell KCommodity to use the actual mouse position as the animation‘s focus. Additionally there is another feature. If you move a window, size it or "zoom" it, frames will be drawed from the window‘s current position to the new one, that means the metamorphosis will be shown. To shorten the time for exploding / imploding the frames on "slow" Amigas you have the possibility to set the number of frames to be drawed for exploding, imploding and sizing / moving / zooming.

1.63 KCommodity/FromToMouse

```
From/To mouse
-------------

   This gadget controls the focus of the Exploding Windows animation. If
selected, frames will be drawed from the current mouse position instead
of the screen‘s center.
```

1.64 KCommodity/ExpFrames

```
Explode frames
--------------

   As mentioned you can define the number of frames to be drawed for
exploding / imploding animations for an active Exploding Windows
function.  Frames in the means of a number of rectangles to be drawed
until the complete size of the window is reached. This gives users of
"slower" computers the possibility to use the Exploding Windows
function because they can find values which do not slow down their
machine a lot. This gadget defines the number of frames to be drawed
when opening a window. The value can range from 0 to 64 frames, where 0
means that no frames will be drawed for that animation.
```

1.65 KCommodity/ImpFrames

```
Implode frames
--------------

   As mentioned above you can select a number of frames here. This time
it is the number of frames to be drawed when closing a window. See
Explode frames, for more information about frame settings.
```

1.66 KCommodity/SizeFrames

```
Size frames
-----------

   Using this gadget you can set the number of frames to be drawed when
resizing a window (see Explode frames).
```

1.67 KCommodity/MoveFrames

Move frames

As with Size frames you control the number of frames for resizing, you can change the number of frames for moving a window using this gadget. (again see Explode frames).

1.68 KCommodity/ClockDef

Clock definitions window
========================

Within this window you find gadgets to control the behaviour and style of the clock/memory display. This display can either be shown in the upper right corner of the Workbench screen, or it may be displayed in a normal window, which can be moved around on the Workbench screen. You have the possibility to choose from different display styles and formats with different information. Here is a list of available "pages".

1. Time/Date display (maybe with mouse coordinates)

2. Memory page, showing chip and fast memory seperately

3. Memory page, showing total amount of free memory

4. Time/Date display combined with memory display

5. Graphical display of available memory. (Gauge shown)

6. Online time, means connect time to a BBS

Here are the gadgets which control the format and look of the clock/memory display. Per default only page 1 will be visible. Refer to HotKey window for more information on how to reach the displays.

Display clock/mem
Display in window
Show time
Show day
Show seconds
Short day string
Show date
Show mouse X/Y
Date Format
Text front
Text back
Gauge front
Gauge back

1.69 KCommodity/DispClock

```
Display clock/mem
-----------------

   You can turn off the displaying of clock/memory informations using
this gadget.  The display will disappear from the Workbench titlebar or
the window will be closed, if you turn off this gadget.
```

1.70 KCommodity/DispWin

```
Display in window
-----------------

   If you prefer to have the display within the titlebar of a small
window, which can be dragged around the Workbench screen, simply turn
this gadget on. The display within the Workbench‘s titlebar will
disappear and the window will pop up. This gadget has no effect if you
turned off Display clock/mem (see Display clock/mem).
```

1.71 KCommodity/ShowTime

```
Show time
---------

   If you do not want to have the time displayed, then you should
simply turn off this gadget.
```

1.72 KCommodity/ShowDay

```
Show day
--------

   You can disable the displaying of the current day using this gadget.
```

1.73 KCommodity/ShowSecs

```
Show seconds
------------

   If the seconds display gets too annoying, then you can disable the
seconds using this gadget.
```

1.74 KCommodity/ShortDay

Short day string

If you believe the day names to be too long for your needs, you may turn on this gadget, in order to get three char day names.

1.75 KCommodity/ShowDate

Show date

You can tell KCommodity to display the current date within the Workbench‘s titlebar or within the window‘s titlebar.

1.76 KCommodity/ShowMouse

Show mouse X/Y

Sometimes you might like to know, where the mouse is positioned at. This function will print out the current mouse position in the clock/memory display. *WARNING*: This function is very CPU-time consuming. If you rapidly move the mouse KCommodity will have to update the display every move it detects.

1.77 KCommodity/DateFormat

Date Format

If you do not like the default date format KCommodity offers. You may choose between six different styles. These are:

- MM-DD-YY
- DD-MM-YY
- MM.DD.YY
- DD.MM.YY
- DD-MMM-YY
- DD.MMM.YY

The latter two will show the month name in an abbreviated form (only three chars).

1.78 KCommodity/TextFront

Text front

You can select a new color for the clock/memory display, if you do not like the default colors KCommodity chooses (depending on which version of the OS you are running). This setting will only affect the colors in which the text will be drawed.

1.79 KCommodity/TextBack

Text back

This gadget will have a similar effect on the clock/memory display as Text front. It will change the text`s background color.

1.80 KCommodity/GaugeFront

Gauge front

As mentioned above KCommodity offers a page for a graphical memory display. This page will show a gauge giving you an overview of available memory. If you do not like the colors used to render the gauge, you may use this gadget, in order to change the gauge`s front color. You may also change the gauges back color using Gauge back.

1.81 KCommodity/GaugeBack

Gauge back

As already explained you may change the gauge`s colors. This gadget is used to change the gauge`s background color. Refer to Gauge front to learn how to change the front pen.

1.82 KCommodity/BlankWin

Blanking
========

KCommodity can blank the screen and mouse for you. For blanking the screen, there are two possibilities. The first is to completely blank the screen, say it will be dark black, or to dimm the screen by a given factor. The mouse may either be blanked after a given timeout and/or it might immediately blank, when a key is hit.

Note that KCommodity‘s mouse blanker is as systemfriendly as possible, that means the mouse will *unblank* when an application tries to change the mouse image or when it opens a window. KCommodity will then reblank the mouse in at least one second. This might lead to a flickering, when the pointer is changed rapidly, so some users might find that one annoying, but I prefer a systemfriendly solution, than a non-stable one, which looks nice.

KCommodity‘s blanker does not have any of these graphical things. I think that there are enough good blankers of this kind out there and Spliner is a very good piece of programmer‘s art. I do not think that it is worth the time to try to copy this program.

Enough of technical talk, let‘s come to the gadgets which control the blanking.

Blank mouse on keystroke
Screen time
Mouse time
Blanker type
Dimm factor

1.83 KCommodity/BlankMouse

Blank mouse on keystroke

If this gadget is active, the mouse will be blanked on every keystroke. If not, the mouse will stay as it is, when you are typing something on the keyboard.

1.84 KCommodity/ScreenTime

Screen time

This slider gadget is used to select the time KCommodity should wait before blanking the Screen. The countdown will always be restarted if the mouse is moved, if a key is pressed, if a disk is inserted/removed, the joystick is used, or a requester pops up. An input of 0 turns off the Screen Blanker/Dimmer.

1.85 KCommodity/MouseTime

Mouse time

As mentioned under Screen time you can enter the amount of seconds before blanking the mouse here. Again, if you enter 0 the MouseBlanker will be turned off.

Note the restrictions on the mouse blanker mentioned in the introduction.

1.86 KCommodity/BlankerType

Blanker type

This gadget is used to determine the type of screen blanking. If *Blanker* is shown the blanker acts as intended. KCommodity will completely blank the screen after the time has elapsed (see Screen time). If this gadget reads *Dimmer* KCommodity will dimm the screen with the given factor. Refer to Dimm factor for more information on setting the dimm factor.

1.87 KCommodity/DimmFactor

Dimm factor

This gadget is only active if *Blanker Type* is set to * Dimmer* (it is not ghosted) (refer to Blanker type, for more information about that). You can now define the factor for dimming the screen. The settings can range from 1 to 48 where 1 is the brightest and 48 the darkest. Just look for the factor which suits your needs. As the dimming is done that way, as it subtracts 1 from each color value until the factor is exceeded, minor color "changes" may occure. If you prefer to have the real colors after dimming then just enter a value which is a multiple of 3.

1.88 KCommodity/CycleWin

```
Cycle settings
==============

   KCommodity offers the possibility to easily flip through screens and
windows. This can either be achieved by using the mouse or by defining
key-combinations to do the cyclings. Within this window you may change
the number of clicks needed to initiate the wanted action and to change
the keys needed to do it.

   Cycling through windows is done by clicking onto a window with the
given key-combinations. If you want to cycle through screens, then
simply click on a backdrop window or the screen itself.


  Cycling enabled
  Front clicks
  Middle clicks
  Back clicks
  Front Key
  Middle Key
  Back Key
```

1.89 KCommodity/CycleOn

```
Cycling enabled
---------------

   Using this gadget you may tell KCommodity not to watch the
key-combinations for screen / window cycling. Turning off this gadget,
no more cycling can be performed.
```

1.90 KCommodity/FrontClicks

```
Front clicks
------------

   This slider gadget determines the number of clicks to occure before a
window/screen to front is to be executed. You can choose from 1 to 5
clicks. These clicks have to occure within the *Double-Click* time.
```

1.91 KCommodity/MiddleClicks

```
Middle clicks
-------------
```

You can define the number of clicks for the middle mousebutton (in conjunction with the key-combination), after which should a screen cycling should be executed.

1.92 KCommodity/BackClicks

Back clicks

This gadget is to be used to determine the number of clicks to be made, in order to execute a screen/window to back.

1.93 KCommodity/FrontKey

Front key

Click on this gadget to define a new qualifier for performing a screen/window to front action. To the right of this gadget you will find the current qualifier. Have a look at Qualifier window, to learn how to use that window for defining a new qualifier.

1.94 KCommodity/MiddleKey

Middle Key

This gadget lets you define a new qualifier to be used in conjunction with the middle mouse button for flipping through screens. When defining a new qualifier (see Qualifier window), you can only change the qualifier itself, not the key to be used (that means the middle mouse button).

1.95 KCommodity/BackKey

Back Key

As easy as you can pop screens/windows to front, you may push them to back using the key-combination shown to the right of this gadget. Clicking onto this gadget will bring up the Qualifier window, allowing you to change the combination for pushing screens/windows to back.

1.96 KCommodity/TagScrs

Tag Screens
===========

Some of you might know TagScreens by Martin Berndt. Thanks to him I implemented the complete TagScreens v1.8 source into KCommodity, and I added some new features and functions to the existing. TagScreens v1.8 only handeled old OS 1.3 screens and patched them for newlook OS 2.x screens. KCommodity goes one step further. It not only patches all screens, it will give you the possibility to do by-task patches, that means you can easily patch PageStream to use a A2024 display. Additionally you may have the patched screens to be auto-centered, if not opened with full width. The AutoScroll bit may be set for a screen and the DrawPens may be selected for the screen to be opened, so that the screen will appear in the new look.

Owners of a ECS-Denise may have the Genlock border blanked, so that there is no unused space on the left and right borders of the screen. KCommodity will turn that border black.

Users of OS 3.0 may have the screens patched to use interleaved BitMaps to increase display update, or to clone the Workbench dimensions and depth by setting one single flag.

So here are the gadget to control the extended Tag Screens features of KCommodity.

TagScreens activated
Task list
Task name
Add entry
Add global
Select entry
Remove entry
Hard Patch
AutoScroll
Font
Monitor
SA-Pens
System Pens
BlackBorder
Auto Center
Interleave
Like WorkBench
ScreenMode
Screen Height
Screen Width

1.97 KCommodity/TagScrsOn

TagScreens activated

This gadget simply turns on/off TagScreens. When the gadget is active, KCommodity will watch all screens to be opened, in order to patch them.

1.98 KCommodity/TaskList

Task list

Within this gadget you will see a list of tasks to be watched. All screens of tasks listed here will be patched to the settings made for them. You might see an entry called "*** GLOBAL TAG ***" which is responsible for all tasks not listed.

1.99 KCommodity/TaskName

Task name

For a selected entry (except for the GLOBAL TAG), you may change the name for the task to be watched. You do not have to worry about the case of the name you enter. KCommodity will check the tasks name case-insensitively. Additionally you do not have to enter the complete name. KCommodity supports all AmigaDOS wildcards.

Here is an example for defining the task of CygnusED

```
#?ED
```

This entry will patch all tasks with a string of *ED* at the end of their names. As you can see, the only important characters are *ED*. You should always use this definition, since CygnusED does something weird while opening it‘s screen, when launched a second time (but who does this that often ?).

Additionally you can tell KCommodity to leave out a task from being patched. This is done by simply inserting a ’-’ in front of the task‘s name.

```
-#?ED
```

This will stop all tasks having a ’ED’ within their names from being patched.

NOTE: You can not change the name of the GLOBAL TAG. If you try to enter **** GLOBAL TAG ****, KCommdity will restore the previously entered name.

1.100 KCommodity/TagAdd

Add entry

This gadget will add a new entry to the list of tasks. The name of the new task will be set to *--- Unused entry ---*. You can now enter the real name of the task to be watched.

1.101 KCommodity/TagGlobal

Add Global

This gadget will tell KCommodity to insert the special *GLOBAL TAG*, which is the one to be used for tasks not available within the list. The entry‘s name may not be changed, it is only a synonym for a global setting.

1.102 KCommodity/SelectTag

Select entry

If you are not sure of how the name of a task is spelled, simply click on this gadget. A window will pop up, with a list of all available tasks currently in the system. Prerequisite for this function to work on the task you desire, is that you first start the task, before selecting this function. If you‘ve found the task within the list, simply click on it‘s name. The name will then be copied to the selected entry. To learn more about that selection window, have a look at Task window.

1.103 KCommodity/RemoveTag

Remove entry

This gadget will simply remove the selected entry from the list.

WARNING: KCommodity will not ask you for confirmation

1.104 KCommodity/HardPatch

Hard Patch

You can tell KCommodity to be safe when patching screens. When this gadget is turned off, KCommodity will only patch those screens, no TagList or extended screen structure is attached to, when opening. So this has only effects on screens opened under OS 1.3 mode.

When the gadget is turned on, KCommodity will patch all screens and set the chosen modes and flags for the screen, no matter whether the screen is opened with a TagList or not. This might be very dangerous, if the application opening the screen relys on the modes and flags set up by the application. So watch out, and try some settings, before blaming me. I told you.

1.105 KCommodity/AutoScroll

AutoScroll

Setting this gadget will tell KCommodity to activate the AutoScroll flag for the screen to be opened.

You might know this feature from the WorkBench screen. When you drag the screen a little bit down and then hit the bottom border of the screen with your mouse, the screen will automatically move upward. The same happens, if you touch the left/right screen borders, and the screen is wider than the maximum display width.

1.106 KCommodity/Font

Font

This gadget tells KCommodity to set the screens font to *topaz8*. If you set *Hard Patch* for the task, the screen will have topaz8, no matter what the application requested (see Hard Patch).

1.107 KCommodity/Monitor

Monitor

This gadget is connected with the *ScreenMode, Screen Height, and

Screen Width* gadgets (see ScreenMode, Screen Height, and Screen Width).

If Monitor is set, these three gadgets are available to be set by you. If not, you can not make any changes to these settings, and KCommodity will not patch the DisplayID, Height, and Width entries of the screen to be opened.

1.108 KCommodity/SAPens

SA-Pens

This is the magic flag for obtaining the OS 2.0 look. If set, KCommodity will insert the appropriate DrawPens when opening the screen. KCommodity will check for OS 2.0 and OS 3.0, when setting the DrawPens, so that the screen always looks good on any system.

1.109 KCommodity/SYSPens

System Pens

Per default KCommodity will determine which set of DrawPens to use (see SA-Pens). These pens will be set according to the version of OS you are running. Under OS 3.0 three new pens were introduced, but adding these to a screen, which believes the colors of the titlebar to be some specific kind, could result in ugly screen rendering (Like CygnusED). If you unset this gadget, KCommodity will always supply the OS 2.0 DrawPens, in order to keep the titlebar untouched.

1.110 KCommodity/BlackBorder

BlackBorder

This flag tells KCommodity to blank the Genlock border to the left and right of the screen, which will normally be black.

NOTE: This gadget/setting is only available, if you have an ECS-Denise equipped computer.

1.111 KCommodity/AutoCenter

AutoCenter

Using this gadget you may horizontally center screens, which do not open with full width.

1.112 KCommodity/Interleave

Interleave

This gadget is only available under OS 3.0 or later.

It will tell KCommodity to set the Interleaved-BitMap-Flag when opening the screen. Interleaved BitMaps will result in a somewhat faster display refreshing/drawing. Somehow there might be problems with older programs or with programs working with the old BitPlane format (like View80).

1.113 KCommodity/LikeWB

Like WorkBench

This gadget is only available under OS 3.0 or later.

If you want to have a new screen to be opened with the dimensions and depth just like WorkBench, simply activate this gadget.

1.114 KCommodity/ScreenMode

ScreenMode

This gadget is only available, if you activated Monitor. You can use this gadget to select a new screen mode for the task‘s screens. A requester will pop up listing all available screenmodes. This will either be an ASL-ScreenModeRequester or a ReqTools ScreenModeRequester (Version 38.1022 of the reqtools.library is included), depending on the OS you are running.

1.115 KCommodity/ScrHeight

```
Screen Height
-------------

   This gadget is used to set a specific screen height for the screen
to be opened. Simply enter the height you prefer for the screen.

   If you supply 0 for the height, KCommodity will simply use the
screen‘s default height.
```

1.116 KCommodity/ScrWidth

```
Screen Width
------------

   Similar to *Screen Height* you can set a specific screen width (see
Screen Height, for more information), for the screen to be opened.
```

1.117 KCommodity/Mouse

```
Mouse
=====

   Within this window you can control mouse specific features offered by
KCommodity. Here you find gadgets for defining the type of window
activation you prefer, gadgets for setting the mouse acceleration and
some other functions, you will get to know, when reading this section.

   For additional informations and topics concerning window activation
refer to Lock windows.


  Window activation
  Sun mouse
  Mouse accelerate
  Window reactivation
  Acceleration
  Threshold
  LeftyMouse
  Hold mouse
  Hold X qualifier
  Hold Y qualifier
```

1.118 KCommodity/WinAct

```
Window activation
-----------------
```

This gadget enables/disables window activation. The type of activation is determined by Sun mouse. If you turn off this gadget, no window activation will occure.

1.119 KCommodity/SunMouse

Sun mouse

This gadget determines the type of window activation. If set, all windows will be activated, that are touched by the mouse. That doesn‘t really mean, every window will be activated. If you move the mouse at a high speed, only that window will be activated, the mouse comes to a stop over. If you move the mouse at a slow speed (of about 5-10 pixels in either direction), all windows will be activated, when they are touched by the mouse.

If you turn off this gadget, KCommodity will only activate windows, the mouse is over *and* when a key is hit.

1.120 KCommodity/MouseAcc

Mouse accelerate

This gadget will turn on/off mouse acceleration. The acceleration is controled by the gadgets Acceleration and Threshold.

1.121 KCommodity/WinReAct

Window reactivation

Per default, KCommodity always remembers the currently active window on each screen you have been on. When returning to a screen, KCommodity will look for the "active" window on that screen and will reactivate it. You can disable this behaviour by using this gadget.

1.122 KCommodity/Accel

```
Acceleration
------------

   This gadget sets the acceleration for the mouse. Values between 2
and 10 are accepted, that means you define the acceleration factor for
the mouse.
```

1.123 KCommodity/Thresh

```
Threshold
---------

   This slider gadget is used in conjunction with Acceleration. It sets
the threshold for mouse acceleration. The higher the value, the faster
you have to move to the mouse, before KCommodity will start to
accelerate.
```

1.124 KCommodity/LeftyMouse

```
LeftyMouse
----------

   This is for users, who prefer to have the mouse buttons swapped
(left and right button). All functions the buttons normally perform,
are swapped from left to right now.
```

1.125 KCommodity/HoldMouse

```
Hold mouse
----------

   In conjunction with the correct qualifier, you can lock
mousemovement in one or both directions (X-direction or Y-direction).
This gadget turns on/off mouse movement locking.
```

1.126 KCommodity/HoldX

```
Hold X qualifier
----------------

   This gadget will open the Qualifier window.  You can then select a
new qualifier to be used in order to lock mouse x-movements.
```

1.127 KCommodity/HoldY

```
Hold Y qualifier
----------------

   This gadget has a similar meaning as the one just described (see
Hold X qualifier). You can re-define the qualifier for locking mouse
y-movements.
```

1.128 KCommodity/HotKeyWin

```
HotKey window
=============

   This window allows you to change all hotkeys supported by
KCommodity. The HotKeys are only available, when the main program is
running. Otherwise you can only change the HotKey definitions. You can
reach several functions only when there are available HotKeys.

   Have a look at Defining a HotKey, to get to know the descriptive
strings for HotKeys.

   If you enter "None" for the descriptive text the corresponding
HotKey will be disabled.

   For a complete list of available HotKeys refer to
KCommodity`s HotKeys.

   This window only offers two gadgets:


  HotKeys
  Def
```

1.129 KCommodity/HotKeys

```
HotKeys
-------

   This listview gadget shows all available HotKeys by their names. If
you click on a HotKey`s name, it`s definition will be copied to the
string gadget. (see Def).
```

1.130 KCommodity/Def

Def

This string gadget is used to define a new HotKey for a selected entry. If you are not used to HotKey definitions the commodities system uses, refer to Defining a HotKey.

If you enter a string, which contains an incorrect definition, KCommodity will restore the previously used definition.

1.131 KCommodity/HotKeyDef

Defining a HotKey

To ease the definition of a HotKey here is a list of key-descriptions the system "knows".

Here is a list of qualifiers.

'ALT'
    Any Alt key.

'RALT'
    Only the right Alt key.

'LALT'
    Only the left Alt key.

'SHIFT'
    Either Shift key.

'RSHIFT'
    The right Shift key.

'LSHIFT'
    Only the left Shift Key.

'CAPSLOCK'
    Capslock must be pressed.

'LCOMMAND'
    Left Amiga key.

'RCOMMAND'
    Right Amiga key.

'CONTROL'
    Control Key.

'NUMERICPAD'
    Keys of the Numeric Keypad are accepted.

'RBUTTON'
     Right mousebutton.

'MIDBUTTON'
     Middle mousebutton.

'LEFTBUTTON'
     Left mousebutton.

'NEWPREFS'
     Preferences have been changed. Maybe a new font was selected.

'DISKREMOVED'
     A disk was removed from a drive.

'DISKINSERTED'
     A disk was inserted into a drive.

These are the qualifiers you can use in almost every combination. Now we come to the descriptions of "normal" keys.

'A - Z, 0 - 9, etc.pp.'
     Normal keys, as you see them on the keyboard.

'F1 - F10'
     The ten function keys.

'UP, DOWN, LEFT, RIGHT'
     The cursor keys up, down, left, and right.

'HELP'
     The Help key.

'DEL'
     Delete key (beside HELP).

'BACKSPACE'
     Backspace key

'RETURN'
     Return key.

'ENTER'
     Enter on Numeric Keyboard (only, if NUMERICPAD was specified).

'ESC'
     Escape key.

'SPACE'
     Space.

'COMMA'
     A comma.

'UPSTROKE'
     The apostroph (to the left of the 1).

You can use as many qualifiers as you wish, but only one key per description. This is the syntax for a HotKey definition:

```
                    [QUALIFIER [QUALIFIER ...]] KEY
```

As you can see the qualifiers can be totally left out, but in how far this is usefull depends on you. Entering "None" (without quotes) will disable a HotKey.

1.132 KCommodity/KCXHotKeys

KCommodity's HotKeys

There are several different HotKeys managed by KCommodity. Each of these may be changed alternatively. To find out, what the HotKeys are used for, simply get to the HotKey in question and check it out.

```
  CX_PopKey
  Bill
  Formatter
  Page
  UserShell
  Map
  Snap memory
  Immediate blank
  Display dump
  Center
  Select Screen
  Clock/Memory on/off
  Function keys on/off
  SunMouse on/off
```

1.133 KCommodity/CXPop

CX_PopKey

You might already know this HotKey. This is the one to let the Preferences Editor pop up. Per default this will be LCOMMAND HELP. Note that the settings will override the ToolType definition (see Using ToolTypes).

1.134 KCommodity/Bill

Bill

This HotKey is used to tell KCommodity to open the Bill window (see Bill window). Per default this will be ALT SHIFT B.

1.135 KCommodity/Formatter

Formatter

This HotKey will let you open the Formatter window (see Formatter window), so you may format drives "manually". The default definition for this HotKey is ALT SHIFT F.

1.136 KCommodity/Page

Page

As mentioned before you can use different pages for the clock/memory display (see Clock definitions window). This is the HotKey to cycle through the different pages available. Have a look at the corresponding section, to see, which pages are available. The definition is RALT HELP.

1.137 KCommodity/UserShellKey

UserShell

As mentioned before, KCommodity can open a shell when a HotKey is pressed. Per default, this HotKey is ALT SHIFT S. You can change it here.

1.138 KCommodity/MapKey

Map "Umlauts"

As mentioned under Map , KCommodity may map the german "Umlauts" for you. You do not have to go to the Settings window every time you want to turn that function on/off. Simply hit the HotKey to toggle that function. Per default this HotKey will be ALT SHIFT U.

1.139 KCommodity/SnapMem

Snap memory

You can define the HotKey for the "Snap memory" function here. The first time you hit the HotKey the screen will flash. This signalizes that KCommodity made a snapshot of the currently available memory. If you hit the HotKey a second time the time/memory display will change to a different page showing how much memory was freed / got lost. A possible display could be:

```
                   Snapped  C:      0   F:        0
```

This display tells you that neither chip memory nor fast memory was occupied / freed during the time between the two calls of the "Snap memory" function. If memory got lost the values shown will be negative, otherwise they will be positive.

This display will only show up if the "Snap memory" function is used. The Page HotKey can *not* be used to switch to that page.

To get back to the normal display just hit the Page HotKey. The default setting for this HotKey is ALT SHIFT G.

1.140 KCommodity/ImmBlank

Immediate blank

If you do not want to wait for the screen timeout to elaps you may tell KCommodity to blank the screen immediately. This is done by hitting this HotKey which is ALT SHIFT D per default.

1.141 KCommodity/DispDumpKey

Display dump

Hitting this HotKey will tell the main program to start the external printer driver. The default HotKey definition is ALT NUMERICPAD *, which stands for the key labeled ’PrtSc’ to the upper right.

1.142 KCommodity/CenterKey

```
Center
------

   If you moved a screen to the left/right and want to recenter it,
then you should simply hit this HotKey, in order to perform that
action. The default definition for this HotKey is ALT SHIFT X.
```

1.143 KCommodity/SelScr

```
Select screen
-------------

   This HotKey will open the Select screen window. You can then choose
a screen to be brought to front. Have a look at Select screens, for
more information about that.  The default setting for this HotKey is
LSHIFT ESC.
```

1.144 KCommodity/ClockMemKey

```
Clock/Memory on/off
-------------------

   You do not have to turn off the clock/memory display via the Clock
definitions window. Simply hit this HotKey to perform that action. The
default definition is ALT SHIFT C.
```

1.145 KCommodity/FuncKey

```
Function keys on/off
--------------------

   If your function keys collide with the ones defined by an
application you may turn off KCommodity`s function keys by hitting this
HotKey. The default definition for this HotKey is ALT SHIFT K.
```

1.146 KCommodity/SunKey

```
SunMouse keys on/off
--------------------

   Using this HotKey you may switch between SunMouse windowactivation
```

and the method of hitting a key in order to activate a window. Default definition is ALT SHIFT M.

1.147 KCommodity/PrefsGfxDump

Graphics dump
=============

This gadget will inform KCommodity (the main program) to launch the external graphics dump module. If you didn‘t start KCommodity, this gadget will be ghosted and so this function is unavailable. Have a look at The Printer Driver, for more information about the graphics dump module.

1.148 KCommodity/FKeys

Function Keys
=============

KCommodity offers the possibility to map strings onto the function keys. You are not restricted to only one mapping per key, you can have up to four different strings for each function key. To reach these strings you simply have to hit a qualifier.

Additionally there are three special *strings*, you can use. These can perform special functions KCommodity offers.

As said under Function keys on/off, KCommodity‘s function key mappings may be turned on/off by hitting a HotKey.

F1 - F10
Qualifier
Active
Ok
Cancel

1.149 KCommodity/F1F10

F1 - F10

These string gadgets are used to enter the strings for each function key to be mapped when hit in conjunction with the chosen qualifier. But not only normal characters are supported. For special control sequences have a look at the following list:

- ANSI - Any ANSI sequences may be entered.

- Special codes starting with \
  1. \n - Return

  2. \r - Return

  3. \t - TAB

  4. \ - \-Code

- Keydefinition between <> An example: abc<alt f1>\nHello. This would insert the definition of ALT + F1 between abc and \n. As you can see these definitions are the same as presented in Defining a HotKey.

And here are the special strings mentioned above:

- MINWIN The window underneith the mouse pointer will be changed to minimum size and moved to the upper left corner of the screen.

- MAXWIN The window underneith the mouse pointer will be brought to maximum size and moved to the upper left corner of the screen.

- ZIPWIN Windows equiped with a Zoom Gadget can be handeled using this command. The window underneith the mouse pointer will zip to it's alternative size/position.

1.150 KCommodity/FKeyQual

```
Qualifier
---------
```

This gadget determines the qualifier to be used in order to insert a given function key's string. Available qualifiers are :

- NONE No key.

- SHIFT Either SHIFT key.

- ALT Either ALT key.

- CONTROL CONTROL key.

1.151 KCommodity/FKeyAct

```
Active
------
```

This gadget sets the state of KCommodity's function key mapping facility. If active, the function keys will be mapped according to your settings. This gadget may also be used via HotKey (see Function keys on/off, for more).

1.152 KCommodity/FKeyOk

Ok
--

Selecting this gadget will inform KCommodity to copy your function key settings to it's internal buffer.

NOTE: The new function keys are only available when you clicked onto this gadget, no matter whether *Active* is selected or not. See Active, for more information.

1.153 KCommodity/FKeyCancel

Cancel

Clicking onto this gadget will tell KCommodity, that you wish to cancel the setting of function key strings. All changes made to the function keys will be lost, and the previously used keys will be re-installed.

1.154 KCommodity/LockWins

Lock windows
============

Here you find some additional possibilities for manipulating the *Window activation* facilities of KCommodity (see Mouse, for more information about window activation).

KCommodity will allow you to specify certain windows which are *not* to be deactivated by the program's window activator, that means, no matter which type of activation selected, the window will stay active, once activated. You can move the mouse out of the window, hit a key, or do anything else, the window will stay active. The only way to activate another window is to use the conventional method (clicking on the window in question). This feature can only be used with windows having a title.

Additionally you may define Tasks, who's windows are to be protected from deactivation. Simply supply the name of the task you which and KCommodity will lock all of it's windows.

All entries are treated case-insensitive. AmigaDOS wildcards are allowed and supported.

```
  Locked list
  Entry
  List
  Add entry
  Select entry
  Delete entry
```

1.155 KCommodity/LockList

Locked list

This listview gadget will show you all names of windows/tasks to be locked. Depending on the type of list being displayed you will see names of windows or names of tasks (see List, for more). Simply click on an entry to change the name. The name will be copied to the string gadget underneith the listview gadget.

1.156 KCommodity/LockEntry

Entry

This string gadget will show the name of the selected entry. You can edit the name here, so that it suits your needs. Do not worry about how to write the name, they are all treated case-insensitive (you will have to spell it correctly, though). AmigaDOS wildcards are supported here.

1.157 KCommodity/LockType

List

As mentioned before, KCommodity offers two methods for locking windows. The first is to lock a window simply by supplying it‘s name. The second is to supply a task‘s name, who‘s windows are to be locked. To tell KCommodity how to treat the entries, you have to select the appropriate list, using this gadget. If it shows *Windows*, all entries are treated to be names of windows. If it shows *Tasks*, the entries are treated to be names of tasks.

1.158 KCommodity/LockAdd

Add entry

Clicking onto this gadget will tell KCommodity to add an entry to the end of the list. This entry will have the name of *--- Unused entry ---*. The string gadget will become the active gadget, allowing you to enter the name of the window/task to be locked.

1.159 KCommodity/LockSel

Select entry

This gadget will give you the possibility to select an existing window/task to be locked. Depending on the type of list being edited, there are two ways to specify the name of an existing entry.

For windows you will have to click onto the window in question. Having selected this gadget, the window‘s titlebar will change telling you what to do : *Click on window to use*. You are questioned to click onto the window you want to add to the list of windows. If you try to select a window which does not have a titlebar text, KCommodity will inform you, that it can not handle something like that.

For tasks a window will pop up, showing a list of all tasks available. Simply click onto the task you want to add to the list of tasks. See Task window, for more information about this window.

1.160 KCommodity/LockDel

Delete entry

This gadget will have the effect of deleting the selected entry.

WARNING: KCommodity will *not* ask for confirmation.

1.161 KCommodity/PrefsFormat

Formatter
=========

This gadget is only available, if the main KCommodity is running. Otherwise it is ghosted. Using this gadget you tell KCommodity to open it‘s formatter window.

1.162 KCommodity/BillPrefs

Bill preferences
================

This window is used to set the various elements for telephone cost calculation. Here you find several gadgets to specify the length of a unit for all three timezones supported (either cheap or normal tariff). Additionally you can tell KCommodity which weekdays are to be treated as *always cheap*, that means, KCommodity will always use cheap tariffs then.

To have more information about telephone cost calculation refer to Telephone cost calculation.

Normal 1 - Normal 3
Cheap A - Cheap C
Cost/Unit
Sunday - Saturday
Start normal/Start cheap
Pop up Bill window
Activate on PopUp
Clear Log
Display total sum
Open Bill on
Log Filename
Select

1.163 KCommodity/NormalT

Normal 1 - Normal 3

These gadgets are used to enter the times for one unit concerning the normal tariff. The numbering from 1 to 3 is for the three available time zones. Within these gadgets only numerical inputs are allowed. Negative inputs will lead to a restore of the previously used value. The program believes these values to be seconds, that means you will have to calculate the correct values to enter minutes.

1.164 KCommodity/CheapT

Cheap A - Cheap C

These gadgets almost have the same meaning as the ones mentioned above (see Normal 1 - Normal 3). The only difference is that the times entered are for the cheap tariffs of the three time zones.

```
                        An example for an input

       Normal time / unit 1: 360   Means 6 minutes for one unit.

       Cheap time / unit 1: 720   Means 12 minutes for one unit.
```

The default times correspond to these valid in Germany.

1.165 KCommodity/CostUnit

Cost/Unit

This gadget only accepts numerical inputs, too. Enter the costs for one unit. These inputs are made in cents, that means that an input of 23 is interpreted as 23 Pfennige (cents). Negative values will lead to a restore of the previously used value, too.

1.166 KCommodity/SunSat

Sunday - Saturday

These gadgets mark weekdays for use of cheap tariff the whole day. A selected gadget tells KCommodity that normal tariff is not to be used on that day.

1.167 KCommodity/NormalCheap

Start normal/Start cheap

These gadgets are used to determine the start times for normal and cheap tariffs. KCommodity will check the system time and alter the tariff accordingly. *Start normal* is used to set the start time for normal tariff and *Start cheap* is used for cheap tariff. If the border between normal/cheap is passed during connection time, no changes to the costs are made, but all forthcoming units will be calculated correctly.

1.168 KCommodity/PopBill

```
Pop up Bill window
------------------

   Using this CheckBox gadget you can determine whether KCommodity
should open the Bill window automatically whenever it recognizes a
Carrier Detect. If the Carrier is gone (hanging up the modem)
KCommodity will close the window again. If the window was already
present on a Carrier Detect the program will not close the window when
you hang up.
```

1.169 KCommodity/PopAct

```
Activate on PopUp
-----------------

   Per default the Bill window will *not* be activated, when opened. You
can tell KCommodity to activate the window on open-up, using this
gadget.
```

1.170 KCommodity/ClearLog

```
Clear Log
---------

   Selecting this gadget will tell the program to clear the log file
with all the information about the past calls.
```

1.171 KCommodity/DispSum

```
Display total sum
-----------------

   Attention, this gadget could become dangerous :-) When you click
onto this gadget a Requester will be opened asking where to put the
listing to. You can choose from a file or a window. If you selected
"Window", a window will be opened showing information about the calls
you‘ve made with your modem. After all entries for one day have been
listed KCommodity will print out the sum for that day. At the end of
the list the program will give you the total sum for all calls made.
This is also done when redirecting the listing to a file.
```

1.172 KCommodity/OpenBillOn

```
Open Bill on
------------

   You can enter a list of Public Screen names here, on which
KCommodity should open the *Bill* window. The names are seperated by
semikolons ";".  Please remember that KCommodity tries to open the
window on the first screen available, that means that if the first
screen of the list is present the window would *never* be opened on the
second screen. An example:

                           TERM;PowerPacker

   KCommodity would try to open the window on Term`s (1) screen. If
this screen is not available the program will then try to find
PowerPacker`s (2) screen to open the window on.

   Remember that the search for the names is case-sensitive, that means
KCommodity can *never* find a screen called "Term".

   ---------- Footnotes ----------

   (1)  A very powerful terminal program by Olaf 'Olsen' Barthel

   (2)  A very efficient packer by Nico Francois
```

1.173 KCommodity/LogName

```
Log Filename
------------

   If you are already sure about the log file`s name and location you
can easily enter the path / filename here. KCommodity will make sure
that it is possible to write the file otherwise the previously used
file will be reselected.

   If you do not really know where to place the logfile, use Select to
do it the easy way.
```

1.174 KCommodity/LogSelect

```
Select
------

   When you select this gadget a file requester will open up giving you
the possibility to select a new log file.
```

1.175 KCommodity/AlarmModes

Alarm modes
===========

KCommodity may wake you up after a given time, and/or to every full hour. It will then display a requester (if desired) and play a sound. This sound can either be a normal DisplayBeep() sound or an IFF-8SVX sample.

Alarm on
Every hour
Show requester
Alarm time
Sound type
Select sample
Test sample

1.176 KCommodity/AlarmOn

Alarm on

This gadget simply turns on/off the alarm function of KCommodity. If turned off KCommodity will not play any alarm.

1.177 KCommodity/EveryHour

Every hour

If you want to have an alarm every full hour (in addition to the specified alarm time), you may tell KCommodity to do so, via this gadget. Turn this gadget on, to be alarmed every hour.

1.178 KCommodity/ShowReq

Show requester

KCommodity may either show a requester when it alarms you, or it will simply play the selected sound. Turning on this gadget will tell KCommodity to pop up a requester to alarm you (in addition to the sound).

1.179 KCommodity/AlarmTime

```
Alarm time
----------

   Using this gadget you select the alarm time. The input does not
accept seconds however, but I think hour and minutes should suffice.
```

1.180 KCommodity/SoundType

```
Sound type
----------

   Here you can select the type of sound to be played on "wake-up". When
using *DisplayBeep()* the display will flash and a soft beep will be
heard. If you do not like that type of behaviour, simply select *IFF
Sample*. Now you can select an IFF-8SVX file to be played every alarm.
```

1.181 KCommodity/SelSample

```
Select sample
-------------

   This gadget is only available, if you selected *IFF Sample* under
*Sound type* (see Sound type). If you click on this gadget a
FileRequester will pop up giving the possibility to select a new
sample. Underneith this gadget the name of the selected sample will be
displayed.
```

1.182 KCommodity/TestSample

```
Test sample
-----------

   Clicking onto this gadget will play the selected IFF-8SVX sample.
```

1.183 KCommodity/PrefsBill

```
Bill
====

   This gadget will tell KCommodity (the main part) to open it`s bill
window.  See Bill window, for more information about this window. This
```

gadget is only available (not ghosted), when KCommodity is running.

1.184 KCommodity/QualWin

Qualifier window
================

This window is used by various functions to define a key-combination for performing special function (eg. Full drag).

When opened all special functions of KCommodity will be deactivated to guarantee a non-interrupted definition. Simply hit the key-combination you want to use. KCommodity will then display all keys hit within a small box. If you wonder why you can not change the code (main key within single quotes) it is because of the function the key-combination is for. For example you can not change the code for ESC Window Close (see ESC Window Close).

NOTE: When defining a combination including the left mousebutton, you must not click within the text box, nor on one of the gadgets because these are all gadgets and Intuition will consume the clicks, before getting to KCommodity.

1.185 KCommodity/TaskWin

Task window
===========

Within this small window you will find a list of all tasks currently loaded into system. The names of the tasks are not sorted, so you may have to search a while before finding the needed task. This window will be opened by some functions, in order to let you decide which task to use for the specified function. You can cancel the selection by closing the window (clicking onto the close gadget).

1.186 KCommodity/DisplayDump

The Printer Driver
****

This external module is used to dump graphics to either a printer (which should support graphics of course) or an IFF-ILBM graphics file. You can simply define the source which should be dumped and select an area or the complete image. KCommodity can have whole screens or only windows as sources.

When launched the printer module will open a screen, which will have the same dimensions and resolutions as the Workbench screen. On that screen, there will be the graphics dump window.

You have complete access to the printer preferences. These will not be copied to/from the original preferences set up by the Printer and PrinterGfx editors, so that you may choose your own defaults.

On startup the printer module will search for the active screen and the active window. These will be given as sources you can dump from and they may changed later.

```
  Aspect
  Shade
  Dithering
  Scaling
  Density
  Threshold
  Image
  Red, Green, Blue
  Size
  Centered
  Smoothing
  FormFeed
  Keep Aspect
  Dump type
  Graphic
  X, Y, W, H
  Source
  Reselect Source
  Define Area
  Dump
  Cancel
```

1.187 KCommodity/Aspect

```
Aspect
======
```

You can select the dump‘s aspect. The area will either be printed " Horizontal", that means horizontally, or "Vertical", say vertically, rotated by 90 degrees.

1.188 KCommodity/Shade

```
Shade
=====
```

You can select the dump-quality, better the creation of shadows. Following possibilities are offered.

- Black & White The dump will simply be black/white.

- Grey Scale 1 Grey scales will be generated for the dump.

- Grey Scale 2 Like "Grey Scale 1", but this one is preferable for high-resolution displays and should only be used with four color screens.

- Color If your printer supports color dumps, you can tell KCommodity to dump with colors.

1.189 KCommodity/Dithering

```
Dithering
=========
```

This gadget controls the dithering facilities, so you can control the dump quality. These are the possibilities:

- Ordered The dump dithering will be ordered.

- Halftone Dithering will be generated using the halftone method.

- Floyd-Steinberg One of the best picture manipulation algorythms around. This option requires a bit more time to produce the dump, but it will give best results.

1.190 KCommodity/Scaling

```
Scaling
=======
```

You have several different options for scaling the output, so that you do not waist precious space on the paper. You can choose from the following scalings:

- Full Page The whole paper will be filled.

- inch x inch The values of Size will be treated as inches (see Size, for more about that).

- mm x mm KCommodity assumes the "Size" values to be milimeters.

- cm x cm Now "Size" values are treated as centimeters.

- Fraction Using this setting, you can control how much percent of the paper should be filled. The "Size" values are treated as these percentages.

1.191 KCommodity/Density

Density
=======

This gadget controls the dump‘s density. The higher the density, the blacker the dark areas of the dump.

1.192 KCommodity/Threshold

Threshold
=========

You can define the threshold for black/white areas. The higher the threshold, the darker the dump.

1.193 KCommodity/Image

Image
=====

Controls the image to be created. It can either be "Positive" or a "Negative" image.

1.194 KCommodity/RGB

Red, Green, Blue
================

When "Color" is selected for the "Shade" gadget, these gadgets are available to control color correction (see Shade).

1.195 KCommodity/GfxSize

Size
====

These gadgets control the scaling sizes for the dump. Take a look at Scaling, to see how these values are interpreted. These values are ignored if you select "Full Page".

1.196 KCommodity/Centered

Centered
========

This gadget tells KCommodity to horizontally center the image.

1.197 KCommodity/Smoothing

Smoothing
=========

This gadget is unavailable, when using "Floyd-Steinberg". Otherwise you can tell KCommodity to smothen sharp edges of the output, in order to avoid that typical "stair-look" of diagonal lines etc.pp.

1.198 KCommodity/FormFeed

FormFeed
========

When finished dumping, KCommodity will perform a FormFeed.

1.199 KCommodity/KeepAspect

Keep aspect
===========

If you manually changed the scaling, this gadget tells KCommodity to keep the picture‘s original aspect/scaling ratio.

1.200 KCommodity/DumpType

Dump type
=========

This gadget determines the dump‘s destination. If you select "Printer" KCommodity will dump the graphics to a printer. If you select "File" instead, you will be prompted for a file to dump the graphics to.

1.201 KCommodity/Graphic

```
Graphic
=======

   This gadget controls the area to be dumped from. If "Complete" is
selected, the whole source will be dumped, but if you select "Area" then
you can define an area which should be dumped. See Define Area, to see
how to define an area.
```

1.202 KCommodity/XYWH

```
X, Y, W, H
==========

   These Integer gadgets control the area to be dumped. They are
unavailable if you select "Complete" for the "Graphic" gadget (see
Graphic).
```

1.203 KCommodity/Source

```
Source
======

   As mentioned earlier KCommodity searches for two sources. Here you
can define which of the two should be used as source for the dump.
KCommodity tries to show the source`s name to the right of this gadget.
```

1.204 KCommodity/NewSource

```
Reselect Source
===============

   Clicking on this gadget will change the window`s titlebar, telling
you to select a new source by clicking on it. So just move the mouse
over a new window/screen you wish to dump and hit the left mouse button.
KCommodity will fetch data for the new sources.
```

1.205 KCommodity/DefArea

Define Area
===========

Having selected "Area" you may now define the dump-area by using the mouse. The selected source will pop to front and a rectangle will be drawed from the upper left corner to the current mouse position. To set a new start position for the rectangle just click the left mouse button. To quit simply hit the right mousebutton. The new values will be copied to KCommodity‘s integer gadgets.

1.206 KCommodity/Dump

Dump
====

This gadget will start the dump. KCommodity will first check your inputs before starting the dump. If any errors are encountered KCommodity will pop up a Requester "Please check your inputs". Otherwise your printer will soon start to dump. To cancel the dump, simply hit the "Cancel" gadget. If the selected printer does not support graphics dumps KCommodity will inform you.

If you are dumping to a file a FileRequester will first pop up and you can select a file then. KCommodity will dump the data to that file.

1.207 KCommodity/DumpCancel

Cancel
======

Guess what ? It aborts the dump.

1.208 KCommodity/TelCost

Telephone cost calculation
****

Some of you own a modem but as you probably know (including me) you can never definitly say how much the calls cost you made. That is why I included this feature.

The telephone bill calculator can be fully configured to your needs. You can choose between three different time zones either normal or cheap tariff for all of which you may set the duration for one unit. KCommodity can not deal with one hundreds of a second but anyway, inaccurances may occure while counting the time because of intolerances (somewhat of about 1/100 seconds is not that bad I guess and fixing that problem would have caused some more trouble). Maybe there will be a better possibility to enter the time. You can enter the "cost" for one unit (without that it really would not make much sense). The input is made in somewhat like cents, Pfennige, say in micros, but I think that is sufficient, since there might not be an currency which counts in millis. Furthermore you can define the times for normal and cheap tariffs. KCommodity will check the actual system time to see which of the two tariffs should be used. Weekdays on which cheap tariff should be used the whole day may be marked, too.

As soon as the program detects a Carrier, that means if the modem establishes a connection to a BBS, KCommodity will start to count the time. You can tell the program to open up the Bill window on a Carrier Detect. Data about the phone call may be written to a log file, so that it can be analysed later and you can get an overview of arisen costs. KCommodity also aids you in listing the data. The program will print out all entries of the log file giving a total summary after the listing of one day and a final sum after all entries have been listed. This printout may be redirected to a file or you can view it in a window.

As mentioned above I do not take any responsibility for the correct working of the telephone bill calculation, because slight inaccuracies may occure within the system‘s timer. Additionaly it is of importance that you enter almost perfect times, because KCommodity‘s results are only as good as your inputs are. This function has only been implemented to have a rough overview of costs.

References: Bill preferences, Bill window

1.209 KCommodity/RCS

Revision Control System
****

This really is a fine feature. It might be interesting for the programmers of you, which are working on bigger, or even on smaller, projects. While working on a program it might be nice to see which changes were made to a program from this to that version. Inserting this data by hand is not quite that what I call simple. This is where KCommodity‘s Revision Control System gets to work. This feature aids you in inserting good looking Revision headers directly into your sourcecode. All you have to do is to tell KCommodity the program‘s name to be revised, what kind of sourcecode it is and what information KCommodity should insert into the Revision header. For the type of sourcecode you can choose between Assembler, Basic, ‘C’, and Pascal / Modula. The main layout of the header is kept but only adjusted to perfectly fit into the desired type of sourcecode.

The special thing about the function is the way it is called. The Revision Control System can *only* be called via ARexx. That is why I present the syntax for that call now:

```
REVISE Filename ASM|BAS|C|PAS
```

As you can see all you have to do is to feed the function with the filename and an identifier for the type of sourcecode. Having started the Revision Control System KCommodity will first search the sourcecode whether there already is a Revision header inserted in the specified language. There is no need to position the header at the first line of the source. KCommodity will search within the first *15* lines to find the header, giving you the possibility to add specific directives in front of the header. So it is possible to insert text in front of the header(1). No matter if there already is a header this search will always be made when the Revision Control System is called. From now on KCommodity will go different paths in manipulating the source, depending on the existence of a header.

KCommodity comes with several ARexx scripts for use with the supported langauges and for the CygnusED and TurboText editors.

Make sure that the logical directory ‘T:’ is accessable, because KCommodity needs that directory to temporarily store it‘s work. To be secure I advise you to make a backup of your sourcecode before you start the Revision Control System. (One can never know)

If you want the program to store your name and the name of your ‘firm’ within the header you will have to set two environment variables.

- AUTHOR Set this variable to your name.

- COMPANY Enter your firm‘s name here.

Now following is a small example for setting these variables:

```
SETENV AUTHOR "Kai Iske"

SETENV COMPANY Gunthersoft
```

As you can see parameters which contain spaces have to be enclosed by quotes. Insert these settings within your ‘s:User-StartUp’ file to have the variables available on every bootup.

First revision	How to invoke the RCS the first time
Following revisions	Work on an already revised sourcecode
Special features	Special features and an example

---------- Footnotes ----------

(1) Some Assemblers require special information within the first line of the source.

1.210 KCommodity/FirstRev

First revision of a sourcecode with the Revision Control System
===

Assuming this situation I will now point out the actions KCommodity performs when first used on a non-revised sourcecode. A window will be opened with the following title:

Enter a Purpose Message

As the title says you are now to enter a purpose for the sourcecode you want to revise. You have the possibility to enter 5 lines and 60 chars per line. When you are done with the purpose message you can enter some more specific data. You can tell KCommodity which version number to use for the first revision. This input is made within the integer gadgets at the bottom of the window.

The integer gadget on the left is used to enter the version number. Into the right gadget you can enter the revision number. The gadgets *OK* and *CANCEL* are self-explanatory. Whilst OK starts the revision CANCEL is used to abort. The gadget Only text will be unavailable for the first revision. Have a look at
Revision of an already revised sourcecode, for more about that.

Assuming that you have made all the input and started the Revision Control System I will now describe the actions KCommodity will perform for building the header. At first a part of the standard header will be built. The production will be made within ‘T:’. If the environment variables are set the author‘s and the ‘firm‘s’ name will be stored within the header. If KCommodity was not able to find one variable, or even both of them, the corresponding fields will be set to ‘--- Unknown ---’. Furthermore the filename and the actual system date will be inserted. The current revision will be set to the value you specified or to the value KCommodity automaticaly uses. Now the ‘purpose’ text will be inserted and finally a ‘log message’ for the current revision will be appended to the header. This log message will be set to ‘--- Initial release ---’ for the first revision, because it is assumable that no changes have been made. Behind the log message KCommodity will insert six macros, which reflect date of the revision, time of the revision, name of the author, who made the revision, the current revision number of the sourcecode and version/revision numbers. Please have a look at The revision header‘s special features, for further information. The Revision header will be created at the start of the sourcecode, so you will have to insert lines which have to be in front of the header, after you revised the source.

1.211 KCommodity/AgainRev

Revision of an already revised sourcecode
===

If there already was a header within the sourcecode KCommodity will now go the second way. Again a window will be opened which does not visually differ from the Purpose window. The window‘s title is:

Enter a Log Message

You can enter a descriptive text for all changes made to the source since the last release (Log message) here. The two integer gadgets at the bottom of the window can be used to set the new revision number KCommodity should continue with. KCommodity will automatically increment the revision of the sourcecode. KCommodity will set the integer gadgets to the next revision. Say if the current revision is "1.1", the gadgets will read "1.2". You can now change the revision in case you want to skip several evolutionary steps. Selecting *OK* you tell KCommodity to continue whilst *CANCEL* aborts the procedure.

For additional revisions the "Only text" gadget is available. This gadget tells KCommodity to *add* the Log message to an already existing Log message, that means only text will be added, but no changes will be made to the revision number. Only the revision date will be changed.

Did you tell KCommodity to continue (using "Ok") only important entries will be changed within the header. The actual revision number will be set to the new value and the log message will be inserted before the previously entered. The log message's header gives you information about the author of the revision and when the changes were made. Finally the macros behind the header will be recreated. For further information have a look at
The revision header's special features. The header's postion within the source will be kept unchanged but the source's length will slightly increase. If the first of the six macros could not be found right after the header, KCommodity assumes that you no longer want the program to rebuild the macros and so it does not make a refresh. So if you insert a blank line between header and macros no more changes will be made to them.

1.212 KCommodity/RevSpec

The revision header's special features
======================================

Here is a small example for the look of the header in Assembler.

```
    ;
    ; A maximum of 14 lines above the header
    ;
   ;* $Revision Header built automatically *************** (do not edit) *
   ;*
   ;* © Copyright by <COMPANY>
   ;*
   ;* File             : test
   ;* Created on       : Monday, 22.02.93  17:50:35
   ;* Created by       : <AUTHOR>
   ;* Current revision : V2.5
   ;*
   ;*
   ;* Purpose
   ;* -------
   ;*   - This is a test
   ;*
```

```
;* Revision V2.5
;* --------------
;* created on Monday, 22.02.93  17:50:41  by  <AUTHOR>.   LogMessage :
;*  -*-  changed on Monday, 22.02.93  17:50:57 by <AUTHOR>. LogMessage :
;*   - Additional text for revision 2.5
;*  -*-  created on Monday, 22.02.93  17:50:41 by <AUTHOR>. LogMessage :
;*   - The next revision. This will be 2.5
;*
;* Revision V1.0
;* --------------
;* created on Monday, 22.02.93  17:50:35  by  <AUTHOR>.   LogMessage :
;*     --- Initial release ---
;*
;**********************************************************************
REVISION  MACRO
          dc.b "2.5"
          ENDM
REVDATE   MACRO
          dc.b "22.02.93"
          ENDM
REVTIME   MACRO
          dc.b "17:50:57"
          ENDM
AUTHOR    MACRO
          dc.b "<AUTHOR>"
          ENDM
VERNUM    EQU  2
REVNUM    EQU  5
  ;
  ; This is where your sourcecode would start at
  ;
```

This is the header of an Assembler sourcecode which was revised once after it was created. The field <COMPANY> will be filled with the name of the company (if present). The <AUTHOR> fields will be filled with the name of the author as long as the environment variable could be found. At the end of the header you find six macros. These macros reflect the actual revision number, date of revision, time of revision, the name of the author having done the revision and version/revision number. You can use these macros within your sourcecode, so that you can tell the user something about the program‘s "evolution" during runtime.

The Basic, ‘C’ and Pascal / Modula pendants take the same names as in Assembler. The ‘C’ type macros will be set as *#defines* into the sourcecode. As you might believe the Basic "versions" are string variables which have a *$* postpended.

There is one thing to remember while using the Pascal / Modula macros: To make these macros available within sourcecodes I chose the possibility to declare the macros as *CONST*ants. Therefore the *CONST* symbol will immediatly follow the header and then following are the macros. So you will have to add any other constant declarations *after* KCommodity‘s.

1.213 KCommodity/ARexx

The ARexx Port
****

KCommodity is equipped with an ARexx Port. You can manipulate almost all functions via the ARexx port. The name for the ARexx Port is ‘KComm.1’. So if you want to send commands to that port you would have to issue

```
ADDRESS ’KComm.1’
```

within your ARexx script to get contact with KCommodity. If you changed the name of the port using the PORTNAME ToolType you would have to insert the name you used, of course (see Using ToolTypes, for more).

Please remember that string parameters for KCommodity‘s commands should be enclosed by quotes, because it could come to errors while interpreting the command.

See Commands for the ARexx Port.

1.214 KCommodity/ARexxComs

Commands for the ARexx Port
===========================

Here is a list of commands for the ARexx port.

‘ACTIVATE 1|0’
Turns "Window activate" function on / off.

‘SUNMOUSE 1|0’
Turns "Sun Mouse" on / off.

‘SETENV 1|0’
Turn on / off environment variables.

‘MENUWRAP 1|0’
Using this command, you can turn on/off "MenuWrap"

‘FKEYS 1|0’
If your function key settings conflict with those of an application, you may turn them on/off using this command.

‘ESCCLOSE 1|0’
The "ESC Window Close" function can be turned on / off via ARexx, too.

‘MAPUMLAUT 1|0’
Turn on / off Map "Umlauts" via ARexx.

‘PAGENUM 0-5’

Choice of a ‘page’ which should be displayed within the Workbench titlebar or the status window. The number‘s meaning is as follows:

1. Time display plus mouse coordinates.

2. Display free Chip / Fast memory.

3. Numerically display total free memory.

4. Timedisplay in conjunction with memory display.

5. Display total free memory as a gauge.

6. Online time.

‘SCREENTIME Time’

Set a new time for the Screen Blanker. These values are treated the same way as mentioned under "Screen time". This value is created using the following formula: Time = (Hours*60)+Minutes.

‘MOUSETIME Time’

Use this command to set a new time for the Mouse Blanker. These values are treated the same way as described under "Mouse time". You calculate the Time the same way as described under "SCREENTIME".

‘CYCLING 1|0’

This command is used to turn on / off "Cycling".

‘LOGCALLS 1|0’

The saving of data as described under "Log call" can be turned on / off.

‘BILLPOPUP 1|0’

Turning on / off the "PopUp Bill window" function.

‘TIMEZONE 0-2’

Change the timezone for a call via ARexx. Simply supply the timezone you wish to use, ranging from 0 to 2.

‘DRIVE 0-3’

Select a new drive to be the source for the Formatter.

‘LABEL ’Name’’

Define a new label for disks being formated using KCommodity.

‘TRASHCAN 1|0’

Control the creation of the Trashcan.

‘QUICK 1|0’

Quick formatting may be turned on and off via ARexx.

‘FFS 1|0’

You can choose between OFS and FFS to be used for formatting disks.

‘NOVERIFY 1|0’

If it takes too long to format a disk using Verify you can turn off Verify for the ’Olsen’ Formatter.

‘INSTALL 1|0’
     Since Bootdisks have to be installed you can tell ’Olsen‘s’ Formatter to install the disk after the format.

‘EJECT 1|0’
     Eject the disk after the format.

‘INTERNATIONAL 1|0’
     This flag will set the International Mode flag for formatting.

‘DIRCACHE 1|0’
     Using this command you may turn on directory caching mode for formatting a disk.

‘WATCHDISKS 1|0’
     Tell KCommodity to check any inserted disks, or you may turn off this feature.

‘FORMAT’
     Start to format a disk using the supplied settings.

‘ENABLE’
     Does the same as mentioned under "Enable".

‘DISABLE’
     Same as for "Disable".

‘REVISE’
     Activate the "Revision Control System". Have a look at the corresponding chapter for further information.

1.215 KCommodity/KCXErrors

Error codes
****

‘x could not be opened.’
     This error code tells you that a required library or device could not be opened. Make sure that the desired file is located at the correct place (Libraries within LIBS: and devices within DEVS:). It is best to always use the latest resources.

‘Broker-Port could not be created.’
     The port for communicating with Exchange and for controlling the main functions could not be installed. It might help you to close some windows and to stop several programs, because a lack of memory could be the reason for this error.

‘Broker could not be set up.’
     The program‘s main control structure could not be installed. Again a lack of free memory could be the reason.

‘Broker could not be set up correctly.’

The program`s main interface could not be set up. The difference between this error and the one mentioned above is that the system did not understand a definition within the structure.

'Sorry, memory for InputRequest could not be allocated.'
Memory for the InputRequest could not be allocated. Try to free memory and restart the program.

'Sorry, MessagePort for InputRequest could not be created.'
A Message Port could not be installed. Without this port the InputHandler can not be installed. Again the reason might be lack of memory.

'Sorry, InputHandler could not be installed.'
The main InputHandler could not be installed. It may either be lack of memory or a wrong structure definition. Try to free memory and restart the program.

'Could not set up ARexx-Port.'
The ARexx Port could not be installed. Lack of memory could be the reason.

'Audio-Port could not be set up. ...'
The Message Port for the audio device could not be generated. Therefore the "Key-Click" option is totally disabled now. You can try to restart the program, after you have stopped several programs.

'Sorry, out of memory.'
Simple : Not enough memory to execute the desired function. Try to quit some programs which are not needed.

'Sorry, could not generate BitMap.'
The BitMap structure for the time display within the Workbench titlebar could not be initialized. The reason for this error definitely is lack of memory.

'Sorry, could not generate RastPort.'
Same as above. The RastPort structure for the display within the Workbench titlebar could not be initialize.

'Could not allocate memory for BitPlanes.'
The BitPlanes for the Workbench titlebar display could not be allocated. Make sure that programs running are not consuming too much Chip memory and stop them in this case.

'Could not set appropriate font.'
The font for Workbench titlebar display could not be set. This error should *never* occure due to the fact that the Workbench is already using this font (but one can never know).

'Could not allocate memory for ARexx-Buffer.'
'Could not allocate memory for Audio-Request.'
'Could not allocate memory for Console-Request.'
Again there is not enough memory. Try to free memory and restart the program.

'User HotKey ... could not be set up ...'
One of the user-defined HotKeys for the several functions could not be installed. The program will take the default HotKey now. Check your HotKey definitions.

'Default HotKey ... could not be set up ...'
A default HotKey could not be installed. The attempt failed and the program quits. Try to free memory.

'Sorry, but the AppIcon could not be installed.'
The AppIcon could not be displayed. This can happen if there is not sufficient memory or the Workbench is not present.

'Sorry, but the AppMenuItem could not be installed.'
Same as above, but the AppMenuItem could not be installed this time.

'The Preferences-File you attempted to load...'
You tried to load a Preferences file which was not generated using this version of KCommodity.

'Could not allocate IFF-Handle for access to file.'
KCommodity was not able to allocate a needed structure, in order to obtain access to an IFF-File. Try to free some memory.

'Could not allocate User Task-Signal.'
A needed task signal could not be allocated. Try to start KCommodity again, or make a reset, because this error code should *never* occure.

'Shared IDCMP-Port for windows could not be allocated.'
Because of the number of windows KCommodity controls, there is need for a single port, all IDCMP messages go through. This Port has to be allocated by the application. If you discover this error message, a lack of memory might be the problem.

'Could not allocate Port for trackdisk.device'
The MessagePort for the trackdisk.device could not be allocated. This device serves for checking inserted disks. Try to free some memory.