IFFMaster

An IFF Browser
Version 1.5

March 23, 1996

Kay Drangmeister

This document was created using TEX3.14, '`texinfo.tex`', and '`makeinfo`'.

1 Introduction

IFFMaster is a program that allows you to have a view inside the structure of IFF files. It is *not* yet another picture viewer or sound sample player, but it displays the internal entities (*chunks*) of a file. For example, the headers of pictures (`ILBM`) or sound samples (`8SVX`) are displayed in clear, so you can directly read the size and depth of the image or the sampling rate.

From time to time you will find an unknown file on your hard disk, let's say a preference file. Since preference files are often stored in IFFormat, there is a chance you can directy examine the contens. There are some programs which write incorrect IFF files (e.g. the `ILBM.CAMG` chunk is a likely candidate), or which write additional information (e.g. copyright or author chunks). With IFFMaster you can easily verify such cases.

During the last time I implemented some ways to manipulate files, e.g. delete and move chunks. Chunk contents can be edited using a text or binary editor depending on the content type. In future versions (see Appendix A [History], page 15) there will be options to add new chunks like a `DPI` chunk for images.

Current features include:

- MUI application
- localized GUI (with AmigaOS 2.1 or later)
- online help, bubble help (with MUI 3.0 or later)
- Extensive chunk type library (currently 51 form types, 333 known chunks, 94 of these with comprehensive structure description)
- Chunk contents are presented alternatively as structure, text or hex dump
- Bit fields and enumeration types are displayed in clear
- Fixed point values are printed in decimal (e.g. `8SVX.VHDR.Volume`)
- Callback hooks for special attributes, e.g. the Mode-ID inside the `CAMG` chunk is de-referenced (e.g. '`PAL: Hires`').
- ARexx interface. E.g. extract Term phonebook entries to the DFA data base.

IFFMaster is distributed under the concept of *freeware*. Standard disclaimer applies to this program.

2 Installation

Nothing to it: just leave all files and the catalogs directory as they are in one directory, or alternatively copy the appropriate '`iffmaster.catalog`' to into the system's locale directory (i.e. '`LOCALE:Catalogs/yourlanguage/`'). By the way: if you mixed up catalog files for different languages, just use IFFMaster to discover the catalog's language... `:)`

System requirements are:

- AmigaOS 2.0 (V37)
- AmigaOS 2.1 (V38) for localized GUI
- AmigaOS 3.0 (V39) for some extended features, like displaying colors in `ILBM/CMAP` chunks.
- MUI version 2.1 ('`muimaster.library`' v8), See Section 6.1 [MUI], page 13.

If you wish to start IFFMaster inside the `user-startup`, you should do this with '`runback IFFMaster ICONIFIED`'. This will cause IFFMaster to start as an *AppIcon*, so that Icons may be dragged onto this Icon.

3 Usage

3.1 Main Window

The main window consists of 3 groups. The topmost one is the *file specification* group, the middle part ist the *chunk list* and below there are the *action buttons*.

The current version has some capabilities to manipulate files, but adding chunks is still not implemented. So the '`Add`' button is always inactive. However, moving ('`Up`', '`Down`'), deleting ('`Del`'), and editing ('`Edit`') chunks is possible. Be warned that files may become unreadable to some applications if you delete mandatory chunks, i.e. if you delete the `ILBM/BMHD` chunk of a picture it will become undecodeable. To activate the manipulation buttons select '`Prefs/Editable file`' from menu. If that menu item is inactive (it can be locked if you are still unsure about the program's functions) you can activate it by opening the preferences window and setting the '`File editable switch`' to be '`off for new files`' or to be '`left unchanged`', see Section 3.3.2 [Safety], page 6.

3.1.1 File Specification

There are different ways to open a file. The most common one is by selecting '`Project/Open...`' from the menu, or by clicking on the popup gadget beside the string gadget for getting a file requester. Alternatively, you can simply drag an icon on IFFMaster's main window (*AppWindow*). Furthermore, you can open the file which is currently in the clipboard by using the menu item '`Project/Open Clip`'.

3.1.2 Chunk List

The chunk list can show several types of information: the left column shows the *chunk ID* (e.g. `BMHD`), then there is the *chunk type* (e.g. `ILBM`), and at the right side there is the size of the chunk. Below the action buttons there is a cycle gadget labeled '`Show`' that specifies the format of the chunk list. If it states '`Description`' you will get a description of the chunk's purpose (e.g. '`Bitmap Header`'), and '`Contents`' will display a *short* summary of the chunk's contents. You can advance the cycle gadget by pressing the `SPACE` key.

Because it is impossible to display the entire contents of a chunk in a single line you can get a comprehensive list by first selecting a chunk in the chunk list and then pressing '`Info`' (you can as well double-click on the chunk). A new window containing the chunk's contents will appear. You can display the contents either as structure, plain text or as a hex dump. Change the presentation by using the register above the contents list, or by pressing `CURSOR LEFT` and `CURSOR RIGHT`.

You need not close the contents window to display the contents of another chunk. Just click on that chunk in the chunk list or press `CURSOR UP` or `CURSOR DOWN` after having activated the chunk list via the `TAB` key. The new contents are displayed using the appropriate data type.

3.1.3 Action Buttons

Each of the buttons in this group performs an action on the currently active chunk, i.e. the on that is marked by the cursor in the chunk list.

The buttons that are used to modify a file are inactive if the menu item (switch) '`File Editable`' is turned off. This is for safety reasons, so that the file structure or contents cannot be damaged unitentionally. Depending on the experience of the user there are different locking modes for that switch, see Section 3.3.2 [Safety], page 6.

`Info` The contents window will be opened, and additional information about the chunk is desplayed in it.

`Edit` Depending on the chunk's contents (text or binary data) a text or binary editor is called to let you edit the chunk's contents. See Section 3.3.1 [General], page 6.

`Add` This action is not implemented yet due to extremely low user feedback. So this button is always disabled.

`Delete` The active chunk is entirely deleted.

`Up` The chunk is moved before its predecessor. Because chunks can not be moved out of their containers, this action will only succed if the preceeding chunk is not one of the type `FORM`, `CAT` or `LIST`. Furthermore, only chunks containing data can be moved, i.e. container chunks (type `FORM`, `CAT` or `LIST`) are not moveable.

`Down` The chunk is moved behind its successor. The same restrictions as in '`Up`' apply.

`Top` The chunk is moved up as far as possible. The same restrictions as in '`Up`' apply.

`Bottom` The chunk is moved up as down as possible. The same restrictions as in '`Up`' apply.

3.2 Contents Window

The contents window displays the contents of the active chunk, see Section 3.1.2 [Chunk List], page 3. Depending on the type of the contents one of the following three display types is (automatically) used.

`Structure` This is the most sophisticated method to display the chunk's contents. All parts of the content are decoded and displayed line by line together with a description.

Example: A sampled sound (`FORM 8SVX`) has a header chunk (`VHDR`) that is 20 bytes long. Displayed as a sequence of hex numbers it would read: `00021432 00000000 00000020 41560100 00010000`. Displayed as a structure it reads like this:

```
                OneShot HiSamples  136,242
                 Repeat HiSamples  0
                Samples / HiCycle  32
                      Samples / s  16,726
                        # Octaves  1
           Compression Technique  None
                    Volume [0,1]  1.00000
```

`Text` Some chunks contain plain text, e.g. `ANNO` chunks. Texts are formatted block-justified, but no other processing is done.

`Hex Dump` If a chunk contains neither structured data nor text the contents are displayed using a hex dump. There are some ways to customize the appearence of the dump, see Section 3.3.4 [Hexdump], page 7.

You can also switch between some display styles by hand, using the register gadgets. For example you can have the above mentioned `VHDR` chunk displayed as a hex dump instead of a structure. To switch between the display styles you can also use the `CURSOR LEFT` and `CURSOR RIGHT` keys, even if the active window is the main window and not the contents window. This way you can control all display functions from the main window.

3.3 Preferences Window

The preferences are grouped on four pages: *General*, *Safety*, *Layout* and *Hexdump*. They are described in the following sections.

After having edited the preferences you may choose to

`Save` For storing the settings on disk. All future invocations of IFFMaster will use that settings.

`Use` For storing the settings in RAM only, so they will only last until you reboot your computer.

`Cancel` For aborting the adjustments you made and use the previously stored settings.

Closing the window with the close gadget has the same effect as klicking on '`Cancel`'.

3.3.1 General

On this page you can specify the editors.

`Text Editor` specifies the editor that is used for chunks containing plain text, e.g. `ANNO`-Chunks. You have to make sure that the editor does not spawn itself from the shell, but runs synchronously. If you are using e.g. the CygnusED you may enter '`ed -sticky`'. If you leave this gadget empty the editor specified in the environment variable `EDITOR` is used.

`Binary Editor` specifies the editor that is used for chunks containing binary data. A common name for these editors is *file zapper*, and are likely to be available on the aminet. This editor has also to run synchronously. If you leave this gadget empty the editor specified in the environment variable `BINEDITOR` is used.

3.3.2 Safety

The following settings are used to prevent inintentional damage to files. Since numerous 'okay to do...?' requests by the program are evenly unacceptable you can adjust between the two extremes safety and ease of use. Beginners and casual users should always use the safe settings.

`File Editable switch`

In the menu '`Settings`' there is a switch '`File Editable`' which turns on or off the file manipulation gadgets, see Section 3.1.3 [Action Buttons], page 4. You can set here the behaviour of that switch.

`off and locked` is the safest setting. The switch is off and disabled, effectively preventing you from accidentially turing on the editing buttons.

`off for new files`
: turns off the manipulation gadgets each time a new file is opened. If you want to edit files only infrequently this is a good setting.

`left unchanged`
: leaves the switch always in the state you switched it to. If you want to edit many files in a row you have to turn on the manipulation gadgets only once.

`Overwrite files`
: If you want to edit many files without keeping backups you can turn the prompting off. Otherwise it is recommended to leave this gadget always on the safe '`prompt`' position, because since IFFMaster can be fully driven by keyboard it just needs two keypresses like `DEL` and `S` (for '`Save`') to possibly destroy a file.

3.3.3 Layout

The layout page currently features the following settings:

`Bubble help`
: If you are familiar with the program's functions you may wish to turn off the bubble help. (NB: Bubble help is only supportet in MUI 3.0 or later.)

`Hex indicator`
: lets you choose your favourite pre- or postfix string to indicate hexadeximal numbers in the structure page of the contents window. For an example file which contains hex numbers open '`ENV:Sys/locale.prefs`' and select the `LCLE` chunk. The first 4 numbers (16 Bytes) are reserved and displayed as hex numbers.

  NB: This setting does not affect the hex dump, see Section 3.3.4 [Hexdump], page 7.

3.3.4 Hexdump

This page contains gadgets that affect the appearence of the hex dump in the contents window.

`Limit # bytes`
: Building hex dumps of very large chunks (e.g. `ILBM.BODY`) can take a long time. Therefore the number of bytes in a hex dump can be limited to a certain ammount. The default limit is 512 bytes. Since in most cases hex dumps are not very meaningful you may wish to leave this value reasonably low. To adjust the value

you man use the slider or the string gadget to the right of it. With the check mark button you can turn the limit off, but this is not recommended.

`Offset format` The leftmost column of the hex dump shows the offset of the first byte in each row. Use this gadget to specify if that offset should be displayed in deximal or hexadecimal.

`Dump characters`

If the (fixed-width) font used in the hex dump contains all 256 characters you can turn on to print even normally non-printable characters like LineFeed, `0x0A`, which will then be displayed as an inverse 'J' or something like that. If only a rectangle appears then your font supports only printable characters and you will have to turn on '`printable only`'. Then all non-printable characters are displayed as a dot.

A good way to test this feature is to load a 24 bit ILBM picture. These pictures normally contain three `CLUT` chunks. Normally these chunks contain all byte values in increasing order.

4 ARexx Interface

Since version 1.5 of IFFMaster many functions can be accessed via an ARexx port. This allows you to gain access to internal IFF data from inside scripts. For example you could list all image (`ILBM`) files (or sort them into a directory) whose depth is up to 8 planes. Or you could save some space if you remove all `ANNO` chunks from the files in your sample (`8SVX`) directory. A few example scripts are in the `rexx` directory of the distribution. The port name of IFFMaster is `IFFMASTER.1`. You can get a quick list of IFFMaster's REXX commands by running '`rexx/ShowREXXCommands.rexx`'.

`load FILE/A`
: Loads a file into IFFMaster.

`loadclip`
: Loads the clipboard's contents.

`saveas FILE/A`
: Saves the project under the given file name. You may wish to turn `overwrite ON` to prevent a requester popping up.

`save`
: Saves the project under the same name as displayed in the string gadget, i.e. under the load file name.

`saveclip`
: Copies the project to the clipboard.

`savechunk FILE/A`
: Saves the chunk marked by the cursor under the given file name.

`saveform FILE/A`
: Saves the sub-FORM marked by the cursor under the given file name. The cursor must be over a `FORM`, `LIST` or `CAT` chunk.

`saveformclip`
: Copies the sub-FORM marked by the cursor to the clipboard.

`entries`
: Gives the number of chunk list entries as the `RESULT`.

`cursorpos ENTRY/N`
: Gives the cursor position in the chunk list as the `RESULT`.

`chunkid`
: Gives the chunk ID of the chunk marked by the cursor as the `RESULT`.

`chunktype`
: Gives the chunk type of the chunk marked by the cursor as the `RESULT`.

`chunksize`
: Gives the chunk size of the chunk marked by the cursor as the `RESULT`.

`chunkdepth`
: Gives the chunk' nesting depth of the chunk marked by the cursor as the `RESULT`.

`chunkinfo`
: Gives short information about the chunk marked by the cursor as the `RESULT`. The information is the same as the 'contents' column in the chunk list.

`chunkstruct SEPSTRING`
: Gives the information as in the 'structure' page of the chunk contents window on a line-by-line basis. Since the list has two columns (description and contents) you can specify a separator string (`SEPSTRING`) for the two entries. The default `SEPSTRING` is `" | "`. See '`rexx/ImportTERMpbook.rexx`' for an example usage.

`chunktext`
: Gives the chunk's contents as a string in `RESULT`. This makes only sense if the chunk is really a text chunk, like `ANNO` or `AUTH`.

`delete`
: Removes the chunk under the cursor from the chunk list. Note that you have to turn `editable ON` to successfully perform a chunk deletion. See '`rexx/RemoveANNO.rexx`' for an example usage.

`infowindow ON/S,OFF/S`
: If you have a non-interactive REXX script which does not require user supervision you may close (`OFF`) open (`ON`) the chunk contents window.

`mainwindow ON/S,OFF/S`
: Same as above, but closes the main window. Note to turn the window back on or quit IFFMaster if your script terminates, otherwise IFFMaster will stay open with no user interface to quit it. Of course the user can quit IFFMaster by hand by using CommoditiesExchange or by sending a break signal to IFFMaster's task or by using the `quit` REXX command.

`editable ON/S,OFF/S`
: Turns the editing facilities on or off.

`overwrite ON/S,OFF/S`
: Turns quiet overwriting of files on or off.

`showformat FORMAT/A/N`
: Changes the state of the '`Show`' cycle gadget that affects the format of the chunk list. Following values are possible:

  0. '`Id - Type - Size`'
  1. '`Id - Type - Description`'
  2. '`Id - Type - Description - Size`'
  3. '`Id - Type - Contents`'

4. 'Id - Type - Contents - Size'

`limithex ON/S,OFF/S,BYTES/N`
: Specify the size limit of the hex dump buffer. `OFF` turns limitation off, `ON` limits size to value set in the preferences, `BYTES/N` changes the preferences to the specified value. Use `limithex BYTES 16` if you don't need to display hex dumps and want to gain speed when processing chunk contents (i.e. moving the cursor).

`editor EDITORNAME`
: Specifies the editor to use when editing text chunks.

`bineditor BINEDITORNAME`
: Specifies the editor to use when editing binary chunks.

There are some standard MUI REXX commands that are also useful:

`quit FORCE/S`
: Terminates IFFMaster. If you specify `FORCE` then no requester pops up when there is an unsaved project.

`hide`
: Iconifies IFFMaster.

`show`
: Uniconifies IFFMaster.

`info ITEM/A`
: Gives information about variuos items (e.g. title). Refer to '`MUI.guide/ARexx`'.

`help FILE/A`
: Lists all REXX commands to '`FILE`'. See '`rexx/ShowREXXCommands.rexx`' for an example usage.

5 Problems?

Building hex dumps of very large chunks (e.g. `ILBM.BODY`) can take a long time. Therefore the number of bytes in a hex dump can be limited to a certain ammount. The default limit is 512 bytes, it can be adjusted or turned off in the Preferences, see Section 3.3.4 [Hexdump], page 7.

Some chunks (e.g. `FORM`) are container chunks with no contents, and therefore there is no presentation. Some other chunks (e.g. `BODY`) contain raw data, presentation of these chunks is limited to hex dump.

6 Credits

I like to thank:

Stefan Stuntz	for MUI, see Section 6.1 [MUI], page 13.
Eric Totel	for his great MUIBuilder, and for keeping MUIBuilder up with the needs of IFFMaster :). This program would not exist without it.
Francesco Dipietromaria	for the italian translation
Marcin Orîowski	for the polish translation
Alessandro Zummo	for fixing the de-iconify problem with the AppIcon
Andreas Mixich	for all ARexx scripts
Kai Iske and Walter Dörwald	for support and betatesting
Thomas Reinhardt, Harald Drangmeister and Ralph Wermke	for their IFF descriptions
H. Phil Duby, Bryan Ewert, Martin Pfingstl and Klaus Seistrup	for bug reports, support and comments

6.1 MUI

```
                    This application uses

                MUI - MagicUserInterface

            (c) Copyright 1993/94 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With
the  aid  of  a  preferences program, the user of an application has the
ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing
lots of examples and more information about registration please look for
a  file  called  "muiXXusr.lha"  (XX means the latest version number) on
your local bulletin boards or on public domain disks.

          If you want to register directly, feel free to send

                DM 30.-  or  US$ 20.-
```

to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

Appendix A History

Version 1.0
: First release

Version 1.1

FIX version 1.0 was compiled with option `68030` - sorry!

NEW many new chunk types

NEW menu (moved '`about`' and '`prefs`' into the menu, gaining some space)

NEW specifying file names at command line

NEW loading files directly from clipboard

NEW improved keyboard handling

Version 1.2

NEW fully localized GUI

NEW chunk contents can be saved

NEW AppWindow and AppIcon, command line option `iconified`

NEW improved formatting capabilities for structure entries

NEW sophisticated IFF parser, even brain-dead formats like `EMOD.EMIC` are handeled appropriately

NEW chunks now have nominal length, a message is displayed if a chunk is too short or too long

BUG Beta testers reported problems with de-iconify function when an icon was dropped onto the AppIcon. I removed the said call, so you have to double-click the AppIcon after dropping an icon.

Version 1.3

FIX prefs window: close gadget now functioning, see Section 3.3 [Preferences Window], page 5.

FIX cycle chain (`TAB`) now works in all windows

NEW new chunk IDs: `IAND`, `IANM`, `DR2D`, `RGB8`, `RGBN`, `SPLT`

NEW hex numbers get a user-chosen indicator, construction of hex dumps is now more than 3 times faster, see Section 3.3.4 [Hexdump], page 7.

NEW IFFs can be saved, even nested `FORM`s, e.g. images inside animations

NEW implemented some editing capabilities (delete chunks), see Section 3.1.3 [Action Buttons], page 4.

NEW preferences can be saved, see Section 3.3 [Preferences Window], page 5.

Version 1.4

NEW Completely changed to GNU-C. Differences are: new startup code, no more ANSI-C functions like `sprintf()` used, enabling some things like localized output of floating point numbers.

FIX should finally run without '`locale.library`'

NEW Online help

NEW new chunk IDs: `AMBA`, `AMDE`, `AMIN`, `AMHU`, `AMUN`, `EQE1`, `RESO`, `VARS`, ...

NEW complete structure descriptions of system preference files, some new hooks. Be sure to try out e.g. '`IFFMaster ENV:sys/locale.prefs`'.

NEW color descriptions (RGB values) are now also displayed as a colorfield, if you have WB 3.0. Check out '`IFFMaster ENV:sys/palette.prefs`' or any `ILBM/CMAP`. NOTE: You need some free pens for this feature, so you may need to specify a deep screen for IFFMaster inside MUIPrefs.

FIX '`<Clipboard>`' is now accepted as a name for the clipboard. So '`Save`' now works as '`Save Clip`' when working on a clipboard file. As a side-effect you can now enter '`IFFMaster "<Clipboard>"`' at the command line to load directly from the clipboard.

FIX button key definitions did overlap

NEW 'non-printable' characters can now be displayed in the hex dump, if the font comprises 256 characters, see Section 3.3.4 [Hexdump], page 7. (Requested by Walter Dörwald)

NEW some more editing capabilities (move chunks), see Section 3.1.3 [Action Buttons], page 4.

NEW Chunks containing text or binary data can be edited via text or binary editors, see Section 3.1.3 [Action Buttons], page 4.

NEW new icon, which is also used as AppIcon now

Version 1.5

NEW ARexx interface. E.g. extract Term phonebook entries to the DFA data base.

NEW Bubble help available (MUI 3.0).

FIX '`File editable`' switch now also enables/disables '`edit`' button.

NEW New Chunks: MUI Prefs,

NEW Support for IEEE 96 bit floats (`ILBM.MAND`)

FIX Key `O` works again (activates string gadget)

FIX AppIcon problem hopefully solved (thanks to Alessandro Zummo)

I would greatly appreciate new ideas and enhancement requests. If you discover any chunk unknown to IFFMaster, please let me know (i.e. try to send me that (short) file or even a description of the chunk's structure, if known). If you dare to translate the catalog to your language, please have a look at '`translators.readme`' first.

Appendix B Chunks

Following chunks are currently known to IFFMaster:

[anywhere]
: CSET, FVER, ANNO, AUTH, CHRS, HLID, NAME, TEXT, (c)

3DDD
: OBJE

8SVX
: ATAK, BODY, CHAN, FADE, PAN , RLSE, SEQN, VHDR

ANIM
: ANSQ

AVCF
: AVFH, GDAT

AVCO
: CDAT, FLAG, IMAG

AVEV
: ACTS, CDAT, FLAG, IMAG, PARS, REFL

CDAF
: BODY, FILE

CDVR
: VARS

COPR
: COPI, WAIT, MOVE

CPRO
: 3DIM, GFRA, SFRA

CTLG
: LANG, STRS

CYBR
: CMON, MD08, MD16, MD24, VER

DECK
: RESO

DEEP
: DBOD, DGBL, DLOC, DPEL

DR2D
: ATTR, CMAP, CPLY, DASH, DRHD, FONS, OPLY, PPRF

DTYP
: DTHD, DTCD

EMOD
: 8SMP, EMIC, PATT

ENVL
: LFOI, COEN, VOEN, PIEN, NOEN, REEN, PHEN

EQED
: EQE1

FAX3
: FXHD, GPHD, PAGE

FAXX
: FXHD, GPHD, PAGE

FTXT
: CYGD, FONS

GXGA
: GADA

GXMN
: MEDA

GXUI GGUI

GXWD WDDA

IAND BPCT, CMAP, BODY

IANM BMHD, CAMG, BODY

ILBM 3DIM, ANHD, ASDG, BHCP, BHSI, BMHD, BODY, CAMG, CCRT, CLUT, CMAP, COLC, CRNG, DEST, DLTA, DMMY, DPAN, DPI , DPPS, DPPV, DRNG, EQE1, FCMP, FFEX, FITR, FXCO, FXD2, FXPL, GFRA, GRAB, IMRT, JUNK, MAND, SFRA, SPRT

ISTG MAXF, SOBJ

KCXM VERS, PREF

LWOB PNTS, POLS, SRFS, SURF

MCXB PREF, VERS

MCXP PREF, VERS

MTRX ARRY, BODY, DTYP, STRU

NAIL NBDY, NDSC, NHDR

PREF ALRT, AMBA, AMDE, AMIN, AMHU, AMUN, CONF, CTRY, DFSS, EVNT, ETXT, FLOP, FONT, GENA, GENC, GTCO, GUI, ICTL, INPT, JFIF, KEYS, LCLE, MENU, MIDI, MUIC, MUIW, OPER, OSCN, PALT, PATH, PDAT, PGFX, PNTR, PRHD, PSPD, PTRN, PTXT, PUNT, SCRM, SERL, SHMN, SOND, TMAC, TMDO, TMEX, TMIC, TMIM, TMMO, TMSO, VERS, WBCF, WBPC, XDOS

PTCH INPF, OUTF, PSEQ, VERS

REAL RANI, RATT, RMTR, ROBJ, RSCR, RSET, RVRS, RWIN

RGB8 BMHD, BODY, CAMG, CMAP, IMRT

RGBN BMHD, BODY, CAMG, CMAP, IMRT

SC3D EDGE, FACE, HIER, LAMP, LNAM, OBSV, PATH, VERT, VNAM, WRLD

SMUS SHDR, INS1, INST, SNX1, TRAK

SPLT INFO, BODY

SWRT ASCI, ATTR, BIBD, BIBH, BIBP, BOXP, DINF, DOC , DSP2, ENDP, FDTA, FNTD, FNTH, GINF, GRMR, HFDA, HYPH, IDXD, IDXH, IDXP, LINP, LMST, OUTD, OUTH, OUTP, OVLP, PAG3, PASD, PASH, PICP, PNDA, PRGH, PRN3, RMST, RULE, SAVP, SEC1, SHD1, SHPP, SPEL, SWCL, TABS, TBDY, TBLK, TOCD, TOCH, TOCP, TOID, TOIH, TOIP, TSSD, TSSH, TXOB

TACF TPAR, TPBR, TPCA, TPCM, TPMA, TPP1, TPPA, TPPX, TPSC, TPSE, VERS

TAKE TFRM, THDR

TDDD INFO, OBJ

TERM CLIP, COMD, CPTR, DATE, DIAL, EMLN, FAST, FILE, MISC, MODM, PATH, PHON, RECV, SCRN, SEND, SERL, SOUN, SPEK, TRML, TRNS, VERS, WINF, WIND, XFER

TVP2 TVRX

VILL CRC , MODE, MONI, VER

I am still looking for descriptions for the chunks in smaller print.

Appendix C Author

Kay Drangmeister
Tannenfelsstraße 17
36115 Hilders
Germany
T. ++49-6681-7566

K.Drangmeister@insider.sub.de

PGP Fingerprint = 5A 88 30 0F BF D7 0F F3 F9 31 A0 88 AB E4 38 66

Index

8

8SVX.VHDR.Volume ... 1

A

action buttons ... 3
Action Buttons ... 4
Add ... 3, 4
altering files ... 3
AppIcon ... 2
AppWindow ... 3
ARexx Interface ... 9
ARexx/bineditor ... 11
ARexx/chunkdepth ... 10
ARexx/chunkid ... 9
ARexx/chunkinfo ... 10
ARexx/chunksize ... 10
ARexx/chunkstruct ... 10
ARexx/chunktext ... 10
ARexx/chunktype ... 9
ARexx/cursorpos ... 9
ARexx/delete ... 10
ARexx/editable ... 10
ARexx/editor ... 11
ARexx/entries ... 9
ARexx/help ... 11
ARexx/hide ... 11
ARexx/info ... 11
ARexx/infowindow ... 10
ARexx/limithex ... 11
ARexx/load ... 9
ARexx/loadclip ... 9
ARexx/mainwindow ... 10
ARexx/overwrite ... 10
ARexx/quit ... 11
ARexx/save ... 9
ARexx/saveas ... 9
ARexx/savechunk ... 9
ARexx/saveclip ... 9
ARexx/saveform ... 9
ARexx/saveformclip ... 9
ARexx/show ... 11
ARexx/showformat ... 10
Author ... 21

B

bineditor (ARexx) ... 11
Bottom ... 4
Bubble help ... 7

C

Chunk ... 1
Chunk contents as hex dump ... 5
Chunk contents as structure ... 5
Chunk contents as text ... 5
chunk ID ... 3
Chunk List ... 3
chunk list format ... 3
chunk liste ... 3
chunk type ... 3
chunkdepth (ARexx) ... 10
chunkid (ARexx) ... 9
chunkinfo (ARexx) ... 10
Chunks known ... 18
Chunks without contents ... 12
chunksize (ARexx) ... 10
chunkstruct (ARexx) ... 10
chunktext (ARexx) ... 10
chunktype (ARexx) ... 9
Container chunks ... 12
contents of chunk ... 3
Contents Window ... 5
Credits ... 13
cursor left ... 5
cursor right ... 5
cursorpos (ARexx) ... 9

D

Del ... 3

Delete 4
delete (ARexx) 10
description 3
Down 3, 4

E

Edit 3, 4
editable (ARexx) 10
editor (ARexx) 11
EMOD.EMIC 15
entries (ARexx) 9
ENV:Sys/locale.prefs 7
Environment variable: BINEDITOR 6
Environment variable: EDITOR 6

F

File editable switch 3
file manipulation 3
file specification 3
File Specification 3
file zapper 6
format of chunk list 3

G

General Preferences 6

H

help (ARexx) 11
hex dump 12
Hexdump Preferences 7
hide (ARexx) 11
History 15

I

iconified 2
ILBM.BODY 7
ILBM.CAMG.ModeID 1
ILBM.CLUT 8
Info 4
info (ARexx) 11
infowindow (ARexx) 10
Installation 2
Introduction 1

L

Layout Preferences 7
limithex (ARexx) 11
list format 3
load (ARexx) 9
loadclip (ARexx) 9

M

Main Window 3
mainwindow (ARexx) 10
manipulation of files 3
MUI 13

O

overwrite (ARexx) 10

P

PREF.LCLE 7
Preferences 12
Preferences Window 5
Preferences: General 6
Preferences: Hexdump 7
Preferences: Layout 7
Preferences: Safety 6
Prefs/Editable file 3
presentation of chunk contents 4
Problems? 12

Q

quit (ARexx) 11

S

Safety Preferences 6
save (ARexx) 9
saveas (ARexx) 9
savechunk (ARexx) 9
saveclip (ARexx) 9
saveform (ARexx) 9
saveformclip (ARexx) 9
Show 3
show (ARexx) 11
showformat (ARexx) 10

T

TAB 4
Top 4

U

Up 3, 4
Usage 3

Table of Contents

1 Introduction **1**

2 Installation **2**

3 Usage **3**

3.1 Main Window 3

3.1.1 File Specification 3

3.1.2 Chunk List 3

3.1.3 Action Buttons 4

3.2 Contents Window 5

3.3 Preferences Window 5

3.3.1 General 6

3.3.2 Safety 6

3.3.3 Layout 7

3.3.4 Hexdump 7

4 ARexx Interface **9**

5 Problems? **12**

6 Credits **13**

6.1 MUI 13

Appendix A History **15**

Appendix B Chunks **18**

Appendix C Author **21**

Index **22**