

ObjectView

COLLABORATORS

	<i>TITLE :</i> ObjectView	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		July 20, 2024
<i>SIGNATURE</i>		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ObjectView	1
1.1	ObjectView V2.10	1
1.2	Introduction - What is ObjectView?	2
1.3	Disclaimer und Copyright	2
1.4	Requirements - What is needed to run the program?	3
1.5	XPK - The Unpack Option	4
1.6	Searching - The Search Function	4
1.7	OVPrefs - The Configuration Editor	5
1.8	The arguments and their meanings	6
1.9	Argument: FILE	7
1.10	argument: NOXPK	7
1.11	argument: SCREEN	7
1.12	argument: WINDOW	7
1.13	argument: PUBSCREEN	8
1.14	argument: MAXWIDTH/MAXHEIGHT	8
1.15	argument: SCREENMODE/TEXTDEPTH	8
1.16	argument - LOOP/IMMEDIATE	9
1.17	argument: FONTNAME/FONTSIZE	9
1.18	Usage - Keys, Menus, Gadgets	10
1.19	General Usage	10
1.20	Usage - Samples/Animations	10
1.21	Usage - Documents	10
1.22	Usage - ObjectView Options	11
1.23	Usage - Keymapping	11
1.24	Usage - Cursorkeys/Numeric keypad	12
1.25	Usage - Menus	13
1.26	Author + Support	14
1.27	Thank yous to the specialists	15
1.28	History	16

Chapter 1

ObjectView

1.1 ObjectView V2.10

ObjectView V2.10

(C)opyright 1995-96 by Matthias Kraft

Introduction What is ObjectView?

Disclaimer More or less important

Requirements What do you need?

XPK-Support The Unpack Option...

Search Function Only for Texts...

Prefs-Editor The Configurations Editor...

ToolTypes & CLI-Arguments Various Presets

Tooltypes CLI-Arguments

>X< FILE/A

>X< DONTUSEXPK=TRUE NOXPK/S

>X< SCREEN=TRUE SCREEN/S

>X< WINDOW=TRUE WINDOW/S

>X< PUBSCREEN=<Name> PUBSCREEN/K

>X< MAXWIDTH=<With> MAXWIDTH/K/N

>X< MAXHEIGHT=<height> MAXHEIGHT/K/N

>X< SCREENMODE=<ModeID> SCREENMODE/K

>X< TEXTDEPTH=<Depth> TEXTDEPTH/K/N

>X< LOOP=TRUE LOOP/S

>X< IMMEDIATE=TRUE IMMEDIATE/S

>X< FONTNAME=<Name.font> FONTNAME/K

>X< FONTSIZE=<YSize> FONTSIZE/K/N

Usage Keys, Menus, Gadgets...

>X< Basic Usage

- >X< Handling of Samples/Animations
- >X< Handling of Documents
- >X< Handling of ObjectView Options
- >X< Preset Keyboard Settings
- >X< Scroll via the Cursorkeys/Numeric Keypad
- >X< Menus
- How to contact the author Me, myself and I
- Credits To the specialists...
- History The way up until now...

1.2 Introduction - What is ObjectView?

Objectview is a file viewer that makes use of the datatypes.

The main features in short:

- + Supports all datatypes
- + You can use this program either with keyboard or menu
- + Supports Workbench and/or CLI parameters
- + Save as IFF
- + Printing
- + XPK and Powerpacker support
- + Search function for texts
- + Very configurable (every object can have its own configuration)
- + Preferences editor so you can easily make a configuration
- + Localized

The installation is best done with the included installer's manual. For people who really want to do it themselves:

- ObjectView + Icon anywhere
- ObjectView.guide + ObjectView.prefs too
- OVPrefs + Icon to Sys:Prefs/
- ObjectView.catalog + OVPrefs.catalog to Locale:Catalogs/[Sprache]/

1.3 Disclaimer und Copyright

ObjectView is (C)opyrighted 1995-96 by Matthias Kraft. All rights reserved!

ObjectView is freeware and may be redistributed in any way, as long as no money (or any equivalent) for its distribution is charged, and may only be distributed in its entire archive form.

ObjectView may be put on a CD, if this CD contains only Public Domain software and is sold at a reasonable price (i.e Aminet CD).

ObjectView may be distributed in free-of-charge software networks (i.e. FRAS-Network and Aminet).

If ObjectView should be part of a commercial product release, or if a magazine wants to use/publish it, they first have to get my written permission to do so.

ObjectView has been tested, and no bugs are known. But they still may exist.

I am not responsible for any damage resulting directly or indirectly from the use of ObjectView. You use this program at your own risk!

If you find a bug that is caused by ObjectView, please tell me and enclose an exact description of how the bug was set off, so I can remove the bug.

All .info-files are NewIcons.

The File 'ObjectView.020' is a for the 68020 optimized version. If you have a 68020 or better, you should use this version!

Additional (C)opyrights:

+++++

XPK is copyrighted by its authors. For further questions refer to umueller@amiga.physik.unizh.ch (Urban Dominik Mnlter) [up to date?]

+++++

ReqTools - reqtools.library

(c) 1991-1994 Nico Fran_ois

1995-1996 Magnus Holmgren

As of release 2.3, the programming is made by:

cmh@lls.se (Magnus Holmgren)

+++++

Triton - An object oriented GUI creation system.

(c) 1993-1995 Stefan Zeiger

szeiger@laren.rhein-main.de

+++++

1.4 Requirements - What is needed to run the program?

Needed is:

-an Amiga with OS 3.x and installed datatypes

-enough free RAM (for pictures or unpacking routines)

-the following libraries must reside in LIBS:

asl.library V38+, datatypes.library V39+

-the icon.library V39+ is used to read the Object-Icons (everyone should have this installed)

-the xpkmaster.library V2+ is being used for unpacking, if it is not present,

the program just can not unpack xpk-packed files
-the reqtools.library V38+ is currently only used for the search function,
if it is not present, you simply can't search texts.
-the triton.library V5+ is used by OVPrefs!
-the locale.library V39+ is used for localization.

1.5 XPK - The Unpack Option

How it works:

If ObjectView identifies an object packed with XPK or PP20 (Powerpacker), the xpkmaster.library is told to unpack the file to T:Objectname. This unpacked file can be deleted after viewing. A possible ending with .xpk or .pp will be deleted so that the datatypes can recognize the file (if required) at its suffix (i.e. the c.datatype can only recognize c files that way)

Besides that you should ensure that you either have enough RAM (if T: points to ram:t) or enough space on your harddisk (if T: points to something like dh0:T). There is an xpk.datatype that can do the unpacking, but AmigaGuide+-documents have problems with it, and the internal routines are quicker, so you should leave that option selected.

1.6 Searching - The Search Function

The search function can be pulled down from the menu items Special/Search and Special/Next.

It has a few limitations though:

At the moment only the c.datatype is supported correctly. The c.datatype is currently the only datatype that uses the line-structure fully, which is required to be able to search.

You can also do a search in pure ASCII texts (and things will be found :), but as the ascii.datatype does not reveal any position data into the line-structure, it will try to determine the position of string with the help of a workaround (which I better not explain in detail :). Occasionally it can happen, that the position to which it jumped is incorrect.

The exe.datatype tries by all means to avoid revealing the position of the line. It seems to fill in the position, but incorrectly, so that here a workaround would also be required. Even though the author of the datatype has mentioned to fix this, I don't see a need for it.

How it works:

A selection of the Menu Special/Search will set off the input requester of

the reqtools.library. You are asked to enter a search pattern, that means for example, if you want the word icon to be found in the text, you would have to enter icon . This is case insensitive. More information about pattern-matching is given in the AmigaDOS Manual where the AmigaDOS special characters/pattern-matching is described.

While searching, the window title will change to Searching... and the mouse-pointer will show up as a waiting mousepointer. When the icon is found, the line will be inverted and that line in the viewable area will be displayed.

If nothing is found, the normal mousepointer is displayed, the normal window title appears and the screen will blink once.

A selection of the menu Special/Next will search for the next appearance of the word that was searched for perviously.

1.7 OVPrefs - The Configuration Editor

OVPrefs V1.4

(C)opyright 1996 by Matthias Kraft

OVPrefs requires the triton.library V5+ in the LIBS: drawer!

OVPrefs is needed to create a configuration of ObjectView. It can handle the global settings as well as the settings of the icons.

Start-up will read the global configuration. The surface will reflect these settings. After the alteration of the settings, you can save them with the buttons 'Save' or 'Use'. The settings saved with the button 'Use' are only valid as long as the computer is not reset.

When dropping the icon into the preferences window, the configuration will be read by that icon. The surface is set accordingly and reflects the settings. After changing the settings you can save them with the menuitem 'Save Icon...'. A file requester will pop up with the path and the filename of the icon (without the .info !!!). After choosing 'OK' the icon will be saved. If no icon exists, one will be created.

The various possibilites of the settings will be explained **later**.

I only want to briefly explain the menu and the setting for 'Screentypes' here.

The Project Menu

->Open Global : Reloads the global settings.

->Save Global : Saves the current settings as global settings

->Open Icon : Opens a file requester and will ask for a filename, whose icon will be loaded.

Beware: the file cannot have the suffix >.info< !!!

->Save Icon : Opens a file requester to check the correct spelling of the

name and saves the icon after selecting the 'OK' button with the current settings.

->About : Shows a little information requester.

->Quit : You have a guess?! This will let OVPrefs water the plants, clean up the kitchen, after that the laundry will be done and the author receives 10DM on his banking account ;-) Well, it would be nice if it happend :)

Screen Mode Settings:

On the left side you can set the position of the show-window, on the right side you have the following setting options

+ use own screen

-> select screen mode via the screen mode requester and select depth.

+ use current public screen

-> no extra settings needed

+ use named public screen

-> you need the name for the public screen you want to use (watch for your spelling!)

1.8 The arguments and their meanings

With the options you can, within a few limits, set where and how an object will be displayed. I recommend using OVPrefs and set the settings for it in the icon of the object. There is however a global settings file in ENV:ObjectView.prefs, which can also be created with OVPrefs. The keywords in the global settings are the same as in the tool types.

The icons of the objects will only be read if ObjectView was started from Workbench. In case ObjectView is executed from CLI (or a script), the global setting will be read once, and will be overwritten with the parameters you have given in the CLI. All objects will then be displayed with those settings.

Explanations to the different arguments:

>X< FILE

>X< DONTUSEXPk

>X< SCREEN

>X< WINDOW

>X< PUBSCREEN

>X< MAXWIDTH / MAXHEIGHT

>X< SCREENMODE / TEXTDEPTH

>X< LOOP / IMMEDIATE

>X< FONTNAME / FONTSIZE

1.9 Argument: FILE

CLI:

FILE/A <name>

If ObjectView is executed from the CLI, you have to enter the filename (with the full path). If no file was found, an error message will be come up and ObjectView will quit.

1.10 argument: NOXPk

OVPrefs:

XPk/PP on/off

Tooltpe:

DONTUSEXPk=(TRUE|FALSE)

CLI:

NX=NOXPk/S

If you have the xpk.datatype installed, you can switch the XPk-Support on or off.

Preset: if the xpk.library is present, the internal XPk-Support will be used.

1.11 argument: SCREEN

OVPrefs:

use own screen (type and depth can be set here too)

Tooltpe:

SCREEN=(TRUE|FALSE)

CLI:

S=SCREEN/S

Has to be set, if you want to view texts in an own screen.

1.12 argument: WINDOW

OVPrefs:

use current public screen

Tooltpe:

WINDOW=(TRUE|FALSE)

CLI:

W=WINDOW/S

If you want to display pictures on the current public screen (which is nearly always the Workbench Screen)

Preset: the pictures and animations are (as far as memory allows it) displayed on an own screen. Texts, amigaguide-files, sound etc., is shown in a window on the current public screen.

1.13 argument: PUBSCREEN

OVPrefs:

use known pubscreen (name must be given)

Tooltype:

PUBSCREEN=<name>

CLI:

PS=PUBSCREEN/K <name>

If everything should be displayed on one public screen (i.e DOPUS.1) <name>

will be replaced by the public screens` name. It is case sensitive.

1.14 argument: MAXWIDTH/MAXHEIGHT

OVPrefs:

width

height

Tooltype:

MAXWIDTH=<width>

MAXHEIGHT=<height>

CLI:

X=MAXWIDTH/K/N <width>

Y=MAXHEIGHT/K/N <height>

You can set the dimensions of a window for objects that have no dimensions.

Objects without dimension are for example ASCII-texts.

If pictures are too big, and it is not possible to open a screen with the size needed, a screen with the given parameters will open.

You can then scroll around the picture with the cursor keys.

MAXWIDTH=640 und MAXHEIGHT=400 are preset. These are the dimensions for a standard NTSC Hires Lace Mode.

1.15 argument: SCREENMODE/TEXTDEPTH

OVPrefs:

Requirements: `use own screen` is selected

Screen mode only selectable with requester

Depth.

Tooltypes:

SCREENMODE=<modeid>

TEXTDEPTH=<depth>

CLI:

SM=SCREENMODE/K <modeid>

TD=TEXTDEPTH/K/N <depth>

You can set the screen mode for texts and documents and the depth of the screen (number of colours). It is ok, if you have set depth to 1 (TEXTDEPTH=1) since you only need 2 colours. The screen mode id must be given hexadecimals (i.e. NTSC Hires Lace => 19004). The size of the screen will be taken from MAXWIDTH and MAXHEIGHT.

If no argument is given, BestModeID() is used to obtain the best screen mode, and a depth of 3 will be used (8 colours).

1.16 argument - LOOP/IMMEDIATE

OVPrefs:

Repeat on/off

Immediate on/off

Tooltypes:

LOOP=(TRUEIFFALSE)

IMMEDIATE=(TRUEIFFALSE)

CLI:

LOOP/S

IM=IMMEDIATE/S

These options are actually designed for sounds and animations.

With IMMEDIATE the sound or the animation is played right after it was loaded. With LOOP you can preset, if it should be repeated endlessly. Unfortunately neither the sound nor the animations.datatype supports this.

Preset: both options are set to false.

1.17 argument: FONTNAME/FONTSIZE

OVPrefs:

Select font for texts (recommended by requester)

Tooltypes:

FONTNAME=<name>

FONTSIZE=<ysize>

CLI:

FN=FONTNAME <name>

FS=FONTSIZE <ysize>

With these two options, you can set the type of font, in which texts is displayed.

You always have to give both options! <name> must always end with the suffix <.font>.

Preset: the system font is used.

1.18 Usage - Keys, Menus, Gadgets

Use with keys or menu. At the moment the only gadgets used are the system gadgets.

>X< Main Usage

>X< Usage of Samples/Animations

>X< Usage of Documents

>X< Usage of ObjectView-Options

>X< Keys

>X< Scrolling via Cursorkeys/Numeric Pad

>X< Menus

1.19 General Usage

The most important first: How do I get rid of the program again? ;-)

By pressing the keys 'q','Q' or <ESC>, if displayed at the close gadget at the top left or, if you can select it, choose the menu item Project/Quit.

The datatypes support only a few of the possibilities, of course! Why would you want to play a picture, or want to go to the next page with a sample? ;-)

1.20 Usage - Samples/Animations

The 'Band' Menu makes a few options available for samples and animations.

Unfortunately the datatypes currently only support one option: 'Play'.

It is also possible to load samples and animations in such a way, that they will be played instantly. See Arguments/IMMEDIATE.

1.21 Usage - Documents

Searching in documents is currently not possible.

The menu item Navigation/Show Contents and Navigation/Show Keywords is especially designed for documents, as they nearly always contain a contents-index, and quite often also an index of keywords.

When changing from Screen->Window or vice versa, the first page of the document will be shown.

1.22 Usage - ObjectView Options

See also the explanation of the Menus.

ObjectView provides an information requester, in which detailed information about the object is displayed. ObjectView also supports printing and saving.

Not every datatype provides the necessary functions to do that, i.e everything is saved in IFF format, but there are datatypes, that

either can't save or do it in an own format. ObjectView lets you search in texts. Big objects can be moved around with the cursor keys or the number pad so they can be seen entirely in a window.

1.23 Usage - Keymapping

Main:

Keys: l, L, 0

-load new object

Keys: d, D

-print indicated object (i.e. a picture or a text)

Keys: w, W

-saves the object as IFF file (samples as 8SVX, pictures as ILBM, texts as FTXT)

Keys: a, A

-show about requester

Keys: i, I

-shows requester with information about the loaded object

Keys: h, H

-shows a help requester with the keymapping

Keys: q, Q, <ESC>

-quit ObjectView

Samples/Animations:

Key: <Space>

-plays the object (i.e Samples or Animations)

Keys: f, F

-"winds" the object forward (is not yet supported by the datatypes)

Keys: p, P

-pauses an object started with <Space> (is not yet supported by the datatypes)

Keys: r, R

-"winds" an object backwards (is not yet supported by the datatypes)

Keys: s, S

-stops an object started with <Space> (is not yet supported by the datatypes)

Documents:

Key: <Tab>

-selects next field (i.e. with AmigaGuide+)

Key: <Shift>+<Tab>

-selects the previous field

Key: <Enter>

-activates selected field

Key: \

-goes one step back

Key: <

-goes to previous page

Key: >

-goes to next page

Keys: c, C

-shows the contents of an object

Keys: x, X

-shows index of an object.

1.24 Usage - Cursorkeys/Numeric keypad

If objects are too big, you can move them in the window:

On the numeric keypad:

1 (End)

-jumps to the last line of an object.

2

-jumps a line below

3 (Pg Dn)

-jumps a page forward

4

-jumps a row left

6

-jumps a row right

7 (Home)

-jumps to first line of the object

8

-jumps to a preceding line

9 (Pg Up)

-jumps a page back

With the cursorkeys:

Cursor Up

-jumps to the previous 10 lines

Cursor Down

-jumps to the succeeding 10 lines

Cursor Left

-jumps 10 rows left

Cursor Right

-jumps 10 rows right

(if it can't move 10 lines or rows, the object will only be moved by 1 line or 1 row)

1.25 Usage - Menus

Project:

Open...

-load a new object

Print

-prints the displayed object (i.e picture or text)

Save as...

-saves the object as IFF (samples as 8SVX, pictures as ILBM, texts as FTXT)

About...

-shows a requester with information about the program

Information...

-shows a requester with information about the loaded object

Help...

-shows a help requester with the keymapping

Quit...

-quits ObjectView

Special:

Search...

-opens an input requester. Here you should enter a search pattern, after which the search function will be named.

Next

-searches the text for the next appearance of the search pattern.

Screen->Window

-closes its own screen or window on the prenamed pubscreen and opens a window on the current public screen.

Window->Screen

-closes the window and opens an own screen

OVPrefs

-not yet implemented

DTPrefs

-not yet implemented

Band:

Play

-plays an object (i.e samples or animations)

Stop

-stops a started object (is not yet supported by the datatypes)

Fast Forward

-"winds" an object fast forward (is not yet supported by the datatypes)

Pause

-pauses an object started with <Space> (is not yet supported by the datatypes)

Wind Backwards

-"winds" an object backwards (is not yet supported by the datatypes)

Navigation:

Jump to start

-jumps to first line of the object

Jump to end

-jumps to the last line of the object

Previous Page

-jumps to the previous page

Next Page

-jumps to the following page

Show Contents

-shows the contents of an object

Show keywords

-Shows the keyword index of the object

1.26 Author + Support

ObjectView-Author-Contact (EMail preferred):

EMail: M.Kraft@gandalf.berlinet.de

SMail: Matthias Kraft

Altenhofer Str. 29

13055 Berlin

Germany

Banking account:

Konto: 8287663000

BLZ : 100 200 00

Bank : Berliner Bank AG

I am always happy to receive comments or money ;-)...

As of August '96, ObjectView has it's own support box. The BBS Gandalf of course.

Port0: +49 (0)30 / 453 82 131 ZyXEL 19200

Port2: +49 (0)30 / 453 82 132 USR V. Every

Port3: +49 (0)30 / 453 82 133 ISDN X75

Port4: +49 (0)30 / 453 82 131 ISDN X75

Login: oview

no password

It always homes the newest version as an entire archive and as a patchfile to the last publicly released version. Besides that, the program 'spatch' and the archive 'tri14usr.lzx' (the triton.library) can be found there.

You can also e-mail me.

1.27 Thank yous to the specialists

I want to thank the people without whom it would never have come this far:

Michael van Elst for his support with the datatypes programming.

The programmers of the XPK-system (U.D.Mueller and others)

The Amiga-Special-Team, who have judged my little program so generously :-) (Amigaspecial 2/96)

Michael Bergmann for his bug reports and suggestions

The programmers of the requester-toolkit (Nico, Magnus)

Sven K. Stullich for his datatypes and his tips.

Stefan Ruppert for his tips and examples.

The Amiga Translators Organisation for all translations.

Stefan Zeiger for his triton.library, with which it is really easy to create a graphical user interface.

1.28 History

ObjectView-History:

(xx-xx-xx) -> not released to the public

last changed: 11-07-96

V2.10 20-09-96 -too lazy to really summarize the changes from Version

1.9. There are too many, beginning with completely rewritten routines, localization, newly added prefs program, new version number is absolutely justified ;-)...

V1.9 (01-07-96) -bug in overwriting removed

-requester will only be shown on an own screen if the window is minimum 320x200

-6 more arguments implemented

-ENV variable only holds the global prefs. These can be overwritten by tooltypes or cli arguments.

-cleaned up code and optimized it

-workaround search function

V1.8 (19-05-96) -memory check built in, so it does not guru when not enough chipmen is available

-error detection, error handling, error messages concerning objectloading redone(1) and built in (2+3)

-fileident built in on the CLI part: before an object is displayed, the name, datatype, with x height and the number of colours is displayed

-overwrite protection while saving implemented

-menus implemented (I ran out of keys)

-sound objects can't be shown on own screens due to some problems (it is weird that ADPCM objects consider themselves pictures which still leads to problems!?)

-current directory is always where the object was loaded from, so the amigaguide+ files (which load external objects) will now work.

V1.7 10-05-96 -XPK/PP20 handling changed --> read docs !!!

-only the window-default font is used. (Screen default font could be a proportional font, thus messing up tables etc.)

-path is displayed in the window title

-bug removed with the renaming of .xpk/.pp files to files

without the suffix

-IDCMP handling changed

-the way things were displayed was reworked, so you can display texts on an own screen or a prenamed public screen

-after displaying a requester on an own screen, the colours will be corrected

-.readme changed to .guide

V1.6 09-02-96 -code optimized

-bug removed. <shift>+<tab> now works.

-scrolling with the cursor keys in steps of 10 possible

-split up help requester

-if an object could not be loaded a file requester will now pop up instead of displaying an empty screen or reloading the previous object again

-if the file requester`s button cancel is selected, the program will terminate correctly (without a crash, and okok, I missed that one)

-reworked manual

V1.5 03-02-96 -removed arexx port (temporarily)

-implemented a print routine ('d','D') at last

-added the about-requester ('a','A')

-2 hits removed with the information requester (somehow I did not notice them)

-3 hits occurred, if an empty window was opened (also not seen by me)

-screen default font is used for text output (Michael Bergmann)

-little odds and ends

V1.4 27-01-96 -reworked code to be able to implement new features more easily

-added AREXX Port, but a little bit buggy (hits)

-for pictures that don`t give a DisplayModeID, BestModeID was implemented

-a few little bugs were removed and the code was optimized a bit

V1.3 18-11-95 -XPK support was added (please read the corresponding part in this manual)

-with 'n' or 'N' a picture can be reloaded and is then not displayed on an own screen

-with 'w' or 'W' you can save an object as IFF according

to its filetype (sound as 8SVX, pictures as ILBM, text as FTXT)

V1.2 12-11-95 -memory is allocated VMM-friendly

-changed information requester, now the

object type, which is easier to read, will be displayed

V1.1 21-09-95 -now the first argument is automatically loaded, or

if no argument is given a file requester will pop up

-removed a few little bugs (including the use of keys)

V1.0 15-09-95 -first public release
