

ObjectView

COLLABORATORS

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ObjectView	1
1.1	ObjectView V2.10	1
1.2	Einführung - Was ist ObjectView?	2
1.3	Bemerkungen - Disclaimer und Copyright	2
1.4	Vorraussetzungen - Was wird gebraucht?	3
1.5	XPk - Die Entpackoption	4
1.6	Suchen - Die Suchfunktion	4
1.7	OVPrefs - Der Konfigurationseditor	4
1.8	Die Argumente und ihre Bedeutung	5
1.9	Argumente: FILE	6
1.10	Argumente: NOXPk	6
1.11	Argumente: SCREEN	6
1.12	Argumente: WINDOW	6
1.13	Argumente: PUBSCREEN	7
1.14	Argumente: MAXWIDTH/MAXHEIGHT	7
1.15	Argumente: SCREENMODE/TEXTDEPTH	7
1.16	Argumente - LOOP/IMMEDIATE	8
1.17	Argumente: FONTNAME/FONTSIZE	8
1.18	Bedienung - Tasten, Menüs, Gadgets	9
1.19	Allgemeines zur Bedienung	9
1.20	Bedienung - Samples/Animationen	9
1.21	Bedienung - Dokumente	9
1.22	Bedienung - ObjectView Optionen	9
1.23	Bedienung - Tastenbelegung	10
1.24	Bedienung - Cursor-/Nummerntastatur	11
1.25	Bedienung - Menüs	12
1.26	Autor + Support	13
1.27	Dankeschön an die Spezialisten	14
1.28	Der Weg hierher	14

Chapter 1

ObjectView

1.1 ObjectView V2.10

ObjectView V2.10

(C)opyright 1995-96 by Matthias Kraft

Einführung Was ist ObjectView?

Bemerkungen Mehr oder weniger wichtig...

Voraussetzungen Was brauchst du?

XPk-Unterstützung Die Entpackoption...

Such-Funktion Nur für Texte...

Prefs-Editor Der Konfigurationseditor...

ToolTypes & CLI-Argumente Diverse Voreinstellungen

Tooltypes CLI-Argumente

>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=<Breite> MAXWIDTH/K/N

>X< MAXHEIGHT=<Höhe> MAXHEIGHT/K/N

>X< SCREENMODE=<ModeID> SCREENMODE/K

>X< TEXTDEPTH=<Tiefe> 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

Bedienung Tasten, Menüs, Gadgets...

>X< Allgemeine Bedienung

>X< Bedienung von Samples/Animationen
>X< Bedienung von Dokumenten
>X< Bedienung von ObjectView-Optionen
>X< Tastenbelegung
>X< Scrollen via Cursortasten/Nummerntastatur
>X< Menüs
Autorkontakt Meiner einer...
Dankeschön An die Spezialisten...
Geschichte Der Weg hierher...

1.2 Einführung - Was ist ObjectView?

ObjectView ist ein kleiner DataTypes-Anzeiger.

Die Highlights in Kürze:

- Unterstützt alle Datatypes
- Bedienung über Tastatur und Menüs
- Unterstützung von Workbench- bzw. CLI-Argumenten
- Speichern als IFF
- Drucken
- XPK- & Powerpacker-Unterstützung
- Suchfunktion für Texte
- Weit Konfigurierbar (jedes Objekt kann eine eigene Konfiguration enthalten)
- Prefs-Programm zum einfachen Erstellen dieser Konfigurationen
- Lokalisiert

Die Installation geschieht am Besten mit dem beiliegendem Installer-Script.

Für Leute die es unbedingt selbst machen wollen:

- ObjectView + Icon irgendwohin
- ObjectView.guide + ObjectView.prefs ebenfalls
- OVPrefs + Icon nach Sys:Prefs/
- ObjectView.catalog + OVPrefs.catalog nach Locale:Catalogs/[Sprache]/

1.3 Bemerkungen - Disclaimer und Copyright

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

ObjectView ist Freeware und darf frei verteilt werden, solange keine Vergütung, in welcher Form auch immer, dafür verlangt wird und solange das Paket vollständig weitergegeben wird.

ObjectView darf auf CDs gebrannt werden, wenn diese ausschließlich Public Domain Software enthalten und zu einem angemessenen niedrigem Preis vertrieben werden. Als Beispiel seien die Aminet-CDs genannt.

ObjectView darf in öffentlichen kostenlosen Softwarenetzen verteilt werden. Als Beispiel seien das FRAS-Netz und das Aminet genannt.

Soll ObjectView Teil eines kommerziellen Paketes werden oder möchte eine Zeitschrift ObjectView auf ihren Disketten/CDs verwenden/veröffentlichen ist zuerst meine schriftliche Erlaubnis einzuholen!

ObjectView wurde getestet und es sind keine Fehler bekannt. Trotzdem sind diese nicht auszuschliessen. Ich bin für KEINEN Schaden, der mittelbar oder unmittelbar mit dem Gebrauch von ObjectView im Zusammenhang steht, verantwortlich zu machen. Sie benutzen das Programm auf eigenes Risiko!

Sollten Sie einen Fehler entdecken, der direkt auf ObjectView zurückzuführen ist, wenden Sie sich bitte mit einer detaillierten Beschreibung an den Autor, damit dieser Fehler beseitigt werden kann.

Inhalt des Archives:

Alle .info-Dateien sind NewIcons.

Die Datei 'ObjectView.020' ist eine auf den 68020-Prozessor optimierte Version. Ist ein 68020 oder besserer Prozessor vorhanden, sollten Sie diese Version benutzen!

Zusätzliche (C)opyrights:

+++++

XPk is copyrighted by its authors. For further questions refer to

umueller@amiga.physik.unizh.ch (Urban Dominik Müller) [noch aktuell?]

+++++

ReqTools - reqtools.library

(c) 1991-1994 Nico François

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

updates: <http://www.rhein-main.de/people/szeiger/triton.htm>

+++++

1.4 Voraussetzungen - Was wird gebraucht?

Benötigt werden:

-ein Amiga mit OS 3.x und installierten Datatypes

-genügend freier Speicher (für Bilder, bzw. zum Entpacken)

-folgende Libraries müssen sich in LIBS: befinden:

asl.library V38+, datatypes.library V39+

-die icon.library V39+ wird zum Auslesen der Object-Icons benötigt (sollte eigentlich jeder installiert haben)

-die xpkmaster.library V2+ wird zum Entpacken benötigt, ist sie nicht vorhanden, kann man halt nicht entpacken ;)

-die reqtools.library V38+ wird z.Z. nur von der Suchfunktion benötigt, ist sie nicht vorhanden, kann man eben nicht suchen...

-die triton.library V5+ wird von OVPrefs benötigt!

-die locale.library V39+ wird für die Lokalisierung benötigt.

1.5 XPK - Die Entpackoption

Funktionsweise:

Wenn ObjectView ein Objekt als XPK- oder PP20(PowerPacker)-gepackt identifiziert, wird die xpkmaster.library mit dem Entpacken nach T:Objektname beauftragt. Diese entpackte Datei kann nach dem Anzeigen wieder gelöscht werden. Eine event. vorhandene Endung .xpk oder .pp wird gelöscht, damit Datatypes ggf. die Datei an der originalen Endung erkennen kann (z.B. der c.datatype benötigt die Endungen).

Weiterhin sollte man sicherstellen, daß entweder genügend RAM zur Verfügung steht (wenn T: z.B. nach RAM:T zeigt) oder genügend Plattenplatz vorhanden ist (z.B. bei T: nach DH0:T).

Es gibt inzwischen auch einen xpk.datatype, der normalerweise das Entpacken erledigen kann. AmigaGuide®-Dokumente haben jedoch Probleme damit, deswegen und weil die interne Funktion schneller ist, sollte man diese eingeschaltet lassen.

1.6 Suchen - Die Suchfunktion

Die Suchfunktion erreicht man über die Menüpunkte Spezial/Suchen... bzw. Spezial/Nächste.

Ein paar Einschränkungen vorweg:

Zur Zeit unterstützt die Funktion nur den c.datatype korrekt. Der c.datatype ist z.Z. der einzige Datatype, der die zum Suchen benötigte Line-Struktur vollständig ausfüllt.

In ASCII-Texten kann zwar auch gesucht werden (es wird auch gefunden :), da der ascii.datatype aber keine Positionsangaben in die Line-Struktur schreibt, wird versucht durch einen Workaround (den ich hier besser nicht näher beschreibe :) die Zeilenposition zu bestimmen. Es kann so gelegentlich vorkommen, daß eine falsche Position im Text angesprungen wird.

Der exe.datatype verhindert geradezu jegliches Bestimmen einer Zeilenposition. Dieser scheint die Position zwar auszufüllen, aber nicht korrekt, so daß hier auch ein Workaround notwendig wäre. Da der Autor des Datatypes aber zugesichert hat, dieses Manko zu beheben, sehe ich hierfür keine Notwendigkeit.

Zur Funktionsweise:

Ein Auswählen des Menüpunktes Spezial/Suchen... läßt einen EingabeRequester der reqtools.library erscheinen. Man wird aufgefordert ein Suchmuster einzugeben, d.h. das Wort, wobei GROSS/klein-Schreibung egal ist, optional mit Suchmustern (z.B. #?). Näheres zu den Mustern steht im AmigaDOS-Handbuch unter AmigaDOS-Sonderzeichen/Namensmuster.

Während des Suchens wird der Fenstertitel in Durchsuche... geändert und der Mauszeiger wird zum Wartezeiger. Wurde icon gefunden, wird die Zeile invertiert und versucht im sichtbaren Bereich darzustellen. Wurde nichts gefunden, wird wieder der normale Mauszeiger und Fenstertitel dargestellt und der Bildschirm blitzt einmal auf.

Ein Anwählen des Menüpunktes Spezial/Nächste sucht nach dem nächsten Auftreten des Suchbegriffs.

1.7 OVPrefs - Der Konfigurationseditor

OVPrefs V1.4

(C)opyright 1996 by Matthias Kraft

OVPrefs benötigt die triton.library V5+ im LIBS:-Verzeichnis!

OVPrefs dient zum Erstellen der Konfiguration von ObjectView. Er kann die globale Einstellung und die Einstellungen in den Icons verwalten.

Beim Start wird die globale Einstellung ausgelesen. Die Oberfläche reflektiert diese Einstellung.

Nach dem Ändern der Einstellung kann diese mit den Buttons 'Speichern' oder 'Benutzen' abgespeichert werden. Wobei die Einstellungen nach 'Benutzen' nur solange gültig sind, bis ein Neustart durchgeführt wird.

Durch Ablegen eines Icons in das Einstellungsfenster wird die Konfiguration aus dem Icon ausgelesen. Die Oberfläche wird entsprechend eingestellt und reflektiert danach die Einstellungen in diesem Icon.

Nach dem Ändern der Einstellungen kann man diese mit dem Menüpunkt 'Speichere Icon...' abspeichern. Es erscheint ein Dateirequester der den Pfad und den Dateinamen des Icons (ohne .info !!!) enthält. Nach 'OK' wird das Icon gespeichert.

Existiert noch kein Icon wird ein neues erstellt.

Die Möglichkeiten der Einstellung werden unter **Argumente** erläutert. Ich gehe hier nur noch kurz auf das Menü und die Einstellung 'Bildschirmtypen' ein.

Das Projekt-Menü:

->Öffne Global : Lädt die globalen Einstellungen erneut

->Speichere Global : Speichert die aktuellen Einstellungen als globale Einstellungen

->Öffne Icon... : Öffnet einen Dateirequester und erfragt einen Dateinamen, dessen Icon danach geladen wird. Achtung: Der Name darf nicht die Endung >.info< enthalten!!!

->Speichere Icon... : Öffnet einen Dateirequester um die Richtigkeit des Namens zu überprüfen und speichert nach dem 'OK' das Icon mit den aktuellen Einstellungen.

->Über... : Zeigt einen kleinen Inforequester.

->Beenden : Na was wohl?! Läßt OVPrefs die Blumen gießen und die Küche aufräumen, anschließend wird die Wäsche gewaschen und Staub gewischt. Wenn das alles erledigt ist, werden dem Autor 10.-DM aufs Konto überwiesen ;-)... schön wär's jedenfalls.

Die Bildschirmtypen-Einstellung:

Auf der linken Seite ist einzustellen, wo das Anzeigefenster geöffnet werden soll. Auf der rechten Seite werden dazu entsprechende Einstellungsmöglichkeiten freigegeben:

·eigenen Bildschirm nutzen

->Bildschirmtyp mittels ScreenModeRequester auswählen, und Tiefe festlegen.

·aktuellen PublicScreen nutzen

->keine zusätzlichen Einstellungen möglich.

·benannten PublicScreen nutzen

->dazu ist der Name (auf richtige Schreibweise achten!) des PublicScreens einzugeben. Ein AuswahlRequester gibt noch nicht dazu.

1.8 Die Argumente und ihre Bedeutung

Mit den Optionen kann man, in gewissen Grenzen, festlegen wie und wo ein Objekt dargestellt werden soll. Ich empfehle immer OVPrefs zu benutzen und die Einstellungen in einem Icon für das Objekt abzuspeichern.

Es gibt auch eine globale Einstellung in ENV:ObjectView.prefs, die ebenfalls mit OVPrefs erstellt werden kann. Die Schlüsselwörter in der globalen Einstellung entsprechen denen der Tooltypes.

Die Icons der Objekte werden nur ausgelesen, wenn ObjectView von der Workbench gestartet wurde. Wird ObjectView aus dem CLI (oder z.B. aus einem Script) gestartet, wird einmal die globale Einstellung ausgelesen und danach mit den eventl. angegebenen CLI-Argumenten überschrieben. Alle Objekte werden dann mit dieser Konfiguration angezeigt.

Erklärungen zu den einzelnen Argumenten:

>X< FILE

>X< DONTUSEXPk

>X< SCREEN

>X< WINDOW

>X< PUBSCREEN

>X< MAXWIDTH / MAXHEIGHT

>X< SCREENMODE / TEXTDEPTH

>X< LOOP / IMMEDIATE

>X< FONTNAME / FONTSIZE

1.9 Argumente: FILE

CLI:

FILE/A <name>

Wenn ObjectView aus dem CLI gestartet wird, muß ein vollständiger Dateiname (mit Pfad) übergeben werden. Wird kein Dateiname gefunden wird eine Fehlermeldung ausgegeben und ObjectView beendet.

1.10 Argumente: NOXPK

OVPrefs:

XPk/PP an/aus

Tooltype:

DONTUSEXPk=(TRUE|FALSE)

CLI:

NX=NOXPk/S

Wenn man sich den xpk.datatype besorgt und installiert hat, kann man die interne XPk-Unterstützung abschalten.

Voreinstellungsmäßig und natürlich nur wenn die xpk.library vorhanden ist, wird die interne XPk-Unterstützung benutzt.

1.11 Argumente: SCREEN

OVPrefs:

eigenen Bildschirm nutzen (Typ und Tiefe können angegeben werden)

Tooltype:

SCREEN=(TRUE|FALSE)

CLI:

S=SCREEN/S

Muß angegeben werden, möchte man auch Texte auf einem extra Screen betrachten.

1.12 Argumente: WINDOW

OVPrefs:

aktuellen PublicScreen nutzen

Tooltype:

WINDOW=(TRUE|FALSE)

CLI:

W=WINDOW/S

Wenn hingegen auch Bilder auf dem aktuellen PublicScreen (meist die Workbench) gezeigt werden sollen.

Voreinstellungsmäßig werden Bilder und Animationen (sofern genügend Speicher vorhanden) auf einem eigenen Screen und Texte, AmigaGuides, Sounds, etc. in einem Fenster auf dem aktuellen PublicScreen dargestellt.

1.13 Argumente: PUBSCREEN

OVPrefs:

benannten PublicScreen nutzen (Namen eingeben)

Tooltpe:

PUBSCREEN=<name>

CLI:

PS=PUBSCREEN/K <name>

Wenn alles auf einem bestimmten PublicScreen (z.B. DOPUS.1) angezeigt werden soll. <name> wird durch den PublicScreen-Namen ersetzt. Es muß auf GROSS/klein-Schreibung geachtet werden!

1.14 Argumente: MAXWIDTH/MAXHEIGHT

OVPrefs:

Breite

Höhe

Tooltpe:

MAXWIDTH=<breite>

MAXHEIGHT=<höhe>

CLI:

X=MAXWIDTH/K/N <breite>

Y=MAXHEIGHT/K/N <höhe>

Man kann für Objekte, die dimensionslos sind, die Fensterdimensionen festlegen, d.h. Breite und/oder Höhe. Dimensionslose Objekte sind z.B. ASCII-Texte.

Sollten Bilder zu groß sein und es ist nicht möglich ein Bildschirm dieser Ausmaße zu öffnen, wird versucht einen Bildschirm mit den hier angegebenen Ausmaßen zu öffnen. Man kann dann mit den Cursortasten das Bild bewegen.

Voreingestellt ist MAXWIDTH=640 und MAXHEIGHT=400. Das sind die Dimensionen für Standart NTSC Hires Lace.

1.15 Argumente: SCREENMODE/TEXTDEPTH

OVPrefs:

Voraussetzung: 'eigenen Bildschirm nutzen' ist angewählt

Bildschirmmodus nur mit Requester auswählbar

Tiefe

Tooltypes:

SCREENMODE=<modeid>

TEXTDEPTH=<tiefe>

CLI:

SM=SCREENMODE/K <modeid>

TD=TEXTDEPTH/K/N <tiefe>

Man kann für Texte und Dokumente den Bildschirmmodus und die Bildschirmtiefe (Anzahl der Farben) einstellen. Es reicht z.B. einen reinen ASCII-Text mit einem 2-Farben-Bildschirm (Tiefe=1) zu lesen, ein AmigaGuide®-Dokument braucht jedoch min. 4 Farben (Tiefe=2) und beim C-Source fängt der Spaß gerade erst bei 8 Farben an (Tiefe=3). Es ist eine maximale Tiefe von 8 möglich (256 Farben)!

Der Bildschirmmodus muß als Hexadezimalzahl angegeben werden (z.B. NTSC Hires Lace => 19004). Die Ausdehnungen des Bildschirms werden aus MAXWIDTH bzw. MAXHEIGHT genommen.

Werden die Argumente nicht angegeben, wird BestModeID() benutzt, um den besten Bildschirmmodus zu finden und es ist eine Tiefe von 3 (8 Farben) voreingestellt.

1.16 Argumente - LOOP/IMMEDIATE

OVPrefs:

Wiederholen an/aus

Sofort an/aus

Tooltypes:

LOOP=(TRUE|FALSE)

IMMEDIATE=(TRUE|FALSE)

CLI:

LOOP/S

IM=IMMEDIATE/S

Die Optionen sind für Sounds und Animationen gedacht. Mit IMMEDIATE wird der Sound oder die Animation sofort nach dem Laden abgespielt. Mit LOOP wird festgelegt, daß Sound/Animation ständig wiederholt werden sollen. Leider unterstützt dies bis jetzt weder der Sound- noch der Animations.datatype.

Voreinstellungsmäßig sind beide Optionen ausgeschaltet.

1.17 Argumente: FONTNAME/FONTSIZE

OVPrefs:

Wähle Schrift für Texte (empfehle über Requester)

Tooltypes:

FONTNAME=<name>

FONTSIZE=<ysize>

CLI:

FN=FONTNAME <name>

FS=FONTSIZE <ysize>

Mit diesen beiden Optionen, kann man die Schriftart festlegen in der Texte dargestellt werden sollen. Es müssen immer beide zusammen angegeben werden! <name> muß immer die Endung <.font> enthalten.

Voreinstellungsmäßig wird der Systemfont benutzt.

1.18 Bedienung - Tasten, Menüs, Gadgets

Die Bedienung ist über Tastensteuerung bzw. Menüs realisiert. Die zur Zeit einzigen Gadgets sind die Systemgadgets.

>X< Allgemeine Bedienung

>X< Bedienung von Samples/Animationen

>X< Bedienung von Dokumenten

>X< Bedienung von ObjectView-Optionen

>X< Tastenbelegung

>X< Scrollen via Cursortasten/Nummerntastatur

>X< Menüs

1.19 Allgemeines zur Bedienung

Das wichtigste zuerst: Wie werde ich dieses Programm wieder los? ;-)... Das erledigen die Tasten 'q','Q' oder <ESC>, wenn dargestellt das Closegadget links oben oder, wenn aufrufbar, der Menüpunkt Projekt/Beenden.

Die einzelnen Datatypes unterstützen immer nur Teile der Möglichkeiten, logisch, denn warum sollte man ein Bild abspielen können oder bei einem Sample die nächste Seite aufschlagen müssen ;-)...

1.20 Bedienung - Samples/Animationen

Das 'Band'-Menü bietet ein paar Optionen zur Bedienung von Samples und Animationen. Allerdings ist die einzige Option, die zur Zeit von den Datatypes unterstützt wird: 'Abspielen'.

Es ist ebenfalls möglich Samples und Animationen so zu laden, daß sie sofort abgespielt werden. Siehe [Argumente/IMMEDIATE](#).

1.21 Bedienung - Dokumente

Das Suchen in Dokumenten ist bis jetzt nicht möglich.

Die Menüpunkte Navigation/Zeige Inhalt und Navigation/Zeige Stichwörter sind speziell für Dokumente gedacht, da diese fast immer ein Inhaltsverzeichnis und oftmals ein Stichwortverzeichnis (Index) haben.

Beim Wechsel via Bildschirm->Fenster, oder umgekehrt, wird wieder die erste Seite des Dokuments aufgerufen.

1.22 Bedienung - ObjectView Optionen

Siehe auch Erklärungen zu den [Menüs](#).

ObjectView stellt einen InfoRequester zur Verfügung, der detaillierte Angaben zu dem geladenem Objekt enthält.

ObjectView unterstützt das Ausdrucken und Abspeichern. Nicht jeder Datatype stellt allerdings die dazu benötigten Funktionen zur Verfügung. D.h. Abspeichern z.B. sollte immer eine IFF-Datei zur Folge haben, es gibt aber Datatypes die entweder gar nicht speichern oder ein eigenes Format verwenden.

ObjectView stellt für Texte eine Suchfunktion zur Verfügung.

Große Objekte können mit den Cursortasten oder der Nummerntastatur durch das Fenster bewegt werden.

1.23 Bedienung - Tastenbelegung

Allgemein:

Tasten: l, L, 0

-laden eines neuen Objekts

Tasten: d, D

-druckt angezeigtes Objekt aus (z.B. Bild oder Text)

Tasten: w, W

-speichert Objekt als IFF (also Samples als 8SVX, Bilder als ILBM, Texte als FTXT)

Tasten: a, A

-anzeigen eines Requesters über das Programm

Tasten: i, I

-zeigt Requester mit Informationen über das geladene Objekt

Tasten: h, H

-zeigt zweiteiligen Hilfe-Requester mit Tastaturbelegung

Tasten: q, Q, <ESC>

-beenden von ObjectView

Samples/Animationen:

<Leertaste>

-spielt ein Objekt ab (z.B. Samples oder Animationen)

Tasten: f, F

-"spult" Objekt vor (wird von den Datatypes noch nicht unterstützt)

Tasten: p, P

-hält ein mit <Leertaste> gestartetes Objekt an (wird von den Datatypes noch nicht unterstützt)

Tasten: r, R

-"spult" Objekt zurück (wird von den Datatypes noch nicht unterstützt)

Tasten: s, S

-stoppt ein mit <Leertaste> gestartetes Objekt (wird von den Datatypes noch nicht unterstützt)

Dokumente:

<Tab>

-wählt das nächste Feld an (z.B. bei AmigaGuide®)

<Shift>+<Tab>

-wählt das vorhergehende Feld an

<Enter>

-aktiviert angewähltes Feld

Taste: \

-geht ein Schritt zurück

Taste: <

-blättert zur vorhergehenden Seite

Taste: >

-blättert zur nächsten Seite

Tasten: c, C

-zeigt Inhalt eines Objekts

Tasten: x, X

-zeigt Stichwortverzeichnis des Objekts

1.24 Bedienung - Cursor-/Nummerntastatur

Wenn Objekte zu groß sind, kann man sie durch das Fenster bewegen:

Auf der Nummerntastatur:

1 (End)

-springt zur letzten Zeile des Objekts

2

-springt um eine Zeile nach unten

3 (Pg Dn)

-springt um eine Seite nach unten

4

-springt um eine Spalte nach links

6

-springt um eine Spalte nach rechts

7 (Home)

-springt zur ersten Zeile des Objekts

8

-springt um eine Zeile nach oben

9 (Pg Up)

-springt um eine Seite nach oben

mit den Cursortasten:

Cursor Up

-springt um 10 Zeilen nach oben

Cursor Down

-springt um 10 Zeilen nach unten

Cursor Left

-springt um 10 Spalten nach links

Cursor Right

-springt um 10 Spalten nach rechts

(können keine 10 Zeilen/Spalten mehr gesprungen werden, wird das Objekt nur noch Zeilen-/Spaltenweise bewegt)

1.25 Bedienung - Menüs

Projekt:

Öffnen...

-laden eines neuen Objekts

Drucken

-druckt angezeigtes Objekt aus (z.B. Bild oder Text)

Speichern als...

-speichert Objekt als IFF (also Samples als 8SVX, Bilder als ILBM, Texte als FTXT)

Über...

-anzeigen eines Requesters über das Programm

Info...

-zeigt Requester mit Informationen über das geladene Objekt

Hilfe...

-zeigt zweiteiligen Hilfe-Requester mit Tastaturbelegung

Beenden...

-beenden von ObjectView

Spezial:

Suchen...

-öffnet einen EingabeRequester. Hier ist ein Suchmuster einzugeben. Danach wird die **Suchfunktion** aufgerufen.

Nächste

-durchsucht Text nach dem nächsten Auftreten des Suchmusters.

Bildschirm->Fenster

-schließt eigenen Bildschirm oder Fenster auf benannten PublicScreen und öffnet Anzeigefenster auf dem aktuellen PublicScreen.

Fenster->Bildschirm

-schließt Anzeigefenster und öffnet es auf einem eigenem Bildschirm

OVPrefs

-noch nicht implementiert

DTPrefs

-noch nicht implementiert

Band:

Abspielen

-spielt ein Objekt ab (z.B. Samples oder Animationen)

Stoppen

-stoppt ein gestartetes Objekt (wird von den Datatypes noch nicht unterstützt)

Schnell vor

-"spult" Objekt vor (wird von den Datatypes noch nicht unterstützt)

Pause

-hält ein gestartetes Objekt an (wird von den Datatypes noch nicht unterstützt)

Schnell rück

- "spult" Objekt zurück (wird von den Datatypes noch nicht unterstützt)

Navigation:

Springe an Anfang

- springt zur ersten Zeile des Objekts

Springe ans Ende

- springt zur letzten Zeile des Objekts

Seite hochspringen

- springt um eine Seite nach oben

Seite runterspringen

- springt um eine Seite nach unten

Zeige Inhalt

- zeigt Inhalt eines Objekts

Zeige Stichwörter

- zeigt Stichwortverzeichnis des Objekts (Index)

1.26 Autor + Support

ObjectView-Autor-Kontakt (EMail bevorzugt):

EMail: M.Kraft@gandalf.berlinet.de

SMail: Matthias Kraft

Altenhofer Str. 29

13055 Berlin

Germany

Bankverbindung:

Konto: 8287663000

BLZ : 100 200 00

Bank : Berliner Bank AG

Ich freue mich immer über Kommentare und monetäre Zuwendungen ;-)...

ObjectView hat ab August '96 auch eine Supportbox. Die Gandalf:

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

kein Passwort

Dort liegt immer die neueste Version als vollständiges Archiv und als Patchdatei auf die letzte veröffentlichte Version. Ausserdem ist das Patchprogramm 'spatch' und das Archiv 'tri14usr.lzx' (enthält die triton.library) dort zu finden.

Es besteht auch die Möglichkeit online eine Nachricht an mich zu senden.

Die jeweils neueste Version der triton.library gibt es auch direkt beim Autor: <http://www.rhein-main.de/people/szeiger/triton.htm>

1.27 Dankeschön an die Spezialisten

Mein Dank geht an die Leute ohne die es nie dazu gekommen wäre:

Michael van Elst für seine Unterstützung beim DataTypes-programmieren.

die Programmierer des XPK-Systems (U.D.Müller u.a.)

das Amiga-Special-Team, das mein kleines Proggy so überaus großzügig bewertet hat ;-) (Amiga special 2/96)

Michael Bergmann für Bugreport und Vorschläge

die Programmierer des Requester-Toolkits (Nico, Magnus)

Swen K. Stullich für seine Datatypes und seine Tips.

Stefan Ruppert für seine Tips und Beispiele.

Die Amiga Translators Organization, für sämtliche Übersetzungen.

Stefan Zeiger für seine triton.library, mit der es wirklich einfach ist, eine GUI zu erstellen.

1.28 Der Weg hierher

ObjectView-Historie:

(xx-xx-xx) -> nicht veröffentlicht

letzte Änderung: 27-07-96

V2.10 20-09-96 -bin zu faul alle Veränderungen seit V1.9 aufzuzählen

es sind zuviele, angefangen bei komplett neu

geschriebenen Routinen über Lokalisierung bis zum neu

dazugekommenen Prefs-Programm, die neue Versionsnummer

ist jedenfalls absolut gerechtfertigt ;-)...

V1.9 (01-07-96) -Bug im Überschreibschutz beseitigt

-Requester (About, Help, Info) werden nur noch

auf dem eigenen Schirm dargestellt, wenn das Fenster

min. 320x200 groß ist

-6 weitere Argumente implementiert

-ENV-Variable enthält nun globale Prefs, diese werden

durch Tooltypes bzw. CLI-Args überschrieben

-Code aufgeräumt und optimiert

-Workaround Suchfunktion

V1.8 (19-05-96) -Speichercheck eingebaut, um Absturz wegen Chipmem-

mangels zu umgehen

-Fehlererkennung, -behandlung, -meldungen beim Objekt-

laden überarbeitet(1) bzw. eingebaut(2+3)

-Fehlermeldungen eingebaut

-im CLI-Part Fileident eingebaut, bevor ein Objekt

angezeigt wird, wird der Name, DTyp, BxH und Anzahl

der Farben ausgegeben

- Suchfunktion für Text-Objekte eingebaut
- Überschreibschutz beim Speichern eingebaut
- Menüs eingebaut (mir gingen die Tasten aus ;-)
- Sound-Objekte können nicht mehr auf eigenem Screen dargestellt werden, führte zu Problemen (seltsamerweise halten sich ADPCM-Objekte für Bilder?!, was immernoch zu Problemen führt)
- aktuelles Verzeichnis ist immer das, des geladenen Objekts, damit sollten nun auch die AmigaGuide®, die externe Objekte laden, funktionieren

V1.7 10-05-96 -XPK/PP20-Handling verändert -> Anleitung lesen !!!

- es wird nun der Window-Default-Font verwendet (Screen-Default-Font kann auch ein Proportionalfont sein -> ASCII-Tabellen etc. stimmen nicht mehr)
- es wird nun immer der Pfad im Windowtitel angezeigt
- Bug beim Beseitigen der .xpk/.pp-Endungen behoben
- ToolTypes/CLI-Args hinzugefügt
- IDCMP-Handling verändert
- Darstellungsarten überarbeitet, so können nun z.B. Texte auf einem eigenem Screen oder einem angebbaren Publicscreen angezeigt werden
- nach dem Zeigen eines Requesters auf dem eigenen Screen, werden nun die Farben korrigiert
- .readme zu einem .guide gewandelt

V1.6 09-02-96 -Codeoptimierungen

- Bug beseitigt, <Shift>+<Tab> funktioniert nun
 - Scrollen mit Cursortasten in 10er Schritten möglich
 - Help-Requester gesplittet
 - kann ein Objekt nicht geladen werden, erscheint nun kein leeres Fenster mehr bzw. wird nicht das alte Objekt nochmal geladen, sondern der File-Requester erscheint erneut
 - wird der File-Requester mit 'Abbruch' quittiert, wird nun das Programm korrekt beendet (ohne abzustürzen, ja ja auch das hatte ich übersehen)
 - Anleitung überarbeitet
- V1.5 03-02-96 -AREXX-Port wieder entfernt (vorläufig)
- endlich Druckroutine implementiert ('d','D')
-

- einen About-Requester dazuprogrammiert ('a','A')
- 2 Hits im Info-Requester entfernt (waren mir irgendwie nicht aufgefallen)
- 3 Hits traten auf, wenn ein leeres Fenster geöffnet wurde (hatte ich auch übersehen)
- es wird nun der Screen-Default-Font für Textausgaben verwendet (Michael Bergmann)
- diverse Kleinigkeiten

V1.4 27-01-96 -Code überarbeitet, um Ergänzungen besser integrieren zu können

- AREXX-Port eingebaut, aber noch leicht buggy (Hits)
- für Bilder die keine DisplayModeID liefern, BestModeID()-Support eingebaut
- div. kleine Bugs beseitigt, bzw. Code optimiert

V1.3 18-11-95 -XPK-Unterstützung eingebaut (Bitte den entsprechenden Abschnitt dieser Anleitung dazu lesen!)

- mit 'n' bzw. 'N' kann ein Bild nochmal geladen werden und wird dann ohne eigenen Screen dargestellt
- mit 'w' bzw. 'W' kann ein Objekt in das entsprechende IFF-Format abgespeichert werden (Sound als 8SVX, Bild als ILBM, Text als FTXT)

V1.2 12-11-95 -Speicher wird nun VMM-freundlich beschafft

- Inforequester verändert, es wird nun der besser lesbarere Objekttyp angezeigt

V1.1 21-09-95 -jetzt wird jeweils das erste Argument automatisch geladen oder bei keinem Argument der Dateirequester dargestellt

- ein paar kleine Bugs beseitigt(u.a. in Tastaturbedg.)

V1.0 15-09-95 -erste Veröffentlichung
