

AView

Guido Mersmann

COLLABORATORS

	<i>TITLE :</i> AView		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Guido Mersmann	August 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AView	1
1.1	AView	1
1.2	Einführung	1
1.3	Systemvoraussetzung	2
1.4	Featureliste	2
1.5	Distribution	2
1.6	Wie kann man sich registrieren lassen ?	3
1.7	Einschränkungen dieser Version	3
1.8	Der Schnellstart	4
1.9	Installieren von AView	4
1.10	AView	5
1.11	Geschichtliches	5
1.12	Zukünftiges	7
1.13	Bugliste	7
1.14	Danksagungen	8
1.15	Der Autor	8
1.16	Andere Programme von mir !	8
1.17	Technische Informationen über AView	10
1.18	Die Befehle von AView	10
1.19	Der Befehl: MATCH	11
1.20	Der Befehl: FIND	12
1.21	Der Befehl: NAME	13
1.22	Der Befehl: FILESIZE	13
1.23	Der Befehl: ASCII	14
1.24	Der Befehl: BINARY	15
1.25	Der Befehl: AND	15
1.26	Der Befehl: OR	16
1.27	Der Befehl: NOT	17
1.28	Der Befehl: ()	18
1.29	Das Pattern: Name/M	18

1.30	Das Pattern: All/S	19
1.31	Das Pattern: ARG=To=From/K	19
1.32	Das Pattern: SetComment=SC/S	20
1.33	Das Pattern: ShowDetect=SD/S	20
1.34	Das Pattern: Debug/S	20
1.35	Die Voreinstellungen von AView	21
1.36	Gadget: Sortieren nach	21
1.37	Gadget: Name	21
1.38	Gadget: Auswahl	22
1.39	Gadget: Löschen	22
1.40	Gadget: Alle	22
1.41	Gadget: Neu	22
1.42	Gadget: Speichern	22
1.43	Gadget: Benutzen	23
1.44	Gadget: Abbrechen	23
1.45	Das Editierfenster	23
1.46	Gadget: Name	23
1.47	Gadget: Klasse	24
1.48	Gadget: Unter Klasse	24
1.49	link	25
1.50	Gadget: Stack	25
1.51	Gadget: Kommando	25
1.52	Gadget: Aktueller Typ	26
1.53	Gadget: Aktuelles Verzeichnis	26
1.54	Gadget: WB Ausgabedatei	27
1.55	Gadget: CLI Ausgabedatei	27
1.56	Gadget: Identifizierer	27
1.57	Gadget: Vordefiniertes %a	27
1.58	Das Auswahlfenster	27
1.59	Die Voreinstellungen der Voreinstellungen	28
1.60	Gadget: Expertenmodus	28
1.61	Gadget: Speichern	28
1.62	Gadget: Benutzen	29
1.63	Gadget: Abbrechen	29
1.64	Die Menüs von AViewPrefs	29
1.65	Menu: Öffnen	29
1.66	Menu: Speichern Als	30
1.67	Menu: Beenden	30
1.68	Menu: Reset to Default	30

1.69	Menu: Last Saved	30
1.70	Menu: Restore	30
1.71	Menu: AView Prefs	30
1.72	AVDeficons	31
1.73	Was macht AVDeficons?	31
1.74	Was kann AVDeficons?	31
1.75	AVDeficons Installation	31
1.76	Wie definiere ich die neuen Piktogramme?	32
1.77	Verzeichnis Icons	32
1.78	Disketten Piktogramme	33
1.79	Datei Piktogramme	33
1.80	AVDeficons und die Probleme, die dahinter stehen!	34
1.81	Probleme durch AVDeficons	34
1.82	AVDeficons Template	35
1.83	Tips,Tricks,Fragen und Antworten	36
1.84	Warum startet AView nicht?	36
1.85	Die Onlinehilfe funktioniert nicht!	37
1.86	Die Onlinehilfe ist nur in englisch	37
1.87	AVViewPrefs ist manchmal sehr langsam!	37

Chapter 1

AView

1.1 AView

AView V1.46

(C) Guido Mersmann in 1996-1998

AVIEW IS A SHAREWARE PRODUCT

READ THE DISTRIBUTION SECTION FOR INFORMATION ON DISTRIBUTION

Contents

Einführung	
Systemvoraussetzungen	
Featureliste	
Distribution	
Die Registrierung	
Installieren	von AView
Der Schnellstart	
AView	
AView_Prefs	
Die Kommandos von AView	
AVDeficons	Der Piktogramm-Manager
Tips, Tricks, Fragen und Antworten	
Geschichtliches	
Zukünftiges	
Bekannte Fehler	
Danksagungen	
Der Autor	Updates, Bugreports, Anregungen...
Andere Programme von mir !	
Technische Informationen	über AView

1.2 Einführung

Einführung

AView ist eine Ergänzung zu den OS-Datentypen und vor allen Dingen zu Multiview. Sie werden jetzt fragen, wozu ? Nun, jede Datei braucht ihren Datentyp. Für viele Dateiformate gibt es keine, schlechte oder langsame Datentypen. Oder man möchte einfach nur eine bestimmte Aktion ausführen oder einen bestimmten Viewer benutzen. Genau hier setzt AView ein.

AView erlaubt das automatische Starten von Programmen und Tools ausgehend von der oder den angegebenen Dateien.

1.3 Systemvoraussetzung

Systemvoraussetzungen

Sie benötigen mindestens OS2. Die Oberfläche ist unter OS3.x besser zu bedienen, was besonders bei Listviews hervorsteicht.

Sollen Programme im Workbench-Modus gestartet werden, ist die WBStart.library V2.x von Stefan Becker nötig. Diese Library kann über das Aminet bezogen werden.

Mehr ist eigentlich nicht nötig.

1.4 Featureliste

Features

\textdegree{} Locale Unterstützung

\textdegree{} Benötigt Kickstart 2.04 oder höher

\textdegree{} Verschiedene Version für die verschiedenen Betriebssysteme.

\textdegree{} Nach Commodore Styleguide programmiert.

\textdegree{} Sehr kurz, da in Assembler geschrieben.

\textdegree{} Online-Hilfe

\textdegree{} vorkonfigurierte Dateitypen

\textdegree{} Flexibel durch hinzufügen eigener Dateitypen.

1.5 Distribution

Distribution

Dieses Programm darf auf allen jederman zugänglichen Medien veröffentlicht werden, solange im Archiv alle Files unverändert beiliegen. Bei CD Veröffentlichungen möchte ich gerne ein kostenloses Exemplar.

AView ist Shareware. AView enthält ein paar kleine Einschränkungen, die an mich erinnern sollen.

Ich erwarte von jedem, der AView benutzt, daß er sich registrieren läßt. Ich denke das tut keinem Weh und mich spornt es an neue Features zu integrieren und schnell neue Anpassungen vorzunehmen.

Updates sind über das Aminet und über die AD SG Homepage <http://www.amigaworld.com/adsg> zu bekommen.

Für Schäden oder Datenverluste, die durch AView auftreten, hafte ich nicht. Benutzung auf eigene Gefahr !!!

1.6 Wie kann man sich registrieren lassen ?

Wie kann man sich registrieren lassen ?

Wer AView komplett benutzen will kann sich über AD SG registrieren lassen. Der Preis dafür beträgt 15DM oder 10\$. Ich denke der geforderte Betrag ist für dieses Programm nicht zu hoch und ist auch für die kleine Briefftasche erschwinglich.

Sie können sich jetzt sofort hier registrieren lassen.

Die Registrierung von AView geschieht über die Amiga Developer Shareware Group (AD SG). Genaueres über die Zahlungs- und Versandmöglichkeiten finden Sie im AD SG-RegTool. Das Keyfile wird Ihnen innerhalb von 1 bis 2 Wochen nach Eingang des Geldes zugestellt.

Bitte benutzen Sie nur das mitgelieferte Programm zur Registrierung, da Sie uns sonst die Arbeit unnötig erschweren und den Versand Ihres Keyfiles verzögern.

1.7 Einschränkungen dieser Version

Einschränkungen dieser Version

AView ist in der unregistrierten Version auf maximal 10 Identifizierer beschränkt.

Einfach gesagt es ist genauso, als wenn der original Multiview nicht

mehr als 10 DataTypes könnte.

Das ist zum Test wohl ausreichend.

Nach der Registrierung erlaubt AView das Einbinden beliebig vieler Dateitypen.

1.8 Der Schnellstart

Schnellstart

Installieren Sie das Archive und starten Sie AViewPrefs. Einige Dateitypen sind bereits defaultmäßig eingebunden und können verändert werden.

Wenn Sie weitere hinzufügen wollen, so betätigen Sie einfach das Auswählen Gadget. Es erscheint ein Requester, der die Auswahl der voreingestellten Dateitypen erlaubt. Mit Ok, Return oder Doppelklick befördern Sie diese Typen in Ihre Voreinstellungsliste.

Nach dem Anwählen durch die Return-Taste oder einen Doppelklick auf dem entsprechenden Dateityp, erscheint das Editier-Fenster. In diesem werden alle Angaben eingestellt, die bei Erkennen dieses Dateitypes erforderlich sind.

Im Schnellstart brauchen wir nur Kommando, in dem Sie das auszuführende Programm einstellen. Ein Prozent %f hinter dem Namen entspricht dem Dateinamen der Datei, die Sie später an AView übergeben.

Diese Aktionen wiederholen Sie, bis Sie alle gewünschten Dateitypen konfiguriert haben. Dann speichern Sie die Liste und fertig.

Beachten Sie, daß die unregistrierte Version nur 10 Dateitypen zuläßt.

AView erkennt jetzt alle Dateien, die sie eingestellt haben und benutzt die unter Kommando eingestellten Programme.

1.9 Installieren von AView

Installieren von AView

Für die Installation benutzen Sie bitte den original ESCOM Installer. Er legt ein Verzeichnis an, in dem AView installiert wird.

Versuchen Sie nicht AView von Hand zu installieren!

Sollten Sie diese neue Version des Installers nicht besitzen, so können Sie sie über meine SupportBox unter dem Namen "Installe.lha" beziehen.

Sollte ihnen AView nicht zusagen, können Sie die ganze Schublade

löschen und damit AView komplett vernichten. Vergessen Sie nicht den Path-Befehl aus der User-Startup zu löschen.

1.10 AView

AView

Ist das Hauptprogramm, daß für das tägliche Arbeiten nötig ist. Der Aufruf erfolgt mittels CLI, Workbench oder als AppIcon durch Toolmanager oder einem vergleichbaren Programm.

Start vom CLI

Das Template für AView lautet:

```
"AView Name/M,ARG=To=From/K,All/S,SC=SetComment/S,SD=ShowDetect/S,
  Debug/S"
```

Start von der Workbench

Dieser Start ist eigentlich nicht sinnvoll, da AView hier keine Dateinamen übernehmen kann und immer einen ASL-Requester öffnet. Es sei den sie haben eine Datei, die im Icon-Tool "AView" stehen hat. Ein doppelklicken dieses Icons startet AView und zeigt die Dateien an.

Toolmanager und Co

Mit Toolmanager oder vergleichbaren Programmen können Sie AView auf Docks legen, ihnen Icons zuweisen und vieles mehr. AView wird ohne zu murren alle Dateien anzeigen, die Sie ihm übergeben.

Natürlich sollten Sie vorher die Dateitypen einstellen, die Sie zu nutzen gedenken.

1.11 Geschichtliches

Geschichtliches

Dieser Text erzählt etwas tagebuchmäßig, was sich in den verschiedenen Versionen getan hat.

0.95B Die erste lauffähige beta Version.

0.96B Kleine Änderung an AView vorgenommen.

- Die Anleitung ein wenig überarbeitet und an die aktuellen Prefs angepaßt.
- 0.97B Kleines CurrentDir-Problem behoben, daß immer auf mfs-handlern und beim AmiCDFS zu einem Absturzführte. Anscheinend macht die DOS-Funktion SystemTagList hier beim Tag "NP_Current" Dinge, die diesen Geräten nicht gefallen und zum Absturz führen. "df0:" ohne mfs machte keine Probleme.
AViewPrefs sortiert jetzt die auswählbaren Einträge automatisch.
 - 0.98B Kleinen Fehler in der Match-Routine behoben.
Debugmodus eingebaut.
Kleinere Änderungen an den Prefs.
 - 0.99B Kleine interne Verbesserungen.
 - 1.00 Kleine Fehler behoben
ShowDetect feature eingebaut. (gewünscht von Andreas Boerner)
Empty-Dateifehler behoben. (gemeldet von Andreas Boerner)
Fanzösische Katalogdateien eingebaut. (Thanks to Jerome Chesnot)
Die Stringgadgets verbreitert. (gewünscht von Andreas Boerner)
AViewPrefs läßt sich jetzt nicht mehrmals starten.
 - 1.01 NOT Kommando eingebaut. (gewünscht von Andreas Boerner)
 - 1.02 Kleine Optimierungen am Source.
 - 1.03 "1" und "0" können jetzt als Variablen benutzt werden. (z.B 1 & 0)
Damit kann man jetzt AView auch dazu benutzen beliebige Gleichungen zu lösen. Ich hab die Funktion allerdings nur eingebaut, damit ich schnell testen kann, ob die berechnung funktioniert, ohne mir Test-Dateien schreiben zu müssen.
 - 1.04 Kleine Aktualisierungen an der Dokumentation vorgenommen.
 - 1.05 Lha kann jetzt auch definiert werden, wenn der DMS schon drin ist.
(gemeldet von Andreas Boerner)
 - 1.06 Neues Kommando FILESIZE. Dadurch wird eine sichere Identifikation von Files möglich, die immer die gleiche länge haben. (z.B ADF)
 - 1.07 BUGFIX: Die ab und an auftretende Endlosschleife ist jetzt behoben. Ganz sicher! Hat auch nur 5 Stunden gedauert, da der Debugger ja nicht mitlaufen konnte. Der Fehler war ja nicht reproduzierbar.
(gemeldet von Andreas Boerner, Silke Bormann und mir)
 - 1.08 Kickstart-Typen eingebaut und einen Fehler in der ShowDetect Routine entfernt.
 - 1.09 Keyfile-Unterstützung.
 - 1.10 Viele neue Dateitypen.
 - 1.11 AView benutzt bei ShowDetect und SetComment jetzt auch die nicht in AView eingestellten Dateitypen.
 - 1.12 AView erlaubt jetzt auch das Abschalten von "" in der Kommandozeile. (gewünscht von Steffen Koch)
 - 1.13 Neue %a Option, erlaubt das gezielte Übergeben von Argumenten an das Zielkommando. Das erlaubt z.b eine Datei in ein bestimmtes Verzeichnis entpacken zu lassen. Durch die Default Option bleibt bei fehlen dieser Angabe alles beim Alten.
 - 1.14 Anleitung überarbeitet.
 - 1.15 Fehler in der WB Ausführung entfernt.
 - 1.16 ASCII erkennt jetzt auch ESC-Sequenzen.
 - 1.17 Das komplette Konzept der Erkennung wurde überarbeitet.
 - 1.17 Neuer Prefs Requester. Die Up/Down Gadgets sind verschwunden und die Einstellungen der Dateitypen erfolgt über ein zweites Fenster.
 - 1.29 Expertenmodus eingebaut.
 - 1.30 Kleinere Fehler behoben und weitere Punkte in der Anleitung behoben.
 - 1.31 Menüs eingebaut
AViewPrefs-Voreinsteller-Fenster eingebaut. Es gibt aber derzeit
-

- nur eine Funktion. (Expertmodus)
- 1.32 AView angepaßt an die neuen Prefs angepaßt.
- 1.33 Kleinen Fehler im Menüpunkt Öffnen behoben.
- 1.34 Die Titelzeile des Bildschirms zeigt jetzt Informationen über den Benutzer an.
- 1.35 = 1.34 (Fehler in den Sprachdateien behoben)
- 1.36 Interne Überarbeitung der Dateierkennung.
- 1.37 Binary-Kommando eingebaut.
- 1.38 AViewPrefs erkennt jetzt Fehler in den Datei-Identifizierern.
- 1.39 Binary/Ascii Schnellunterscheidung eingebaut.
Nach einer Fehlermeldung aktiviert sich automatisch das Identifizierer-Gadget.
- 1.40 AVDefIcons liegt jetzt dem Archive bei. Es erlaubt die automatische Zuordnung von Dateien und Icons. (z.B spezielles Icon für Lha)
- 1.41 Kataloge gekürzt
Kleinen Fehler in der Sortierroutine der AView.library behoben.
Installerskript erlaubt nun die Deinstallation und das (De)installieren von AVDeficons
- 1.42 = 1.41 (AVDeficons 1.1 eingefügt)
- 1.43 AVDefIcons paßt den Piktogrammtyp automatisch an. Dadurch werden Projekt-Piktogramme automatisch zu Drawer/Disk-Piktogrammen. EnforcerHit entfernt.
- 1.44 AViewPrefs übergibt jetzt die voreingestellten Prefs automatisch an die Library. (gemeldet von Hans-Joachim Reinsch)
Deinstaller berichtigt. (gemeldet von Tony Belding)
def_File-Piktogramme beigelegt.
- 1.45 AView erlaubt jetzt MultiSelect im ASL Requester.
AVDeficons wurde ein klein wenig verbessert, und ist jetzt noch kompatibler.
- 1.46 Bugfix: AView hatte seit 1.45 Probleme mit Pfaden.
(gemeldet Jerome Chesnot)

1.12 Zukünftiges

Zukünftiges

\textdegree{} Keine Ahnung (-8 Ich richte mich da voll nach Ihren Wünschen.

1.13 Bugliste

Buglist

\textdegree{} Mir ist gemeldet worden, daß AViewPrefs auf einem 68060 schon beim Start abstürzt. Auf einem 68000/68030/68040 läuft es, daß habe ich persönlich getestet. Ich kann die Probleme bei einem 68060 nicht nachvollziehen, daher bitte ich jeden, der einen solchen Prozessor hat, sich bei mir zu melden und mir sein Resultat mitzuteilen.

\textdegree{} Derzeit sind ansonsten keine Fehler bekannt

Sollten Fehler auftreten, oder Formate, die Sie benutzen, nicht in AView vorhanden sind, dann senden Sie mir diese zu und ich werde diese einbauen. Bedenken Sie, daß ich mindestens 2 verschiedene Dateien benötige, damit ich die Identifizierer korrekt einbauen kann.

1.14 Danksagungen

Danksagungen

Timo Hegemann : Beta und Enforcertest

Andreas Vierkant : Betatest

Jerome Chesnot : für die fanzösischen Kataloge.

Andreas Boerner : Für die vielen Tips, Vorschläge und Bugreports.

Steffen Koch : Für die vielen Vorschläge.

Und natürlich allen, die mich finanziell für dieses Tool entlohnt haben.

1.15 Der Autor

Der Autor

Sollten Sie Bugs in AView finden, so bitte ich um Mitteilung, damit ich diese in neuen Versionen beseitigen kann.

Auch für Anregungen zur Erweiterungen von AView bin ich jederzeit dankbar. Besonders Formate sind willkommen.

Guido Mersmann
Glatzer Straße 12
48477 Hörstel
Deutschland

FIDONET : 2:2449/246.15
INTERNET: geit@studST.FH-Muenster.DE
TEL : 05978-225
FAX : 05978-705

1.16 Andere Programme von mir !

Andere Programme von mir !

BoulderDäsh [game/jump/boulderdaesh.lha]

Ein Klone des Original Boulderdäsh für den Amiga. Dies ist die einzige Version, die auf dem Amiga schneller ist als das Original. Sie sieht genauso aus und verhält sich genauso. Ich wurde schon gefragt, wo sich der C64 befindet. (-8 Das alles schon auf einem 68000 ohne Fastmem. Grafikkartenbenutzer wird freuen, daß dieses Spiel uneingeschränkt auf Grafikkarten funktioniert.

SimpleCat [dev/misc/simplecat.lha]

Dieses Tool ist für Programmierer und Anwender. Mit diesem Programm können die auf einfachste Weise mit ihrem Lieblings Texteditor Kataloge von Programmen ändern. Diese Programme müssen allerdings ein CS-File beiliegen haben. Sehen Sie dazu im Archiv in der Catalogs-Schublade nach.

AFind [util/misc/afind.lha]

Dieses Programm ermöglicht es ihnen auf einfachste Weise ihre AminetCDs zu durchsuchen. Es funktioniert eigentlich wie das Originaltool von der CD, nur werden hier AminetCDs und AminetSets gemischt benutzt und man ist z.B nicht gezwungen die ersten 5 Aminets zu durchsuchen, um ein Programm zu finden, das auf Set1 Diskb liegt.

AView [util/misc/aview.lha]

AView ist ein Programm, das eigentlich wie Multiview funktioniert. Es arbeitet allerdings ohne Datentypen, sondern benutzt eigene Definitionen. Man kann für jeden Dateityp ein eigenes Programm angeben, daß benutzt werden soll.

Guideformat [text/edit/guideformat.lha]

GuideFormat ist ein kleines Tool, das Blocksatz auch bei AmigaGuide Dateien ermöglicht. Dabei ist es egal, ob in der Zeile "Links" oder sonstige Kommandos vorkommen.

SiedlerBoot [game/patch/siedlerboot.lha]

Dieses Programm erlaubt das Spielen von Siedler direkt von der Amiga Plus Sonderheft 9 CD. Das geht auch ohne? Klar, aber die Spielstände können nicht gespeichert werden. Mit SiedlerBoot befindet sich nur ein kleines Programm und die Spielstände auf der Platte. Der Rest bleibt auf CD.

1.17 Technische Informationen über AView

Technische Information über AView

AView ist zu 100% in Assembler geschrieben.

Der Sourcecode benötigt 234320 bytes und 34 files. AView wurde mit DevPac 3.04 assembliert. DevPac benötigt dazu jedesmal 7 Sekunden und 755150 KB

AView wurde auf einem A2000-030/25Mhz (9 MEG) programmiert.

Ich habe diesen Guide mit Golded 4.x geschrieben. Die Faltefunktionen machen ihn dazu ideal. Damit der Blocksatz auch mit "Links" zum Kinderspiel wird, benutze ich GuideFormat um die Blöcke zu formatieren.

1.18 Die Befehle von AView

Die Befehle von AView

Die Identifiziererzeile in den Voreinstellungen ist eine Liste von sehr einfachen Befehlen, die eine Erkennung der Datei möglich machen.

Generell

\textdegree{} Kommandos müssen durch mindestens ein Leerzeichen getrennt sein.

\textdegree{} Kommandonamen sind Groß/Kleinschrift unabhängig. "MATCH" = "MaTcH"

\textdegree{} Alle Kommandos haben nur eine Wirkung auf den von AView ←
eingeleseenen
Teilbereich einer Datei. Das sind immer die ersten 8KB einer Datei.
Wird ein Bereich außerhalb des Bereiches überprüft, so ist die
Bedingung automatisch nicht erfüllt. Das gilt auch, wenn die Datei
kleiner ist als 8KB

\textdegree{} Die Gleichungen entsprechen immer dem mathematischen Sinn, aber es ←
gibt
eine kleine Änderung. Findet AView als erstes Kommando ein ASCII oder
ein BINARY, so wird dieses Kommando unabhängig vom Rest der Gleichung
ausgeführt.

Eine Liste aller Kommandos

MATCH	Bytes oder String an bestimmter Stelle prüfen
FIND	Bytes oder String in einem Bereich suchen
NAME	Namen der Datei prüfen
FILESIZE	Dateilänge prüfen
ASCII	Datei auf ASCII prüfen

BINARY	Datei auf BINARY prüfen
AND	Und-Verknüpfung
OR	Oder-Verknüpfung
NOT	Nicht-Verknüpfung
()	Klammern

1.19 Der Befehl: MATCH

MATCH

TEMPLATE:

MATCH [Start,] Zeichenfolge

KURZBESCHREIBUNG:

Dieses Kommando erlaubt das Prüfen von Bytes oder Strings ab einer bestimmten Stelle der Datei.

BEISPIELE:

MATCH "Hallo"

Prüft, ob die Datei mit dem Wort "Hallo" beginnt.

MATCH 1,"Hallo"

Prüft, ob die Datei an der Position "1" das Wort "Hallo" enthält.

MATCH 27,\$1234

Hier wird geprüft ob an der Position "27" die Bytes \$12 und \$34 folgen.

MATCH 0,"FORM????ILBM"

Die Datei muß an der Position "0" das Wort "FORM" haben, die nächsten 4 Zeichen sind egal und jetzt muß ein "ILBM" folgen, um die Bedingung zu erfüllen.

WICHTIG:

Die Positionsangabe ist optional und immer Null, wenn es nicht angegeben wird.

Die Positionsangabe darf nur Dezimal sein.

Die Zeichenfolge muß mit einem Anführungszeichen beginnen und enden oder aber durch ein "\$" eingeleitet werden.

Eine Stringzeichenfolge kann "?" enthalten. Diese Bytes werden nicht

beachtet.

1.20 Der Befehl: FIND

FIND

TEMPLATE:

FIND [erster Suchbegin,] [letzter Suchbegin,] Zeichenfolge

KURZBESCHREIBUNG:

Dieses Kommando sucht nach einer Byte- oder Zeichenfolge innerhalb eines bestimmten Bereichs der Datei.

BEISPIELE:

FIND "Hallo"

Sucht den ganzen Buffer nach "Hallo" ab.

FIND 10,"Schlumpf"

Sucht von Position "0" bis zur Position "10" den Buffer nach "Schlumpf" ab.

FIND 1,10,"Smurf"

Sucht zwischen 1 und 10 nach dem Wort "Smurf". Die 10 bestimmt nur den letzten Suchstart. Das heißt Byte 10 der Datei wird mit dem "S" verglichen, Byte 11 mit dem "m" und so weiter.

FIND 1,127,\$0001

Hier wird von Position "1" bis "127" nach dem Bytes \$00 und \$01 gesucht.

WICHTIG:

Die Positionsangaben sind optional. Wird nur eine angegeben, so handelt es sich um das Suchende. Wenn keine Angabe gemacht wird, so sucht AView im kompletten Buffer.

Beide Positionsangaben müssen dezimal sein.

Die Zeichenfolge muß mit einem Anführungszeichen beginnen und enden oder aber durch ein "\$" eingeleitet werden.

1.21 Der Befehl: NAME

NAME

TEMPLATE:

NAME Pattern

KURZBESCHREIBUNG:

Dieses Kommando vergleicht den Dateinamen mit dem anzugebenen Pattern.

BEISPIELE

NAME "#?.gif"

Dieses Kommando ist nur erfüllt, wenn die Datei z.B. "Schlumpf.gif" heißt.

NAME "#?.(jpeg|jpg)"

Nur Dateien, die mit .jpeg oder .jpg enden, erfüllen die Bedingung.

WICHTIG:

Das Pattern muß mit einem Anführungszeichen beginnen und enden.

1.22 Der Befehl: FILESIZE

FILESIZE

TEMPLATE:

FILESIZE Number

KURZBESCHREIBUNG:

Dieses Kommando vergleicht die Dateilänge mit dem angegebenen Wert.

BEISPIELE

FILESIZE 901120

Dieses Kommando ist nur erfüllt, wenn die Datei genau 901120 Bytes lang ist.

```
FILESIZE 1024 AND NAME "BootBlock#?"
```

Hier muß die Datei genau 1024 Bytes lang sein und der Name der Datei muß mit "BootBlock" beginnen.

WICHTIG:

Die Zahl muß in Dezimal angegeben werden.

1.23 Der Befehl: ASCII

```
ASCII
```

```
-----
```

TEMPLATE:

```
ASCII
```

KURZBESCHREIBUNG:

Dieses Kommando vergleicht alle Zeichen im Buffer und prüft ob der Buffer nur lesbare Zeichen enthält.

BEISPIEL:

```
ASCII
```

Dieses Kommando ist nur erfüllt, wenn der Dateibuffer keine unlesbaren Zeichen enthält, also eine normale Textdatei ist.

Wichtig:

Um die Erkennungsgeschwindigkeit zu steigern, haben ASCII und BINARY eine spezielle Eigenschaft, wenn sie als erstes Kommando benutzt werden. Dann nämlich wird auf ASCII oder BINARY getestet und im Falle eines negativen Ergebnisses wird die weitere Überprüfung nicht durchgeführt:

```
"ASCII AND FILESIZE 123443"
```

ASCII steht vorne, daß bedeutet wenn die Datei BINARY ist, wird die Gleichung nicht gelöst, sondern gleich abgebrochen.

Wo das Problem liegt? Nun, eigentlich dürfte es keine Probleme geben, aber denkbar wäre eine folgende Gleichung:

```
"ASCII OR FILESIZE 123443"
```

Durch die Sonderfunktion wird FILESIZE im Falle einer Binärdatei nicht geprüft, obwohl es wegen dem OR gemacht werden müßte. Sollte eine

solche Gleichung wirklich mal nötig sein, dann einfach das ASCII nach hinten schreiben.

1.24 Der Befehl: BINARY

BINARY

TEMPLATE:

BINARY

KURZBESCHREIBUNG:

Dieses Kommando vergleicht alle Zeichen im Buffer und prüft ob der Buffer nicht lesbare Zeichen enthält.

BEISPIEL:

BINARY

Dieses Kommando ist nur erfüllt, wenn der Dateibuffer unlesbare Zeichen enthält, also keine normale Textdatei ist.

Wichtig:

Um die Erkennungsgeschwindigkeit zu steigern, haben ASCII und BINARY eine spezielle Eigenschaft, wenn sie als erstes Kommando benutzt werden. Dann nämlich wird auf ASCII oder BINARY getestet und im Falle eines negativen Ergebnisses wird die weitere Überprüfung nicht durchgeführt:

"BINARY AND FILESIZE 123443"

BINARY steht vorne, daß bedeutet wenn die Datei ASCII ist, wird die Gleichung nicht gelöst, sondern gleich abgebrochen.

Wo das Problem liegt? Nun, eigentlich dürfte es keine Probleme geben, aber denkbar wäre eine folgende Gleichung:

"BINARY OR FILESIZE 123443"

Durch die Sonderfunktion wird FILESIZE im Falle einer Textdatei nicht geprüft, obwohl es wegen dem OR gemacht werden müßte. Sollte eine solche Gleichung wirklich mal nötig sein, dann einfach das BINARY nach hinten schreiben.

1.25 Der Befehl: AND

AND

TEMPLATE:

AND

KURZBESCHREIBUNG:

Dieses Kommando erlaubt das Und-Verknüpfen von verschiedenen Kommandos.

BEISPIELE:

NAME "#?.txt" AND ASCII

Diese Bedingung ist nur erfüllt, wenn es sich um eine Textdatei handelt und der Dateiname mit ".txt" endet.

NAME "#?.txt" & ASCII

Dieses Beispiel hat die gleiche Wirkung wie das Obige.

NAME "#?.guide" & ASCII AND FIND 40,"@Database"

Der Dateiname muß auf ".guide" enden, die Datei muß eine Textdatei sein und innerhalb der ersten 40 Bytes muß das Wort "@Database" beginnen.

WICHTIG:

Eine Und-Verknüpfung kann auch durch "&" abgekürzt werden.

Eine Und-Verknüpfung hat immer Vorrang vor einer Oder-Verknüpfung, wenn die Oder-Verknüpfung nicht durch Klammern bevorzugt behandelt wird.

1.26 Der Befehl: OR

OR

TEMPLATE:

OR

KURZBESCHREIBUNG:

Dieses Kommando erlaubt das Oder-Verknüpfen von verschiedenen Kommandos.

BEISPIELE:

```
NAME "#?.txt" OR ASCII
```

Diese Bedingung ist nur erfüllt, wenn es sich um eine Textdatei handelt oder der Dateiname mit ".txt" endet.

```
NAME "#?.txt" | ASCII
```

Dieses Beispiel hat die gleiche Wirkung wie das Obige.

```
NAME "#?.guide" | ASCII OR FIND 40,"@Database"
```

Der Dateiname muß auf ".guide" enden oder die Datei muß eine Textdatei sein oder innerhalb der ersten 40 Bytes muß das Wort "@Database" beginnen.

WICHTIG:

Eine Oder-Verknüpfung kann auch durch "|" abgekürzt werden.

Eine Und-Verknüpfung hat immer Vorrang vor einer Oder-Verknüpfung, wenn die Oder-Verknüpfung nicht durch Klammern bevorzugt behandelt wird.

1.27 Der Befehl: NOT

NOT

TEMPLATE:

NOT

KURZBESCHREIBUNG:

Dieses Kommando erlaubt das negieren von Operationen.

BEISPIELE:

```
NAME "#?.hft" AND NOT ASCII
```

Diese Bedingung ist nur erfüllt, wenn es sich um keine Textdatei handelt und der Dateiname mit ".hft" endet.

```
NAME "#?.hft" AND ~ ASCII
```

Dieses Beispiel hat die gleiche Wirkung wie das Obige.

```
~ (NAME "#?.data" & ASCII)
```

Der Dateiname darf nicht auf ".data" enden und die Datei darf kein ASCII sein.

WICHTIG:

Eine NOT-Verknüpfung kann auch durch "~" abgekürzt werden.

Eine NOT-Verknüpfung hat immer Vorrang vor einer Und und einer Oder-Verknüpfung, wenn diese Verknüpfungen nicht durch Klammern bevorzugt behandelt wird.

=> "NOT NAME "Hallo" AND ASCII" = "(NOT NAME "Hallo) AND ASCII"

1.28 Der Befehl: ()

(und)

TEMPLATE:

(
)

KURZBESCHREIBUNG:

Klammer erlauben das Verbinden von Bedingungen exakt im mathematischen Sinne.

BEISPIELE:

NAME "#?.guide" AND (ASCII OR FIND 40,"@Database")

Der Dateiname muß auf ".guide" enden und entweder muß die Datei eine Textdatei sein oder aber sie muß innerhalb der ersten 40 Bytes das Wort "@Database" beinhalten.

Dieses Beispiel entspricht dem Mathematischen "a*(b+c)".

WICHTIG:

Es müssen immer gleich viele offene wie auch geschlossene Klammern vorhanden sein.

Die Gleichung wird immer von der innersten Klammer nach außen gelöst. Klammern werden also immer als erstes gelöst.

1.29 Das Pattern: Name/M

Name/M

"Name" bestimmt den Namen der Datei, die AView erkennen soll. Das /M bedeutet, daß beliebig viele Dateinamen angegeben werden können. Zusätzlich kann der Name DOS-Pattern enthalten, also z.B. "#?.gif"

Beispiele:

AView #?.gif

Zeigt alle Dateien an, die mit ".gif" enden.

AView Smurf.jpg PapaSchlumpf.gif #?smurf#?

Dieser Aufruf zeigt die Dateien "Smurf.jpg" "PapaSchlumpf.gif" und alle Dateien, die das Wort "smurf" enthalten. Ist Ihnen was aufgefallen? Genau! Die erste Angabe wird auch durch die letzte Angabe erfüllt. Die Datei "Smurf.jpg" wird zweimal angezeigt.

Natürlich können auch Pfade angegeben werden.

AView sys:pictures/smurfs/Papa.gif sys:Pictures/old/A500.pic

AView setzt hier keinerlei Grenzen.

1.30 Das Pattern: All/S

All/S

"All" ermöglicht die Suche in allen Unterverzeichnissen.

1.31 Das Pattern: ARG=To=From/K

ARG=To=From/K

"ARG", "To" oder "From" sind besonders nützlich. Diese Angabe ersetzt den in AViewPrefs angegebenen %a String durch den hier angegebenen.

Wenn Sie zum Beispiel in den AViewPrefs den Lha anwählen, dann zeigt die Kommandozeile folgenden Inhalt:

"LHA x -M -X %n %a"

%n wird durch den Namen der Datei ersetzt, den wir bei Name angegeben haben. %a wird defaultmäßig durch "T:" ersetzt. Also entpackt AView eine Datei immer nach "t:", wenn kein "ARG" angegeben wird. Das Default von %a kann für jeden Befehl einzeln in den AViewPrefs eingestellt werden.

Wenn wir aber "AView Dummy.lha ARG ram:" angeben, dann wird ins RAM: entpackt.

Wichtig ist, das immer eines der drei Schlüsselwörter (ARG, To oder From) angegeben wird, wenn Sie eine Änderung von %a wünschen.

Natürlich hat "From" die selbe Funktion, wie "To". Ich habe nur beide Varianten und den neutralen ARG Typ gewählt, damit man nicht auf unsinnig anhörende Kommandos angewiesen ist. (s.u)

Beispiele:

AView BoulderDäsh.lha to ram:

Dieses Beispiel entpackt das Archive "BoulderDäsh.lha" ins ram: Verzeichnis.

AView BoulderDäsh.lha from ram:

Macht genau das gleiche, wie das obige Beispiel. Aber es sieht etwas dümmlich aus, weil wir ja nicht vom RAM: (from) holen, sondern etwas dahin schreiben.

1.32 Das Pattern: SetComment=SC/S

SetComment=SC/S

"SetComment" oder "SC" schaltet vom Ausführungsmodus in den Markierungsmodus. Alle Dateien werden untersucht und das Ergebnis wird in den Filekommentar geschrieben. Es werden keine Kommandos gestartet.

1.33 Das Pattern: ShowDetect=SD/S

ShowDetect=SD/S

"ShowDetect" oder kurz "SD" zeigt im CLI die entsprechenden Dateien an. Es wird ausgegeben, welche Datei als was erkannt wurde. Es werden keine Kommandos gestartet.

1.34 Das Pattern: Debug/S

Debug/S

"Debug" gibt genaueren Aufschluß, warum bestimmte Dateien nicht erkannt wurden. Dieser Schalter ist eigentlich für den Normalanwender nicht vonnöten. Derzeit werden nur unbekannte Kommandos und Syntaxfehler

ausgegeben.

1.35 Die Voreinstellungen von AView

Die Voreinstellungen

Das Voreinstellungsprogramm erlaubt Ihnen die gezielte Einstellung aller Dateiformate und deren Aktionen.

Sortieren nach
Name
Auswahl
Löschen
Alle
Neu
Speichern
Benutzen
Abbrechen

1.36 Gadget: Sortieren nach

Sortieren nach

Dieses Gadget ist nur im Experten Modus verfügbar.

Dieses Gadget erlaubt die Sortierung der Liste aller Identifizierer. Normalerweise werden alle Einträge nach dem Namen sortiert.

AView bearbeitet die Liste aber nach der Reihenfolge, die durch die Unterklassen definiert wird.

Die Sortierung nach der Unterklasse ist hilfreich, wenn man eigene Dateitypen erstellt hat und sehen will, warum der eigene Eintrag nicht erkannt wird.

1.37 Gadget: Name

Name

Diese Liste enthält alle von AView erkannten Dateiformate. Benutzen Sie Auswahl und gegebenenfalls New, um Dateitypen zu erstellen.

Mit den Cursortasten kann durch die Einträge gescrollt werden. "Return" oder ein Doppelklick mit der Maus öffnen das Editier-Fenster, das die Konfiguration der Befehle ermöglicht.

1.38 Gadget: Auswahl

Auswahl

Hier öffnet sich das Auswahlfenster. Es ermöglicht eine gezielte Auswahl der schon vorhandenen Identifizierer. Der neue Eintrag wird an die Liste angehängt.

1.39 Gadget: Löschen

Löschen

Löscht den angewählten Identifizierer.

1.40 Gadget: Alle

Alle

Nach einer Sicherheitsabfrage werden alle noch nicht in der Liste stehenden Einträge aus der internen Liste zu übernommen.

Bereits vorhandene Einträge werden nicht modifiziert.

1.41 Gadget: Neu

Neu

Dieses Gadget ist nur im Experten Modus verfügbar.

Hier können Sie eigene Identifizierer erzeugen. Es öffnet sich ein Fenster und sie können einen Namen eingeben, der nur für Sie zur Erkennung in der Liste nötig ist.

1.42 Gadget: Speichern

Speichern

Speichert die Einstellungen und beendet die Voreinstellungen.

1.43 Gadget: Benutzen

Benutzen

Die Einstellungen werden nicht dauerhaft gespeichert und gehen nach einem Reset oder abschalten des Rechners verloren, da sie im "ENV:" Verzeichnis liegen.

1.44 Gadget: Abbrechen

Abbrechen

Ohne die Veränderungen zu speichern werden die Voreinstellungen beendet.

1.45 Das Editierfenster

Das Editierfenster

In diesem Fenster werden alle Einstellungen gemacht, die Sie für diesen einen Dateityp wünschen.

Name
Klasse
Unter Klasse
Ausführungs Typ
Stack
Kommando
Aktueller Typ
Aktuelles Verzeichnis
WB Ausgabedatei
CLI Ausgabedatei
Identifizierer
Vordefiniertes %a

1.46 Gadget: Name

Name

Hier geben Sie den Namen an, den der Dateityp bekommen soll. Dieser Eintrag sollte nicht mehrfach vergeben werden, da er das Finden von Einstellungsfehlern erleichtert.

1.47 Gadget: Klasse

Klasse

Dieses Gadget ist nur im Experten Modus verfügbar.

Hier geben sie einfach den Typ an, den dieser Dateityp verkorpert. Sollten Sie keinen passenden finden, dann greifen sie auf den Eintrag "User" zurück.

Derzeit wird diese Angabe nur für Ausgaben und die Sortierung benutzt. Geben Sie die Klasse denoch korrekt an, damit Sie keine Probleme bekommen. ↔

1.48 Gadget: Unter Klasse

Unter Klasse

Dieses Gadget ist nur im Experten Modus verfügbar.

Dieses Gadget ist mit das wichtigste überhaupt. Es bestimmt, in welcher Reihenfolge AView die Dateitypen behandeln soll.

Es ist einzusehen, das eine Datei, die nur anhand ihrer Dateiendung erkannt wird, wesentlich unsicherer erkannt wird, als eine Datei, die sich durch ihr Format beschreibt.

Daher sollten nur durch Namen identifizierte Dateien immer am Ende geprüft werden. Schließlich kann es ja passieren, daß eine LHA-Datei "hallo.c" heißt. Eine Erkennung auf den Dateinamen "#?.c" würde diese erkennen und z.B dem C-Compiler übergeben, der damit nicht anfangen kann. Hätte man erst den Lha geprüft, dann wäre die Datei nur entpackt worden.

Ich habe hier einige dieser Vorschriften bereits benannt:

Benutzer:	Wird immer eingestellt, wenn Sie einen neuen Dateityp erzeugen. Sollte aber nicht weiter benutzt werden und auf einen der folgenden eingestellt werden.
Normal:	Sollte eigentlich fast immer ausreichen.
Mittel:	Diesen habe ich verwendet, wenn Gruppen zusammengefaßt werden mußten, (Kickstart - Global, Icon - Global) da es nicht viel Sinn macht einzelne Kickstarts nach diesen globalen Prüfungen zu testen.
xxx oder Name	Dieser Typ sollte verwendet werden, wenn es sich um eine ODER des Names handelt: z.B "(FIND "<html>" OR NAME "#?.htm(l)") AND ASCII" Wenn die Datei kein "<html>" enthält, ist die Erkennung genauso gut wie "Name and ASCII" aber ansonsten ist sie besser.
Name und ASCII:	Diese Erkennung ist nicht besonders gut, aber immer noch besser, als die Folgende. Also immer nur

verwenden, wenn die Datei nach dem Namen und ASCII beurteilt wird. Also ziemlich weit hinten, da ASCII mit Inhaltsbezogenem Text als Normal eingestuft sind.

Nur Name: Wie gesagt die Erkennung nur nach dem Namen ist sehr schlecht, wird also weit hinten durchgeführt.

Die anderen Unterklassen sollten Sie nicht benutzen. Wenn wichtige Dateitypen fehlen, dann schicken Sie mir doch einfach die von ihnen erstellte Prefs-Datei.

Kleiner Tip am Rande. Mit der Öffnen-Funktion des Menüs können Sie auch das "Data/AView.Data"-File einladen und mit der Speichern als Funktion wieder speichern. So können Sie die Default-Einstellungen selber ändern.

Dies ist aber nur zu empfehlen, wenn Sie sich mit dem AView-System intensiv vertraut gemacht haben. Außerdem sollten Sie ein Backup ihres original DatenFiles machen.

1.49 link

Ausführungs Typ

Hier können Sie wählen auf welche Art AView ihre Programme starten soll. Normalerweise ist CLI angebracht. Einige Programme bestehen auf einen Start per Icon, weil Sie aus dem Tooltype einige Informationen auslesen wollen.

1.50 Gadget: Stack

Stack

Hier können Sie den Stack einstellen, der dem zustartenden Programm zur Verfügung steht. Normalerweise reichen 8192 Bytes aus.

Sollte ein Programm abstürzen, so sollten Sie den Wert erhöhen.

1.51 Gadget: Kommando

Kommando

Wenn ein Dateityp erkannt wird, wird dieses Kommando gestartet. Es können folgende Angaben in den Kommandostring eingefügt werden.

%p : Fügt den Pfad der Datei ein. (z.B "SYS:Pictures/")

%~p : Wie %p nur erzeugt AView hier keine Anführungszeichen.

`%n` : Der Dateiname wird eingefügt. (z.B. "Smurf.gif")

`%~n` : Wie `%n` nur erzeugt AView hier keine Anführungszeichen.

`%f` : Der Dateiname wird mit dem kompletten Pfad eingefügt.
(z.B. "SYS:Pictures/Smurf.gif")

`%~f` : Wie `%f` nur erzeugt AView hier keine Anführungszeichen.

`%a` : Hier wird das "Arg" aus dem DosPattern übernommen. Wenn dort nichts angegeben wird, dann trägt AView hier das Default Argument aus den AViewPrefs ein.

`%~a` : Wie `%a` nur erzeugt AView hier keine Anführungszeichen.

Eigentlich gibt nur vier `%` Kommandos (`p,n,f,a`). Ein `~` nach dem Prozent schaltet nur die Anführungszeichen aus.

Beim Ausführungs Typ "WB" werden bis auf `%a` und `%~a` alle "`%`" Sequenzen entfernt ohne Sie zu beachten.

Wenn Sie ein Prozentzeichen benötigen, dann schreiben Sie einfach `%%!` AView wird diese beiden Prozentzeichen in ein einzelnes Umwandeln, ohne es als Kommando zu verstehen.

1.52 Gadget: Aktueller Typ

Aktueller Typ

Hier wird die Einstellung vorgenommen, die als `CurrentDir` bezeichnet wird. Einfacher gesagt funktioniert es wie ein `CD`-Kommando, das Sie von Dos her kennen.

Dateiverzeichnis: Es wird ein `CD` ins Verzeichnis gemacht, in der die an AView übergebene Datei liegt.

Aktuelles : Das Verzeichnis wird aus dem CLI übernommen.

Verz. angeben : Das Verzeichnis kann im folgenden Gadget angegeben werden.

1.53 Gadget: Aktuelles Verzeichnis

Aktuelles Verzeichnis

Dieses Gadget ist nur benutzbar, wenn im Gadget `Aktueller Typ` der Modus "Verzeichnis angeben" eingestellt ist.

1.54 Gadget: WB Ausgabedatei

WB Ausgabedatei

Ausgaben, die vom zustartenden Programm gemacht werden, werden hier hin umgeleitet. Das passiert aber nur, wenn AView über die Workbench gestartet wurde. Wenn Sie also "Con:" eintragen, so wird sich jedesmal wenn AView dieses Programm startet ein Shell-Fenster geöffnet.

1.55 Gadget: CLI Ausgabedatei

CLI Ausgabedatei

Dieses Gadget hat die gleiche Funktion wie WB Ausgabedatei. Nur wird diese Datei immer benutzt, wenn AView von der Shell gestartet wurde. Ist dieses Gadget leer, so wird immer das aktuelle Shell-Fenster benutzt.

1.56 Gadget: Identifizierer

Identifizierer

Hier wird die Erkennung eingestellt. Dazu gibt es einige Befehle, die ein eindeutiges Erkennen erst möglich machen.

1.57 Gadget: Vordefiniertes %a

Vordefiniertes %a

Dieser Eintrag wird immer für %a benutzt, wenn im Dos-Kommando kein spezielles Argument angegeben wurde.

1.58 Das Auswahlfenster

Das Auswahlfenster

Dieses Fenster zeigt alle Dateiformate an, die das AView_Prefs Programm bereits eingebaut hat.

Die Liste ist nach den Dateiklassen eingeteilt und bietet so eine schnelle Konfiguration von AView.

Benutzen sie die Cursortasten, um einen Eintrag auszuwählen und bestätigen Sie die Wahl durch Betätigen von Return. Natürlich können Sie auch die Maus benutzen und einen Eintrag doppelklicken.

Die Gadgets "Ok" und "Abbrechen" sollten eigentlich klar sein. "Ok" Übernimmt den angewählten Eintrag und "Abbrechen" sowie auch das Schließsymbol des Fensters brechen den Vorgang ab.

1.59 Die Voreinstellungen der Voreinstellungen

Die Voreinstellungen der Voreinstellungen

Es ist einfacher als es sich anhört. Hier können einige Dinge verstellt werden, die nur dieses Voreinstellungsprogramm betreffen und für AView selber bedeutungslos sind.

Der Expertenmodus
Speichern
Benutzen
Abbrechen

1.60 Gadget: Expertenmodus

Expertenmodus

Dieses Gadget sollten Sie nur anwählen, wenn Sie sich mit Dateierkennung auskennen, bzw mindestens die komplette Anleitung gelesen.

Sie sollten das Prinzip der prioritäts abhängigen Dateierkennung begriffen und deren Problematik erkannt haben.

Es ist selbst für mich nicht manchmal nicht so ganz einfach einen Fehler in den vordefinierten Dateitypen zu beheben.

Dieser Schalter läßt eine Reihe von Gadgets erscheinen, die für den Normalanwender nicht nötig sind.

1.61 Gadget: Speichern

Speichern

Speichert die Einstellungen und beendet die Voreinstellungen.

1.62 Gadget: Benutzen

Benutzen

Die Einstellungen werden benutzt, aber nicht gespeichert.

1.63 Gadget: Abbrechen

Abbrechen

Ohne die Veränderungen zu speichern werden die Voreinstellungen beendet.

1.64 Die Menüs von AViewPrefs

Die Menüs von AViewPrefs

AViewPrefs enthält die folgenden Menüpunkte:

Projekt

- Öffnen...
- Speichern Als...
- Beenden

Default

- Reset to Default
- Last Saved
- Restore

Preferences

- AView Prefs

1.65 Menu: Öffnen

Öffnen

"Öffnen" erlaubt das Laden einer beliebigen AView Voreinstellungsdatei.

Die gerade im Editor gemachten Einstellungen an den Dateitypen gehen verloren.

1.66 Menu: Speichern Als

Speichern als

Dieser Menüpunkt erlaubt das Speichern der Voreinstellungen in eine beliebige Datei.

1.67 Menu: Beenden

Beenden

AViewPrefs wird beendet. Die gemachten Einstellungen gehen verloren.

1.68 Menu: Reset to Default

Reset To Default

AViewPrefs löscht die bestehenden Einträge und erzeugt einige vordefinierte Typen. Diese Funktion wird auch ausgeführt, wenn AViewPrefs beim Programmstart keine Voreinstellungsdatei findet.

1.69 Menu: Last Saved

Last Saved

AViewPrefs lädt die zuletzt auf Festplatte gespeicherten Voreinstellungen wieder ein.

1.70 Menu: Restore

Restore

Diese Funktion lädt die zuletzt benutzen Voreinstellungen wieder ein.

1.71 Menu: AView Prefs

AView Prefs

Dieser Menüpunkt öffnet die AViewPrefs Voreinstellungen.

1.72 AVDeficons

AVDeficons

Was macht AVDeficons?
 Was kann AVDeficons?
 Installation von AVDeficons
 Wie definiere ich die neuen Piktogramme?
 Probleme durch AVDeficons

1.73 Was macht AVDeficons?

Was macht AVDeficons?

AVDeficons ist ein kleines, intern aber sehr kompliziertes Programm.

Es erlaubt das Definieren von Piktogrammen für jeden Dateityp, den AView kennt, darüber hinaus erlaubt es noch zusätzliche Features, die Sie bestimmt gut gebrauchen können.

1.74 Was kann AVDeficons?

Was kann AVDeficons?

\textdegree{} Keine Konfiguration nötig! Nur die Icons erstellen und fertig.

\textdegree{} Default Piktogramme für jeden Dateityp und dessen Klasse.

\textdegree{} Default Piktogramme für jedes Verzeichnis.

\textdegree{} Default Piktogramme für Disketten basierend auf: Filesystem, Device ↔
 .

\textdegree{} Zusätzliche Diskettenpiktogramme für "BUSY", "BAD", "NDOS".

\textdegree{} Automatische Freigabe der Piktogrammpositionen.

\textdegree{} Automatische Umschaltung der Workbench auf "Zeige Icons und alle ↔
 Dateien".

\textdegree{} Schubladengrößen werden gesetzt.

1.75 AVDeficons Installation

Wenn Sie bei der Installation "AVDeficons" nicht in die User-Startup haben eintragen lassen, dann holen Sie das jetzt nach. Dazu gibt es

zwei Möglichkeiten:

1. Die Beste Möglichkeit

Sie nehmen das original Archiv und installieren AView nochmal. Dabei lassen Sie diesmal AVDefIcons in die "User-startup" eintragen. Keine Sorge Ihre Konfiguration bleibt voll erhalten. Jetzt den Rechner neustarten und AVDeficons steht zur Verfügung.

2. Für Fortgeschrittene

Nehmen Sie einen Texteditor und laden Sie die Datei "s:user-startup" ein. Dort befindet sich die Angabe ";END AView". Genau vor dieser Zeile fügen Sie "AVDeficons" ein. Jetzt den Rechner neustarten und AVDeficons steht zur Verfügung.

1.76 Wie definiere ich die neuen Piktogramme?

Wie definiere ich die neuen Piktogramme?

Die Piktogramme können Sie ganz normal mit ihrem Lieblingseditor (z.B. "IconEdit" von der Workbench) erzeugen, oder aus den diversen Sammlungen aus dem Aminet entnehmen.

Im Folgenden benutze ich immer Piktogrammnamen wie :
"disk/def_df0.info". Das bedeutet, daß dieses Piktogramm in das Piktogrammverzeichnis "disk" gehört und den Namen "def_df0.info" trägt.

Durch AVDeficons wird ihr System um drei mächtige "def_"-Typen erweitert.

Verzeichnis Piktogramme
Disketten Piktogramme
Filetype Piktogramme

1.77 Verzeichnis Icons

Verzeichnis Icons

Diese Icons sind mein absoluter Favorit. Sie kennen sicher diese schönen Verzeichnis Piktogramme, die auf jeder Schublade ein kleines Bildchen haben.

Tja, und wenn Sie ein neues Programm installieren, dann hat dieses auch gleichnamige Schubladen, aber die Icons sind wieder ohne oder haben keine Icons.

Genau hier setzt AVDeficons ein. Nehmen wir den "Libs" Ordner. Nehmen Sie das Schubladen-Piktogramm, das sie für alle "libs"

Schubladen haben wollen und kopieren es unter dem Namen "def_libs.info" in die Piktogrammschublade "Drawer".

Alle "Libs" Verzeichnisse nehmen dieses nur Aussehen an, wenn die Schublade selber kein Icon hat.

Der Aufbau ist immer der Selbe: drawer/def_Name.info, wobei Name natürlich der Name der Schublade ist.

1.78 Disketten Piktogramme

Disketten Piktogramme

Dieses Feature erlaubt die Einstellung von Icons für jedes Laufwerk und jeden Zustand der Selbigen. Der Aufbau ist generell immer der Selbe: disk/def_name.info.

AVDefIcons benutzt diese Icons nur, wenn die Medien keine Icons haben.

Für Zahlen in den Angaben können auch x für beliebig gesetzt werden.

Beispiele:

```
disk/def_busy.info ;Disk nicht bereit (df0:BUSY)
disk/def_ndos.info ;keine DOS Diskette
disk/def_badd.info ;keine lesbare Diskette
disk/def_dos2.info ;Dos Diskette mit einer "DOS2" Formatierung.
disk/def_dosx.info ;Dos Diskette mit einer beliebigen "DOS" Formatierung.
disk/def_kick.info ;Kickstart Diskette
disk/def_msd.info ;MS-DOS Disk
disk/def_cd0.info ;CDROM
disk/def_df0.info ;Diskette in "df0:"
disk/def_dfx.info ;Diskette in "df0:", "df1:", "df2:" oder "df3:"
```

1.79 Datei Piktogramme

Datei Piktogramme

Datei Piktogramme sind die Spezialität von AVDefIcons. Es nimmt dazu die vordefinierten Einträge, die auch in den AVViewPrefs zur Verfügung stehen.

Das Prinzip ist einfach und eine Konfiguration ist nicht nötig. Sie müssen lediglich die Piktogramme definieren.

Wenn die z.B. ein Piktogramm für LHA Files definieren wollen, so erzeugen sie einfach ein Icon mit dem Namen: "file/def_lha.info". Das ist alles.

Genauso verfahren Sie mit den anderen Filetypen. Die anderen Namen

erfahren Sie in den AViewPrefs.

Beachten Sie, daß Leerzeichen, "/" und ":" entfernt werden und daher nicht in den Namen für die Piktogramme eingetragen werden dürfen. Lassen Sie diese Zeichen einfach weg.

Wenn AVDeficons bei der Suche nach einem Piktogramm (z.B. "file/def_lzx.info") nicht fündig wird, so sucht er nach der Klasse. Da er LZX als Dateityp gefunden hat lautet die Klasse "Archive". Daher wird jetzt nach "file/def_archive.info" gesucht.

Wird also z.B. "file/def_guide.info" nicht gefunden, sucht AVDeficons nach dem Klassenpiktogramm, also hier "file/def_text.info".

Beachten Sie, daß der Name der Klasse immer der englische Name ist und nicht der der Übersetzung.

1.80 AVDeficons und die Probleme, die dahinter stehen!

AVDeficons und die Probleme, die dahinter stehen!

Einfach starten und ein paar Patches organisieren eben mal die nötigen Piktogramme. Das alles sieht so einfach aus. Ist es aber nicht! 6 verschiedene Patches sind nötig, damit alles so funktioniert, wie es ist.

Das größte Problem an der Sache ist, daß das System nicht die Icon.library über den JumpTable benutzt, sondern direkt im ROM springt. Das erhöht zwar die Geschwindigkeit, macht das Patchen aber zum Alptraum.

Daher kann und wird es vorkommen, daß dieser Patch bei zukünftigen ROMs nicht mehr vernünftig arbeitet. Bis zur Version 3.1 läuft jedoch alles. Speziell für Kick V2.x werden schon leichte Änderungen gemacht, damit es auch noch funktioniert.

Wer meint er kann auch so ein Programm schreiben, der soll es mal versuchen. Ich will aber auch die doch "so einfachen" DrawerIcons sehen. (-8

1.81 Probleme durch AVDeficons

Probleme durch AVDeficons

\textdegree{} AVDeficons kann nicht mit "Deficons" Programm aus dem NewIcons- Archiv gleichzeitig funktionieren. Daher entfernen Sie dieses. Ansonsten kann es zu unerwünschten Konflikten führen.

\textdegree{} Neue Kickstarts können AVDeficons unbrauchbar machen. (größer 3.1)

\textdegree{} Das "Drag and Drop" kopieren von Verzeichnissen läßt ein eventuell ←
 von
 AVDeficons erzeugtes Verzeichnis Piktogramm verschwinden. Nach dem
 Updaten des Zielfensters erscheint es aber wieder. Für dieses Problem
 gibt es derzeit keine Lösung.

Wenn in einer Schublade ein Fenster geöffnet wird und man die erste
 Schublade aktualisieren läßt, dann hängen die von AVDeficons erzeugten
 Piktogramme alle an den falschen Dateien. Für dieses sehr unschöne
 Problem gibt es derzeit leider auch keine Lösung. Man wird aber nur
 sehr selten in diese Situation kommen. Ansonsten hilf nur das Schließen
 der Tochter Schubladen und dann die Schublade aktualisieren.

\textdegree{} Es erscheinen nicht die gewünschten Piktogramme:

Falls Sie einen Piktogrammpfad angegeben habe, prüfen Sie ob dieser
 korrekt ist und ob die Unterverzeichnisse auch den richtigen Namen
 haben. (Drawer, Disk, File)

Wenn das auch nichts hilft, so könnte ein höherwertiges benutzt werden.
 "disk/def_dfx.info" kann nicht für df0: erscheinen, wenn es ein
 "disk/def_df0.info" gibt. Die Disk darf natürlich auch kein "disk.info"
 haben. "Disk/def_dos0.info z.B hat auch Vorrang vor beiden obigen
 Piktogrammen.

Das gleiche gilt für Dateipiktogramme. Das "file/def_archive.info"
 Piktogramm kann nicht für eine LHA-Datei erscheinen, wenn es ein
 "file/def_lha.info" gibt.

\textdegree{} AVDefIcons startet nicht

Sie dürfen AVDeficons und die aview.library nicht außerhalb des
 Programmverzeichnisses lagern, da dann keine Prefs mehr gefunden
 wurden.

1.82 AVDeficons Template

AVDeficons Template

```
Deficons IconPath,Wait/S,NoFreeIcons=NFI/S,NoShowIcons=NSI/S
      NoDrawerReSize=NDR/S
```

IconPath Default: "Deficons/"

Dieses Argument erlaubt die Definition einer Piktogrammschublade. Das
 hier angegebene Verzeichnis wird von AVDeficons zur Piktogrammsuche
 benutzt.

Wait/S

Normalerweise kommt AVDeficons nach dem Installieren der Patches immer

sofort wieder ins CLI zurück. Durch die Angabe dieses Schalters wird auf einen "Break C" gewartet. Im DOS bedeutet das, daß Sie AVDeficons durch Betätigen von "CTRL c" beenden können.

AVDefIcons beendet sich nicht, wenn andere Programme die selben System-Funktionen gepatcht haben, da hier kein sicherer Ausstieg möglich ist.

NoFreeIcons=NFI/S

Normalerweise werden die Positionen von "def_#?" Piktogrammen entfernt, damit die Icons vom System nicht übereinander gelegt werden. Durch die Angabe von "NoFreeIcons" oder "NFI" schalten Sie diese Funktion ab.

NoShowIcons=NSI/S

Dieser Schalter wirkt sich nur auf die Verzeichnis-Piktogramme aus. AVDeficons schaltet im normalen Modus die Piktogramme so um, das immer "Inhalt Anzeigen : Alle Dateien" und "Inhalt auflisten: Als Piktogramme" aktiviert wird. Dieser Schalter deaktiviert dieses Feature.

NoDrawerResize=NDR

Normalerweise werden die Größen der Verzeichnisfenster automatisch verändert und auf einen Defaultwert gestellt. (Breite:512, Höhe:144 und die Screenposition bei (64,48)) Dieser Schalter deaktiviert diesen Patch.

1.83 Tips,Tricks,Fragen und Antworten

Tips,Tricks,Fragen und Antworten

Hier finden Sie die Lösungen zu Problemen, die öfter auftraten oder zu Verwirrungen führten, aber auch Tips, Ideen, Vorschläge und Anwendungsmöglichkeiten.

AView startet nicht.

Wenn ich auf HELP drücke, dann erscheint keine Online-Hilfe.

Die Onlinehilfe erscheint nur in englisch.

AVViewPrefs ist manchmal sehr langsam!

1.84 Warum startet AView nicht?

\textdegree{} Warum startet AView nicht ?

Wenn Sie bei der Installation die OS2.1+ oder OS3.0+ Version installiert haben, muß sich die Locale.library im Libs: Verzeichnis befinden. Außerdem werden noch die Texte benötigt, die AView in seinen Fenstern anzeigen soll. Diese befinden sich im Programmverzeichnis von

AView im Verzeichnis "Catalogs".

Sie dürfen keins der Programme in eine andere Schublade verschieben. AView und AViewPrefs benötigen einige Dateien aus dem Programmverzeichnis, das sie im Falle des Verschiebens nicht mehr finden.

1.85 Die Onlinehilfe funktioniert nicht!

\textdegree{} Wenn ich auf HELP drücke, dann erscheint keine Online-Hilfe. Was mache ich falsch ? ←

AView hält sich streng an die "locale.library" und an deren Einstellungen. Wenn bei der Installation nur deutsch installiert haben, muß die "locale.library" mittels des Voreinstellungsprogramms Locale auch so eingestellt sein.

Außerdem muß die "Amigaguide.library" vorhanden sein.

1.86 Die Onlinehilfe ist nur in englisch

\textdegree{} Wieso bekomme ich nur die englische Anleitung, wenn ich die Online-Hilfe benutze ? ←

Wenn Sie OS2.0 benutzen, dann sollten Sie den Guide aus dem "docs/deutsch"-Verzeichnis nach "docs/english" kopieren, wenn sie eine deutsche Anleitung wünschen:

```
CD sys:tools/AView/Docs
copy deutsch/aview.guide english/
```

Ansonsten fehlt die deutsche Anleitung. Sie haben diese bei der Installation abgeschaltet. Installieren Sie AView einfach nochmal mit dem Installer drüber. Normalerweise werden defaultmäßig alle Anleitungen installiert.

1.87 AViewPrefs ist manchmal sehr langsam!

AViewPrefs ist manchmal sehr langsam!

Dies ist völlig normal! Beim Start sortiert AViewPrefs die Daten und die von ihnen erstellten Voreinstellungen nach dem Namen. Das dauert besonders bei den vielen vordefinierten Dateitypen ein paar Sekunden.

Beim Speichern sortiert AView die Daten wieder nach den Unterklassen, speichert die Daten und sortiert dann wieder nach der Vorgabe im sortieren nach Gadget. Das ist nötig, damit AView die Daten nach den

Unterklassen sortiert vorfindet und nicht selber sortieren muß! Es würde wohl mehr stören, wenn bei jedem start von AView die Sortierung einsetzen würde.

Sie können im Expertenmodus ja mal mittels des sortieren nach Gadgets aktivieren. Wenn Sie vorher mal mit dem Alle Gadget alle Daten hinzugefügt haben, dann werden Sie sehen, das dort viel Rechenzeit benötigt wird.
