

PIEd

Thomas Igracki

COLLABORATORS

	<i>TITLE :</i> PIEd		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Thomas Igracki	January 18, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	PIEd	1
1.1	ProductInfo-Editor Anleitung	1
1.2	PIEd/Inhalt des Archivs	1
1.3	PIEd/Was macht das Programm?	2
1.4	PIEd/Aufruf des Programms	2
1.5	PIEd/Beschreibung der Menüs	3
1.6	TextField-Gadget	4
1.7	PIEd/Programmentwicklung	6
1.8	PIEd/Wie kann man mich erreichen?	7
1.9	Das Magic-User-Interface	8
1.10	Product-Info fields	8
1.11	Product-Info field: .address	10
1.12	Product-Info field: .aminet-dir	10
1.13	Product-Info field: .author	15
1.14	Product-Info field: .construction	16
1.15	Product-Info field: .comment	16
1.16	Product-Info field: .contents	16
1.17	Product-Info field: .date	17
1.18	Product-Info field: .described-by	17
1.19	Product-Info field: .description	17
1.20	Product-Info field: .distribution	18
1.21	Product-Info field: .docs	19
1.22	Product-Info field: .email	19
1.23	Product-Info field: .exectype	20
1.24	Product-Info field: .fax	20
1.25	Product-Info field: .fullname	21
1.26	Product-Info field: .installsize	21
1.27	Product-Info field: .keywords	21
1.28	Product-Info field: .locale	21
1.29	Product-Info field: .name	22

1.30 Product-Info field: .phone	22
1.31 Product-Info field: .price	22
1.32 Product-Info field: .reference	22
1.33 Product-Info field: .requirements	23
1.34 Product-Info field: .restrictions	23
1.35 Product-Info field: .run	24
1.36 Product-Info field: .short	24
1.37 Product-Info field: .source	25
1.38 Product-Info field: .stored-in	25
1.39 Product-Info field: .submittal	25
1.40 Product-Info field: .tested	26
1.41 Product-Info field: .type	26
1.42 Product-Info field: .version	27
1.43 Product-Info field: .beschreibung	27
1.44 Product-Info field: .release	27
1.45 Product-Info field: .uploader	28

Chapter 1

PIEd

1.1 ProductInfo-Editor Anleitung

```
ProductInfoEditor / von Thomas Igracki  
-----/  
AFD - C O P Y R I G H T
```

```
Inhalt des Archivs  
Programmentwicklung  
Was macht das Programm?  
Wie kann man mich erreichen?  
Aufruf des Programms  
Beschreibung der Menüs  
Product-Info Felder  
The Magic-User-Interface  
Das TextField-Gadget
```

```
-----  
Diese Software unterliegt der "Standard Amiga FD-Software Copyright Note"  
Sie ist GIFTWARE wie definiert in Absatz 4g. [...]  
Für mehr Informationen lies bitte "AFD-COPYRIGHT" (Version 1 oder höher).
```

1.2 PIEd/Inhalt des Archivs

Folgende Dateien sollten sich in diesem Archiv befinden:

```
help/deutsch/PIEd.guide      --> deutsche Dokumentation, AmigaGuide Format
```

```

help/deutsch/AFD-COPYRIGHT.de --> deutscher COPYRIGHT-Text
help/english/PIEd.guide       --> englische Dokumentation, AmigaGuide Format
help/english/AFD-COPYRIGHT   --> englischer COPYRIGHT-Text
Product-Info                 --> Product-Info Datei für PIEd
PIEd.liesmich                --> kurze, deutsche Doku, ASCII
PIEd.readme                  --> kurze, englische Doku, ASCII
PIEd.iff                     --> IFF-Bild von PIEd
PIEd                         --> das Hauptprogramm

```

1.3 PIEd/Was macht das Programm?

Mit diesem Programm können Sie sog. 'Product-Info' Dateien '↔ bequem', d.h. per Maus bedienbar (aber natürlich auch per Tastatur), erstellen und natürlich auch nur ansehen.

Es wird dazu das

```

MagicUserInterface
von Stefan Stuntz und das

```

```

TextFieldGadget
von Mark Thomas verwendet.

```

Klicken sie bitte hier rauf um einen ersten Eindruck von PIEd zu bekommen.

Oben links befindet eine Liste mit allen möglichen Feldern.

Wenn ein Feld benutzt wird, erscheint es in einer anderen Farbe als die unbenutzten Felder, so daß man auf einen Blick erkennen kann, welche Felder schon definiert wurden und welche noch definiert werden müssen!

Rechts daneben sind ein paar StringGadgets mit den wichtigsten/gebräuchlichsten Felder. Diese sollte man mindestens ausfüllen! (Entweder direkt in den StringGadgets eingeben, oder per Listview, doch dazu später mehr. Man muß aber nicht alle Felder gebrauchen;-)

Unten befindet sich schließlich das

```

TextFieldGadget
, in dem der

```

Inhalt des aktuellen Feldes angezeigt und verändert werden kann!

Darunter befinden sich noch einpaar Buttons mit folgendem Sinn:

- Aktivieren: Aktivieren des Editors
- Setzen : Den Inhalt des Editors ins Feld schreiben
- Holen : Den Inhalt des Feldes ins Editor schreiben
- Löschen : Den Inhalt des Feldes (und im Editor) löschen

1.4 PIEd/Aufruf des Programms

Start von einer Shell:

```
[Run] PIEd [<ProductInfoFile>] [FOTL=FieldsOnTheLeft]
```

Der Parameter [<ProductInfoFile>] ist optional!.
Er gibt die PI-Datei an, die geladen werden soll.

Der FOTL=FieldsOnTheLeft Parameter gibt an, ob die Liste mit den Feldern größer dargestellt wird, sie werden dann ganz links von oben bis unten dargestellt.

1.5 PIEd/Beschreibung der Menüs

Projekt

Neu

Eine neue Datei erstellen.
Falls gerade eine verändert wird die noch nicht gespeichert wurde,
kommt ein Requester.

Öffnen...

Öffnen einer 'Product-Info' Datei.
Es erscheint ein FileRequester in dem man sich eine
'Product-info' Datei aussuchen kann.
Sie muß foldendem AmigaDOS-Muster entsprechen:

```
(.Product-Info|.Produkt-Info|Product-Info|Produkt-Info|#?.pi)
```

Speichern

Speichern der Datei.

Speichern als...

Speichern mit vorheriger Frage wohin.

Speichern als '.readme'

Speichern der Datei als .readme, dafür werden die Felder

short

,

aminet-dir

oder

type

,

author

,

version

und

description

.

Speichern als '.liesmich'

Speichern der Datei als .liesmich, dafür wird das Feld
beschreibung
benötigt.

Über...

Anzeigen einiger Informationen über PIEd.

Über MUI...

Anzeigen einiger Informationen über MUI.

Verbergen

Verbergen des Fensters. Es kann mittels ExChange oder per PopupKey
(einstellbar per MUI) wieder hervorgeholt werden.

Beenden...

Beenden von PIEd, wenn man die aktuelle Datei nicht gespeichert hat, erscheint ein Requester.

Bearbeiten

Ausschneiden

Den markierten Bereich ins Clipboard ausschneiden.

Kopieren

Kopieren des markierten Bereichs ins Clipboard.

Alles kopieren

Alles wird ins Clipboard kopiert.

Einfügen

Einfügen des Inhalts vom Clipboard in die aktuelle Position.

Löschen

Löschen des markierten Bereichs

Undo

Zurücknehmen der letzten Änderung (NUR Löschen und Einfügen!)

1.6 TextField-Gadget

PIEd benutzt zum anzeigen und verändern das TextField-Gadget von Mark Thomas in der v3.1. Es liegt im Aminet (dev/gui).

Hier ist mal ein Auschnitt aus der Doku, welche Tasten es unterstützt:

DOCS FOR USERS

You can mark text for cutting, copying, and erasing by simply clicking and dragging. Hitting alphanumeric keys replaces the text that is highlighted. Hitting cursor keys moves you to the front or end of the highlighted text.

If your cursor is already somewhere in the textfield, you can hold the SHIFT key and click to mark the text from the current cursor position to the place where you clicked.

And the last way to mark text is to double-click, which will mark the word you clicked on. If you didn't click on a word, but rather you clicked on spaces, the whole block of spaces is marked. And if you clicked on word delimiters, the whole block of delimiters is marked.

While you drag to scroll, the farther away from the gadget your mouse pointer is, the faster the gadget will scroll.

For key sequences, the Amiga Style Guide was followed. Anywhere the undo buffer is mentioned, the statement is only valid if the UndoStream is supplied (see tag section below).

Key Sequence	Function
TAB	Activate next gadget (if GA_TabCycle)
SHIFT TAB	Activate previous gadget (if GA_TabCycle)

SHIFT CURSOR UP	Move to the top line in the current page, or scroll up one page if cursor is on top line
SHIFT CURSOR DOWN	Move to the bottom line in the current page, or scroll down one page if cursor is on top line
CTRL or SHIFT CURSOR RIGHT	Move to the right end of the current line
CTRL or SHIFT CURSOR LEFT	Move to the left end of the current line
SHIFT BACKSPACE	Delete all text to the left of cursor on the current line
SHIFT DELETE	Delete all text to the right of the cursor on the current line (in block cursor mode this also includes the highlighted character)
CTRL CURSOR UP	Move to the top line of the text
CTRL CURSOR DOWN	Move to the bottom line of the text
ALT CURSOR RIGHT	Move to the next word (using the delimiter characters provided by the programmer)
ALT CURSOR LEFT	Move to the previous word (using the delimiter characters provided by the programmer)
ALT CURSOR UP	Move to first character in gadget
ALT CURSOR DOWN	Move to last character in gadget
ALT BACKSPACE	Deletes the word to the left of the cursor starting at the current cursor position
ALT DEL	Deletes the word to the right of the cursor starting at the current cursor position
CTRL X	Deletes the whole line that the cursor is on
RAMIGA [Switch to left justification (if TEXTFIELD_UserAlign is set)
RAMIGA \ or RAMIGA =	Switch to center justification (if TEXTFIELD_UserAlign is set)
RAMIGA]	Switch to right justification (if TEXTFIELD_UserAlign is set)
RAMIGA E	Erase all text in gadget (saved in undo buffer) (no read-only)

RAMIGA V	Paste text from clipboard to current cursor position (no read-only)
RAMIGA A	Mark all text
RAMIGA U	Undeletes (pastes) the last block of text marked, or recover from RAMIGA E (no read-only)

When text is highlighted the following keys have functions:

BACKSPACE	Erase marked text (saved in undo buffer)
DEL	Erase marked text (saved in undo buffer)
RAMIGA X	Cut marked text to clipboard (no read-only)
RAMIGA C	Copy marked text to clipboard
RAMIGA V	Replace marked text with text from clipboard (save marked text in undo buffer) (no read-only)
(any text key)	Replace marked text with that character

1.7 PIEd/Programmentwicklung

Release 1.0:

- 37.15 (28.05.96): New: If a field wasn't set and you select another one then a requesters appears asking if you would like to save. ↵
- Chg: Now all last '\n' will be removed not just the last.
New: Added field .uploader for writing .readmes.
- 37.14 (27.05.96): New: HelpNodes for the String-Gadgets
New: The PI-Fields-Listview can be made larger if you use FOTL=FieldOnTheLeft/S on startup.
New: Menu/Settings/MUI added
: New: Menu "Project/SaveAs '.liesmich'
- 37.13 (01.05.96): New: 7 new fields (used in pitoool) added: .aminet-dir, .comment, .contents, .fax, .keywords, .locale, .phone, ↵
Chg: The .aminet-dir field is used for the "type:" line .readme's ↵
if available, else .type is used like before.
New: After succesfully saving a .readme a requester comes saying ↵
that it is saved.
New: The last \n of a field content will be removed on setting. ↵
- 37.12 (25.04.96): New: The user has control over left/center/right ↵
justification
of text through RIGHT-AMIGA [,=,] keyboard shortcuts.
New: SPACE for activating the textfield.

- New: Some more references to guide file.
- New: rightAmiga RETURN for activating the textfield.
- New: If you have made changes and want to quit, you will be asked if you really want to quit, else quit immediatly.
- 37.11 (29.03.96): Fix: Changed the hotkeys of the editor buttons so they can be used out of the editor gadget (with ramiga).
- 37.10 (01.03.96): New: WindowTitle displays the name of the file
- New: Clipboard support for the editor.
- New: Pass amiga-commands within the editor.
- 37.9 (23.11.95): Fix: Fix für MUI3, der Scrollbar fehlte bei den Listviews!
- 37.8 (31.10.95): Fix: fixed a small bug
- New: Added field ".beschreibung" and ".release"
- 37.7 (04.09.95): New: Menu "Project/SaveAs '.Readme'"
- 37.6 (28.08.95): Bug: Das letzte Feld wurde nicht eingelesen.
- 37.5 (08.08.95): Data typed in string gadget will be recognized and the FieldList will be updated.

- 37.4 (11.09.94): ·Änd: Der 1. Parameter ist nicht mehr nötig, falls er weggelassen wird, kommt ein FileRequester.
- 37.3 (27.08.94): ·Neu: Project/Hide zum verbergen des Programms
- Neu: ARexxPort-Name wird im About-Requester angezeigt
- 37.2 (23.08.94): ·Bug: Falls die Datei ein Verzeichnis war oder nicht existierte. ←
- Neu: Menu "Edit/Copy" zum kopieren des Inhaltes des aktuellen Feldes ins Clipboard.
- Neu: Jetzt werden nur noch die Felder angezeigt, die vorhanden sind! ←
- 37.1 (13.08.94): ·So, nun läuftts erstmal!
- Shell-Support eingebaut
- 37.0 (12.08.94): Angefangen, da ich heute die FrozenFish CD bekam und ein Programm brauchte, womit ich mir die Product-Info Files komfortabel ansehen kann!
- Erstmal die Oberfläche programmiert.

1.8 PIEd/Wie kann man mich erreichen?

Falls ihr Verbesserungsvorschläge, irgendwelche Fragen, Bugreports usw. habt, dann könnt ihr mir schreiben, am besten per e-mail!

Z-Netz: T.Igracki@BAMP.berlinet.de
 FIDO : Thomas_Igracki%2:2410/309.34
 UseNet: lokai@cs.tu-berlin.de -oder-
 t.igracki@bamp.berlinet.de (bevorzugt!)
 Post : Thomas Igracki, Siegenger Str. 34, Germany-13583 Berlin

Falls ihr mir eine kleine (oder auch große!) Geldspende für meine Arbeit an PIEd zukommen lassen wollt, dann schickt es mir in Bar oder überweist es mit einem entsprechenden Vermerk (und EMail) auf folgendes Konto:

Empfänger: Igracki, Thomas
 KontoNr. : 860169138 BLZ: 100 500 00
 bei : Sparkasse Berlin

Vermerk : Spende für PIEd, <Dein Name + EMail>

Wenn ihr mir dann noch eine eMail schickt, werde ich euch informieren, wenn eine Version von PIEd fertig ist! Und, wenn gewünscht automatisch zumailen!

Schreibt mir aber wenigstens eine eMail, wenn ihr PIEd benutzt!

Viel Spaß mit dem Programm!

1.9 Das Magic-User-Interface

This application uses

MUI - MagicUserInterface

(c) Copyright 1993/94 by Stefan Stuntz

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

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

If you want to register directly, feel free to send

DM 30.- or US\$ 20.-

to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

1.10 Product-Info fields

The following are the fields defined by the Product-Info ↔
Specification

v6 as designed by Fred Fish and Udo Schuermann (with some extra fields used by pitoool!).

address

aminet-dir
author
construction
comment
contents
date
described-by
description
distribution
docs
email
exectype
fax
fullname
installsize
keywords
locale
name
phone
price
reference
requirements
restrictions
run
short
source
stored-in
submittal
tested

type
 version
 Now three I have added:

beschreibung
 release
 uploader

1.11 Product-Info field: .address

```
.address:
<<<OPTIONAL>>>
PURPOSE:    Describe a full postal address of the author, to
             be used if it becomes necessary or desirable to
             contact the author. Do not specify the author's
             name, as this is already in the
author
             field.
FORMAT:     Multiple lines; formatting symbols n are not
             required, as physical line breaks are equivalent.
NOTES:      SEE THE
author
             FIELD FOR IMPORTANT INFORMATION
```

1.12 Product-Info field: .aminet-dir

```
.aminet-dir:
<<<OPTIONAL>>>
PURPOSE:    Sets the directory where to put it on aminet.
FORMAT:     1 line, see below for a list of possible strings.
NOTES:
```

All directories on Aminet

new	Upload area
priv	Private uploads
recent	Files uploaded the last seven days
biz	Business software
biz/dbase	Data base software
biz/dopus	Directory Opus files
biz/demo	Demo versions of commercial software
biz/misc	Misc business type soft
biz/patch	Commercial software patches

comm	Communications
comm/ambos	AmBoS (BBS) related
comm/amiex	Ami-Express BBS related
comm/bbs	BBS programs
comm/cnet	CNET BBS related
comm/dlg	DLG BBS related
comm/envoy	Envoy network software
comm/fido	FIDO software
comm/mail	Mailing soft
comm/maxs	MaxsBBS related
comm/mebbs	MEBBSNet BBS related
comm/misc	Misc communications
comm/net	Networking
comm/news	Newsreaders
comm/tcp	AmiTCP networking
comm/term	Terminal programs
comm/ums	Universal Message System
comm/uucp	UUCP software
comm/xeno	Xenolink BBS related
demo	Gfx and sound demos
demo/40k	40K intros
demo/aga	Demos requiring AGA chipset
demo/disk	Demos on diskettes
demo/ecs	Demos for the ECS chip set
demo/euro	Euro demos in file form
demo/file	Single file demos
demo/funet	From the Funet collection
demo/intro	Intros
demo/mag	Demo coder magazines
demo/mega	Euro demos requiring floppies
demo/slide	Image oriented demos
demo/sound	Sound oriented demos
demo/ta95	The Assembly 1995 demos
demo/tg93	The Gathering 1993 demos
demo/tg95	The Gathering 1995 demos
demo/tp92	Demos from The Party 92
demo/tp94	Demos from The Party 94
demo/track	Trackmos
dev	Development software
dev/amos	AMOS BASIC compiler
dev/asm	Assembly language related
dev/basic	BASIC language related
dev/c	C language related
dev/cross	Cross development
dev/debug	Debugging software
dev/e	E language related
dev/gcc	GNU C compiler
dev/gui	GUI builders
dev/lang	Various languages
dev/m2	Modula-2 related
dev/misc	Misc development software
dev/moni	Monitors/disassemblers
dev/obero	Oberon related software
dev/src	Tutorial sources

disk	Disk/Harddisk tools
disk/bakup	Backup software
disk/cache	Disk caching
disk/cdrom	CD-ROM related software
disk/misc	Various disk related tools
disk/moni	Disk monitors
disk/optim	Disk optimizers
disk/salv	Disk recovery software
docs	Text documents
docs/anno	Announcement documents
docs/etext	Project Gutenberg books
docs/help	Help on various issues
docs/hyper	Hypertext documents
docs/lists	Lists and tables
docs/mags	Magazines
docs/misc	Various documents
docs/rview	Amiga product reviews
game	Games
game/2play	2 and more player games
game/actio	Action games
game/board	Board games
game/demo	Demos of commercial games
game/gag	Gag programs
game/hint	Game hint documents
game/jump	Jump-n-run games
game/misc	Miscellaneous games
game/patch	Patches for games
game/role	Role, adventure games
game/shoot	Shoot-em-up games
game/think	Mind games
game/wb	Workbench games
gfx	Graphics
gfx/3d	3D graphics software
gfx/3dobj	3D objects
gfx/aga	AGA chip set software
gfx/board	Custom gfx board software
gfx/conv	Gfx conversion tools
gfx/edit	Paint programs
gfx/fract	Fractals and Mandelbrots
gfx/misc	Misc graphics software
gfx/pbm	Pbm+ gfx conversion
gfx/show	Image/anim displayers
gfx/x11	Amiga X-Windows
hard	Hardware
hard/drivr	Drivers for hardware
hard/hack	Selfmade hardware
hard/misc	Misc hardware related files
info	General FTP information
info/adt	Machine readable filelists
info/index	Various Aminet index files
info/sites	Info on other FTP sites

info/start	Beginner information
info/stats	Statistics about this site
misc	Miscellaneous
misc/amag	German Amiga Mag PD
misc/antiq	Nostalgic stuff
misc/edu	Educational software
misc/emu	Emulations
misc/math	Mathematics
misc/misc	Uncategorizable
misc/sci	Scientific software
misc/unix	Amiga related UNIX software
mods	Music modules
mods/8voic	8-voiced songs
mods/airon	Mods by Airon
mods/boing	Boing collection
mods/chart	Module hits
mods/chip	Mods without samples
mods/ctp	MODs by Counterpoint
mods/demo	Demo style music
mods/ephnx	Modules from Euphonix
mods/fby	Mods by Fabio Barzagli
mods/funet	Funet collection
mods/funk	Funky/jazzy mods
mods/hardc	Hardcore techno mods
mods/hje	Mods by H.J.Ehrentraud
mods/hw	Mods by Hillware
mods/instr	Instrumental mods
mods/jogei	Mods by Jogeir Liljedahl
mods/jungl	Jungle/Breakbeat mods
mods/maxym	Mods by Maxym
mods/med	MED modules
mods/misc	Miscellaneous songs
mods/mpeg	Samples in MPEG format
mods/otis	Modules by Otis
mods/piano	Piano mods
mods/pop	Pop music
mods/pro	Protracker songs
mods/przk	Modules by Parzek
mods/rated	Misc rated modules
mods/s3m	Multichannel S3M mods
mods/smpl	Sound samples
mods/sets	Module collections
mods/spark	Modules by P.D. Spark
mods/sidew	Modules by Sidewinder
mods/slc	Modules by Slice
mods/slow	Slow/melodic mods
mods/synth	Synthesizer modules
mods/rock	Rock'n'Roll
mods/techn	Techno music
mods/tranc	Trance techno mods
mods/voice	Music with voices
mods/xm	Multichannel XM mods
mus	Musical software
mus/edit	Composing software

mus/midi	MIDI software
mus/misc	Miscellaneous
mus/play	Sound players
mus/smpl	Sound samples
pix	Pictures
pix/anim	Animations
pix/astro	Astronomy pictures
pix/art	Hand drawn imagery
pix/bill	Bill Graham traces
pix/clip	Clip art
pix/eric	Eric Schwartz anims
pix/fauna	Animals
pix/fract	Fractals
pix/guard	Guardians traces
pix/icon	Icons and backdrops
pix/illu	Illustrations
pix/imagi	Imagine traces
pix/irc	The people on IRC
pix/misc	Miscellaneous images
pix/mwb	MagicWB icons
pix/real3	Real3D traces
pix/sport	Sports pictures
pix/trace	Raytraced pictures
pix/vehic	Vehicle images
pix/views	Scenery, countries
pix/wb	Example WB screens
text	Text related
text/bfont	Bitmap fonts
text/dtp	Desk top publishing
text/edit	Editors
text/font	Fonts of all formats
text/hyper	Hypertext tools
text/ifont	Compugraphic IntelliFonts
text/misc	Misc text software
text/pfont	Adobe Postscript fonts
text/print	Printing tools
text/show	Text displayers
text/tex	TeX formatting package
util	Utilities
util/app	Appicons, appwindows
util/arc	Archivers
util/batch	Script software
util/boot	Startup sequence software
util/blank	Screen blankers
util/cdity	Commodity type software
util/cli	Command line interface
util/conv	File conversion
util/crypt	Encryption tools
util/dtype	Datatypes for OS 3.x
util/dir	Directory utilities
util/gnu	GNU utilities
util/libs	Libraries
util/misc	Miscellaneous
util/moni	System monitors

```

util/mouse  Mouse accelerators
util/pack   Packers
util/rexx   AREXX tools
util/shell  Shells
util/sys    Commo file replacements
util/time   Clocks and timing
util/virus  Virus checkers
util/wb     Workbench utilities

```

1.13 Product-Info field: .author

```

      .author:
PURPOSE:      Any and all authors who have a part in the program
FORMAT:       Any number of lines, treated as one line (n in the
              text will "break up" the line into multiple visual
              lines.)
EXAMPLE:      Joe R. User, Tea Rexx.
EXAMPLE:      J. Jonesn
              Random Hackern
              B. Clinton
NOTES:        Addresses should be placed in the
              address
              field.
              There should be only one
              address
              field for each

              author
              field.
              If more than 1
              author
              field is specified, then the
              same number of
              address
              and
              email
              fields must also
              be given in a 1-to-1 relationship (i.e. the 3rd

              author
              field must be associated with the 3rd
              address, and the 3rd
              email
              field.)
              EX: see the example "Joe R. User, Tea Rexx" above;
              Assume that Joe R. User has long vanished and no
              known address, but that Tea Rexx has supported the
              program for a while. If an
              address
              and/or
              email
              field is available for Tea Rexx, then you must
              specify EMPTY
              address
              and/or

```

```

email
  fields for the
    author listed BEFORE the ones for Tea Rexx.
    Likewise, if the two authors names were reversed,
    you would NOT have to specify blank
address
  and/or

email
  fields for the second author.  I hope that
    makes sense.

```

1.14 Product-Info field: .construction

```

      .construction:
<<<OPTIONAL>>>
PURPOSE:      Describe the type of language(s) used to create
                this program and the methods used to build the
                final executable.  If possible, include the
                compiler version(s) and possibly important
                options, such as optimization.
FORMAT:       Free form; see
                description
                for more information.
EXAMPLE:      SAS/C++ 6.5 with full optimization.
EXAMPLE:      AdaEd.
EXAMPLE:      Fortran with self-made compiler.
EXAMPLE:      AMOS
NOTES:        This is usually closely related to the
                exectype
                field but differs from it in that the
                exectype
                might be "Compiled C" but the compiler used ←
                was
                "RottenC 0.97"

```

1.15 Product-Info field: .comment

```

.comment:
  <<<OPTIONAL>>>
PURPOSE:

FORMAT:

NOTES:

```

1.16 Product-Info field: .contents

```
.contents:
  <<<OPTIONAL>>>
  PURPOSE:

  FORMAT:

  NOTES:
```

1.17 Product-Info field: .date

```
.date:
  PURPOSE:      The program's official release date; not the date
                 it made it into the database.
  FORMAT:       year.month.day
                 1 line only
  EXAMPLE:      1993.09.27
  NOTES:        The date format is chosen to be easily sortable.
                 Note the use of leading zeros in month and day.
                 The full year is to be given in anticipation of
                 the coming change to a new millenium.
```

1.18 Product-Info field: .described-by

```
.described-by:
  <<<OPTIONAL>>>
  PURPOSE:      Specifies who created the description (Product-Info
                 file) for the program.
  FORMAT:       Free form; should include an electronic mail
                 address, too, if available.
  EXAMPLE:      Fred Fish (fnf@fishpond.cygnus.com)
  EXAMPLE:      Udo Schuermann <walrus@wam.umd.edu>
```

1.19 Product-Info field: .description

```
.description:
  PURPOSE:      A full-text description of your program, containing
                 anything that is NOT ALREADY available through the
                 other fields (see above and below.) The reader
                 should gain a good understanding what your program
                 can and cannot do. If you mention other programs
                 please do not forget to provide a
reference
  field
                 for each such mention.
  FORMAT:       Any number of lines, treated as one line.
                 Formatting is permitted, but generally discouraged.
  NOTES:        Do not indent your text if you choose to format
                 your text into multiple paragraphs. Do not use t
                 as a tab. Leave paragraph formatting to KingFisher.
```

1.20 Product-Info field: .distribution

.distribution:
 <<<OPTIONAL>>>
PURPOSE: Describes the distribution and ownership status
 of this software. Please see below for a list of
 common (and recommended!) terms to use.
FORMAT: 1 line
EXAMPLE: Shareware
NOTES: Please see the table below for descriptions of the
 recommended terms.

LIST OF SOFTWARE STATUS KEYWORDS

Commercial Commercial software is owned and distributed through licenses. It costs money to individual end-users and is not freely distributable. SUCH PIECES SHOULD NOT APPEAR ON DISKS THAT ARE MEANT FOR FREELY DISTRIBUTABLE SOFTWARE!

Commercial Demo Represents a demonstration of a commercial package. As such, commercial demos are freely distributable and may have limitations as described in the .limitations field.

Giftware Like shareware, usually.

Shareware Such software is owned and copyrights are held by the author(s). The software may be distributed freely, but not sold for profit, of course. Shareware often specifies a limit of some time after which you are requested or required to register the software (i.e. pay for it.) This provides you with the means to evaluate the software thoroughly before paying for it.

Freeware Such software is owned and copyrights are held by the author(s). The software may be distributed freely, but not sold for profit, which would mean the software is no longer FREEware. No payments are required for such software.

Public Domain Software labeled PD (Public Domain) belongs to the public, i.e. ANYONE. Some people release their software into the public domain with the mistaken idea that they can continue to own and control the program. Not so. Software that is labeled Public Domain (or said by the author to be released into the public domain) truly belongs to anyone and everyone. It is quite legal for someone to take such a program and sell it for profit as is. Likewise, it is perfectly acceptable to modify public domain

software to build a better product (or whatever) out of it and then sell it for profit.

GNU Public License

The terms and conditions of this license are long and not easily reproduced here. Suffice to say that software released under the GNU Public License must be distributed with source code. They are not public domain, however.

GNU Library Public License

The terms and conditions of this license are long and not easily reproduced here. Suffice to say that software released under the GNU Library Public License must be distributed with source code. They are not public domain, however.

Copyright but Freely Redistributable

The author holds all copyrights but allows the material to be freely distributed under specified conditions.

1.21 Product-Info field: .docs

.docs:

```
<<<OPTIONAL>>>
PURPOSE:   List all documentation files, possibly for viewing
           from within KingFisher for more detailed info.
FORMAT:    1 line per file
EXAMPLE:   HomeBase.guide
           HomeBase.dvi
           HomeBase.doc
NOTES:     KingFisher examines the EXTENSION and invokes the
           appropriate viewing tool: MultiView/AmigaGuide for
           .guide files, ShowDVI for .dvi files, more for
           anything else. These files can also be sent to the
           printer via KingFisher (i.e. print .ps or .doc
           files.) KingFisher will honor the PAGER
           environment variable (defaults to 'more') to
           display standard text.
NOTES:     Omit any path to these files, unless it is a
           relative path from within the program's CD-ROM or
           disk directory. Do not specify these files if
           they are located within archive files; remember:
           the files must exist as they are given here!
```

1.22 Product-Info field: .email

```
.email:
<<<OPTIONAL>>>
PURPOSE:   Describe a full electronic mail address. Make
           sure that this address is complete and reachable
```

even from less well-connected sites. The author of KingFisher, for example, can be reached as walrus@wam.umd.edu

It would be an error to specify only "walrus" or "walrus@wam" even though these will work within the particular organization where this address is valid.

FORMAT: Multiple lines; formatting symbols n are not required, as physical line breaks are equivalent. Do not specify more than one email address per line. The more you abide by RFC-822 specifications the better.

EXAMPLES: walrus@wam.umd.edu (Udo Schuermann)
 Udo Schuermann <walrus@wam.umd.edu>
 "Udo Schuermann" <walrus@wam.umd.edu>
 <walrus@wam.umd.edu> Udo Schuermann

NOTES: You may specify multiple electronic mail addresses in order of decreasing reliability and permanence, each on its own line.

SEE THE

author

FIELD FOR IMPORTANT INFORMATION

1.23 Product-Info field: .exectype

.exectype:

<<<OPTIONAL>>>

PURPOSE: Describe the type of executable(s) that make up your program. Examples: 68xxx, AMOS, Script, ARexx, Compiled basic, Amigabasic, etc.

FORMAT: Free form; see description for more information.

EXAMPLE: AMOS

EXAMPLE: 68000, 68020, and 68040.

EXAMPLE: Compiled BASIC

EXAMPLE: Compiled ARexx

NOTES: AMOS-based software has been said to not work on some systems at all; this entry allows a user to determine if the software is worth obtaining in the first place.

1.24 Product-Info field: .fax

.fax:

<<<OPTIONAL>>>

PURPOSE:

FORMAT:

NOTES:

1.25 Product-Info field: .fullname

```
.fullname:
<<<OPTIONAL>>>
PURPOSE:      The program's full (or complete) name
FORMAT:       1 line only
EXAMPLE:      Amiga Intuition Based Benchmarks
EXAMPLE:      GNU C Compiler
NOTES:        If the
              name
              is not an abbreviation then omit the
              fullname. No sense in giving the name twice!
```

1.26 Product-Info field: .installsize

```
.installsize:
<<<OPTIONAL>>>
PURPOSE:      Indicate the minimum and maximum sizes of the
              executable as it is installed. The minimum size
              should give an indication of how much disk space
              is required for a minimal installation (perhaps
              lacking help files and miscellaneous tools) while
              the maximum size should indicate the absolutely
              highest amount of disk space required by the
              program.
FORMAT:       1 or more lines; Only the first line has a fixed
              format, the rest are free-form. See examples.
              Always indicate the number scales with a capital
              K (for kilobyte) or M (for megabyte)
EXAMPLE:      220K - 2M
              Most of the database files can be kept on floppy
              disks, so valuable harddisk space is not wasted.
EXAMPLE:      18K
EXAMPLE:      38K - 500K
              Lots of documentation and example scripts make up
              the bulk of the installation.
```

1.27 Product-Info field: .keywords

```
.keywords:
<<<OPTIONAL>>>
PURPOSE:
FORMAT:
NOTES:
```

1.28 Product-Info field: .locale

.locale:
 <<<OPTIONAL>>>
 PURPOSE:

 FORMAT:

 NOTES:

1.29 Product-Info field: .name

.name:
 PURPOSE: The program's name
 FORMAT: 1 line only
 EXAMPLE: KingFisher
 EXAMPLE: HomeBase VI
 EXAMPLE: AIBB
 EXAMPLE: gcc

1.30 Product-Info field: .phone

.phone:
 <<<OPTIONAL>>>
 PURPOSE:

 FORMAT:

 NOTES:

1.31 Product-Info field: .price

.price:
 <<<OPTIONAL>>>
 PURPOSE: Describes the cost of this program to the user.
 FORMAT: Any number of lines, treated as one line.
 EXAMPLE: \$50 (US), DM75.
 NOTES: In order to make this field more useful, it is
 STRONGLY recommended that the FIRST currency
 listed is United States Dollars as shown in the
 EXAMPLE above. This allows a search to be limited
 to a common price base. If you charge no money
 for this program, omit this field!

1.32 Product-Info field: .reference

```
.reference:
  <<<OPTIONAL>>>
  PURPOSE:    Full path to where this program's files are stored,
              as well as the version that is stored there.
  FORMAT:     2 lines per reference: the first line specifies
              the full path (with trailing slash) and the second
              line, the version.
  NOTES:     Multiple such fields may be provided to reference
              previous versions of this program, as well as
              other programs that might be of interest. The
              versions should be listed in reverse chronological
              order and SHOULD include the CURRENT entry.
              Please note that it is VERY VERY VERY important
              that you specify the CORRECT PATH! Without a
              correct path, this entry will be nearly useless!
              SPECIFY THE PATH WITH A NEW SUBMISSION ONLY IF YOU
              KNOW WHERE IT IS STORED; NEW SUBMISSIONS WILL HAVE
              A PATH ASSIGNED HERE AUTOMATICALLY. YOU SHOULD
              PROVIDE THE PROPER PATHS TO KNOWN AND EXISTING
              SOFTWARE.
  EXAMPLE:   FishROM-0002:Productivity/Databases/HomeBase VI/
              417.0
              FishROM-0001:Productivity/Databases/HomeBase VI/
              415.12
```

1.33 Product-Info field: .requirements

```
.requirements:
  PURPOSE:    List requirements for your program. These should
              give the reader enough information to determine if
              the software will run on his/her system or not.
              Be sure to specify operating system versions,
              (hard)disk space requirements, etc. If your
              program requires any external libraries that are
              not part of the system software, it would be nice
              to list them here and comment on whether or not
              they are included in the archive.
              If your program is known to run on every existing
              (Amiga) platform, state this in this field!
  FORMAT:     Free form; see
              description
              for more info.
  EXAMPLE:   68020, 68030, or 68040 CPU; 3M free RAM; 18M disk
              space; at least 640x480 display capabilities!
  EXAMPLE:   Requires WB2.1 (V38)
  EXAMPLE:   Requires 1024x768 (or larger) display capability.
  EXAMPLE:   Works only with 4096-channel, 230db BLAZETHUNDER
              Audio board.
  EXAMPLE:   Requires MUI (MagicUserInterface) version 5.
```

1.34 Product-Info field: .restrictions

.restrictions:

PURPOSE: List restrictions placed upon this program. These should indicate in which way this program has been made dysfunctional (for demo purposes), problems (bugs) known to exist with this program, or any other thing that lets the user know that this program, as seen in this distribution, may not fully satisfy the user in some form.

FORMAT: Free form; see description for more info.

EXAMPLE: Demo version has SAVE and PRINT options disabled.

EXAMPLE: The ReadOperatorsMind command fails to work with CDTV units. Incompatible with the Discus Ejector utility.

EXAMPLE: Crashes if iconified while loading a sample or image larger than 64K.

EXAMPLE: Requires a PAL display.

EXAMPLE: The program is in German but the documentation offers translations into English and Swahili on a menu-by-menu and gadget-by-gadget basis.

NOTES: Do NOT use this field for things like "won't work with KS 1.3" or "won't run with less than 2 Megs of RAM."

1.35 Product-Info field: .run

.run:

<<<OPTIONAL>>>

PURPOSE: Specifies how to start the program.

FORMAT: visible=type,command
Where 'type' is either WB or CLI to indicate the required startup environment.

EXAMPLE: HomeBase VI=WB,HomeBase VI
HomeBase VI=CLI,ExecuteMe.HB6
HomeBase VI Fixer=CLI,ExecuteMe.HB6Fixer

EXAMPLE: FishTub=WB,ExecuteMe

NOTES: KingFisher requires that this entry strictly follows the above format.
The user is shown all text up to the first equal sign (the 'visible' portion.) The 'type' portion must be terminated with a comma (,) and following it will be the command to be executed.
Selecting it will either invoke the program from the Workbench (invoking it as if double clicked on its icon (if the .info file exists), or execute the indicated shell command line as if it has been typed at an open console window.

1.36 Product-Info field: .short

```
.short:
  <<<OPTIONAL>>>
  PURPOSE:      A one-line description, preferably not exceeding
                 40 characters in length. This description is to
                 give a single-glance insight into the program's
                 purpose.
  FORMAT:       1 line only.
  EXAMPLE:      Software catalog/search/maintenance tool, multi-user.
```

1.37 Product-Info field: .source

```
.source:
  <<<OPTIONAL>>>
  PURPOSE:      Describe what source code is available with this
                 program. If source code is not available then
                 omit this field. The
                 construction
                 field often
                 helps further identify the type of source if you
                 omit details here. How large is the source?
  FORMAT:       Free form; see
                 description
                 for more information.
  EXAMPLE:      SAS/C,Manx,DICE source (750K) available for $15
  EXAMPLE:      Oberon source included. 85K
  EXAMPLE:      Limited C source (15K) included.
  EXAMPLE:      All source plus custom libraries, included: 12MB
```

1.38 Product-Info field: .stored-in

```
.stored-in:
  PURPOSE:      Specifies where and especially HOW the application
                 is stored. This field should specify EITHER the
                 name of a directory (ending with a : or a /) OR the
                 name of a file (one that does NOT end with : or /)
  FORMAT:       1 or more lines.
  EXAMPLE:      FF1000:Disks701-1000/Disks941-960/Disk950/Enforcer/
                 FF1000:BBS/Disks501-1000/Disks941-960/Disk950/Enforcer.lha
  NOTES:       It is up to the particular application to decide
                 how to handle this information. If the extension
                 on the file is .lha, .lzh, .Z, .zoo, .pak, .zip,
                 etc. then you could, for example, call upon the
                 archiver of choice to unpack the application into a
                 temporary directory and let the user run the
                 program or list the files, or whatever.
```

1.39 Product-Info field: .submittal

```
.submittal:
  <<<OPTIONAL>>>
  PURPOSE:      Identifies who submitted the program to Fred or
                  else how this program came to be on the reference
                  disk.
  FORMAT:       Free form; usually one line.
  EXAMPLE:      Submitted on disk directly by the author.
  EXAMPLE:      Downloaded from wuarchive.wustl.edu in pub/aminet/util/misc
```

1.40 Product-Info field: .tested

```
.tested:
  <<<OPTIONAL>>>
  PURPOSE:      Give an indication of which configurations have
                  served as test environments.
  FORMAT:       Free form; see
                  description
                  for more information.
  EXAMPLE:      A500(512K Chip, 0K Fast, 1 Floppy), A2000(1M Chip,
                  2M Fast, 40M HD, 1 Floppy); not tested on 68020+
                  CPUs.
  EXAMPLE:      A1000, A500, A600, A2000, A2000/30, A3000, A1200,
                  A4000/30, A4000/40 with various amounts of Chip
                  and Fast RAM, with and without MMU or FPU. Found
                  to be free of Enforcer hits and able to work with
                  virtual memory products; compatible with Retina,
                  EGS/Spectrum, and Picasso software. Also tested
                  under V33 through V40 system software.
```

1.41 Product-Info field: .type

```
.type:
  PURPOSE:      A keyword that describes the nature of the program
  FORMAT:       Preferrably a single word or two.
  EXAMPLE:      Database
  EXAMPLE:      Spreadsheet
  EXAMPLE:      Animation Player
  EXAMPLE:      Animation Tools
  EXAMPLE:      Communications
  EXAMPLE:      Display Commodity
  EXAMPLE:      Mouse Commodity
  NOTES:       Avoid abbreviations. Refer to the list below for
                  suggestions.
```

EXAMPLES OF "TYPE" WORDS

Action Game	Animation	Animation Player
Animation Tool	Archiver	CLI Tool
Communications	Compiler	Compression
Database	Disk Tool	Display Commodity
Drawing	Image Conversion	Image Processing

Library	Mouse Commodity	Music Composition
OS Utility	Painting	Picture
Printing	Sound Analysis	Sound Editing
Sound Playing	Spreadsheet	Strategy Game
Text	Text Editing	Text Viewer
Thinking Game	Word Processing	Workbench Tool

1.42 Product-Info field: .version

```
.version:
  PURPOSE:    The program's version number
  FORMAT:     MAJOR.MINOR
              1 line only
  EXAMPLE:    37.100
  NOTES:     Please note that the Commodore guidelines specify
              that the number after the period is NOT a FRACTION
              but rather a WHOLE NUMBER! Thus, the following is
              a valid progression:
              37.1 37.17 37.39 37.100 37.170
              The following are all vastly different versions:
              37.1 37.10 37.100 37.1000
  NOTES:     The format given for this field is really more of a
              SUGGESTION rather than a RULE. There is no reason
              why you can't store "Today's Version" or "v940205"
              instead of 18.173. In an ideal world everyone
              would use Commodore guidelines, but there are
              enough exceptions.
```

1.43 Product-Info field: .beschreibung

```
.beschreibung:
  PURPOSE:    Like
              .description
              but for german users.
  FORMAT:     Any number of lines, treated as one line.
              Formatting is permitted, but generally discouraged.
  NOTES:     Do not indent your text if you choose to format
              your text into multiple paragraphs. Do not use t
              as a tab. Leave paragraph formatting to KingFisher.
```

1.44 Product-Info field: .release

```
.release:
  PURPOSE:    The program's release number
  FORMAT:     MAJOR.MINOR
              1 line only
  EXAMPLE:    1.3
  NOTES:     This should be the "public number".
```

```
.version
```

should be the internal, if exists.

1.45 Product-Info field: .uploader

```
.uploader:
  <<<OPTIONAL>>>
  PURPOSE:      Specifies who will upload the archive to the aminet.
  FORMAT:       1 line with 1 email-address.
  EXAMPLE:      T.Igracki@Bamp.berlinet.de
  NOTES:        Main purpose is for .readme files.
```
