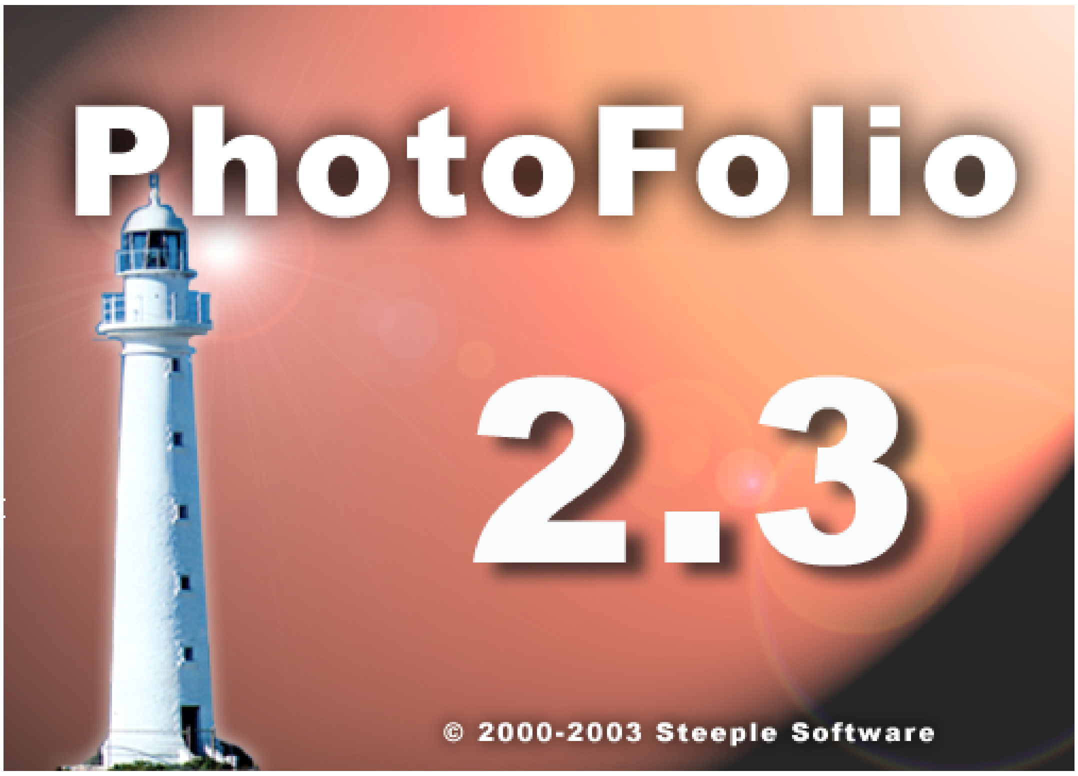


English document, version 1.0

1st January 2003

Quick contents

Major window List

Copyright and warranty information

PhotoFolio license agreement

PhotoFolio and its related documentation, utilities, and examples (hereinafter "program", "programs" or "the program") are © Copyright 1999-2003 Steeple Software. All Rights Reserved.

1. Specific third party files from the PhotoFolio distribution may be reproduced and distributed separately under their own licensing arrangements subject to agreement with the third party authors.

2. The program may not be reproduced, modified or sub-licensed in any way.

3. The license to use this copy of the program is extended to the original purchaser only and is not transferable. Upgrades and technical support are available to the original purchaser only.

4. The program and related documentation, utilities, and examples provided by Steeple Software may not be reproduced or distributed, in whole or in part.

5. The licensee shall hold Steeple Software and its management and employees harmless from any and all claims, damages and liabilities resulting from or arising out of the programs use.

6. Steeple Software may terminate this license at any time, with or without cause. This license and the Licensee's rights herein will automatically terminate upon any breach of these terms and conditions. Upon any such termination, the licensee shall destroy all copies of the program and the related documentation, utilities, and examples in its possession.

7. You are not required to accept this License, since you have not signed it. Should you choose not to accept this License, you should return the program to the place of purchase.

Warranty

1. Except when otherwise stated in writing the copyright holders and/or other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you.

2. In no event unless required by applicable law or agreed to in writing will Steeple Software and its management and employees be liable for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to the loss of data or data being rendered inaccurate or losses sustained by you or a failure of the program to operate with any other software or hardware), even if you have been advised of the possibility of such damages.

Disclaimer

No guarantee of any kind is given that the program described in this document is 100% reliable. You are using this software at your own risk.

PhotoFolio is supposed to be used to view and catalogue images. Even though every effort has been made to make PhotoFolio as compatible as possible, Steeple Software cannot rule out the possibility that PhotoFolio might have bugs that have side effects (possibly harmful) on your system.

Steeple Software hereby rejects any liability or responsibility for these or any other consequences from the use of PhotoFolio whatsoever. This includes, but is not limited to, damage to your equipment, to your data, personal injuries, financial loss or any other kinds of side effects.

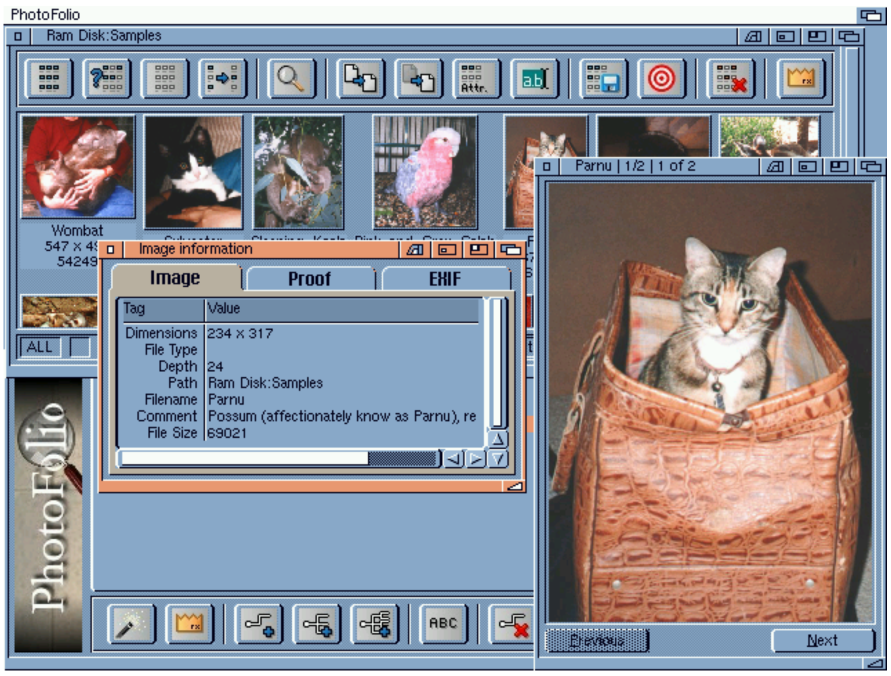
PhotoFolio is provided as-is. This means Steeple Software does not guarantee that PhotoFolio is fit for any specific purpose and Steeple Software does not guarantee any bug fixes, updates or help during error recovery.

Introduction

PhotoFolio 2.3, Copyright © 1999-2003 Steeple Software, distributed by Schatztruhe

http://steeplesoftware.dragonfruit.org/

PhotoFolio is designed for anyone wishing to view, store or catalogue images. It runs on any Commodore-Amiga (020 and above), Draco computer, or Amiga emulator running Kickstart 3.0 or higher.



Tag	Value
Dimensions	234 x 317
File Type	
Depth	24
Path	Ram Disk:Samples
Filename	Parnu
Comment	Possum (affectionately know as Parnu), re
File Size	69021

Features include

- Configurable generation of proofs.
- Amiga User Interface Style Guide compliant user interface using MUI.
- Compatible with OS 3.5 and OS 3.9.
- Displaying of images, including scaled image loading and external viewer support.
- Cataloguing of images into owners (255), groups (4) and categories (22).
- Copying, moving, renaming, deleting and saving of images and proofs.
- Optional gamma correction of proofs.
- Limited PhotoAlbum compatibility (JPEG thumbnails only).
- Any number of windows can be opened at once (limited by system resources).
- Online help.
- Datatype support.
- CyberGraphX support or works with any native Amiga display mode.
- VLab image grabbing support.
- Drag and drop across windows.
- English, Spanish, Italian, German, French, Dutch, Czech, Polish, Swedish translations available.

Overview

Throughout this document, the word *proof* is used. A proof is a scaled down representation of an image. Some applications call these *thumbnails.*

Online help via AmigaGuide is available at all times. Position the mouse over the area of the *active* window you want help with and press the `Help` key.

PhotoFolio uses proof.library to create proofs. These can then be stored on disk for later use. On average a proof that is about 85 by 85 pixels in size takes up about 2-3k of disk space. This makes the program very efficient when it comes to both speed and storage.

Proofs can be stored in a mirrored directory structure in which case the proof maintains the same name as the original image (linked), or a proof can be stored along with the original image in which case the proof is given a name different to the image's name (non-linked).

Of the two types of proofs stored on disk, we recommend using linked proofs in preference to nonlinked proofs. There are a number of reasons for this:

- Linked proof files can be moved to different disk locations without disturbing the original images they were created from.
- Proofs created for nonwriteable media (e.g. CD-ROM) can be stored on hard disk and accessed later whilst retaining a link to the otiginal image.
- Your original image directory does not become cluttered with proof files that you will see when accessing the directory from other applications.

Throughout this document there are visual indicators to assist in locating important information:

Power user

This information is intended for experienced users. Inexperienced users can most likely disregard this information.

Important information

Important notes about a function of PhotoFolio are contained here.

New feature

This is a new feature in PhotoFolio 2.3.

Requirements

All software required to run PhotoFolio is included in the PhotoFolio distribution. PhotoFolio requires at *minimum* the following to run. Newer versions of these libraries and modules may include additional features that are also supported by PhotoFolio.

Minimum requirements

- A 68020 based Amiga computer running Kickstart 3.0 (revision 39.106) and Workbench 3.0 (revision 29.29)
- MUI version 3.8 (included) Available from http://wuarchive.wustl.edu/aminet/dirs/util_libs.html
- Listtree.mcc version 17.53 (included)
- NList.mcc and NListview.mcc version 19.90 (included) Available from http://wuarchive.wustl.edu/aminet/dirs/dev_mui.html
- jpeg.library version 6.2 (included) Available from http://steeplesoftware.dragonfruit.org.
- proof.library version 2.9 (included) Available from http://steeplesoftware.dragonfruit.org.
- imageio.library version 3.5 (included) Available from http://steeplesoftware.dragonfruit.org.
- imageprocess.library version 2.0 (included) Available from http://steeplesoftware.dragonfruit.org.
- guigfx.library version 16.2 (included) Available from http://www.neoscientists.org/~bifat/binarydistillery/guigfxlib.shtml.
- render.library version 30.0 (included) Available from http://www.neoscientists.org/~bifat/binarydistillery/renderlib.shtml.

Additional support software

To display extended information about images:

- exif.library version 1.0 (included) Available from http://steeplesoftware.dragonfruit.org.

If you have a VLab card and want to grab images:

- VLab.mcc version 13.0 (included) Available from http://steeplesoftware.dragonfruit.org.

If you want to use internal viewer support for MysticView.library:

- mysticview.library version 4.2 (included) Available from http://www.neoscientists.org/~bifat/binarydistillery/mysticlib.shtml.

To click on http and email links in the about window:

- openurl.library version 3.0 (included) Available from http://wuarchive.wustl.edu/aminet/dirs/comm_www.html.

PhotoFolio automatically detects if a CyberGraphX card is installed and uses the native CyberGraphX calls for optimum speed and less memory overhead. If a CyberGraphX card is not found then PhotoFolio uses guigfx.library to render to the screen automatically.

Language support

PhotoFolio supports language catalog files as introduced with Workbench 2.1 locale.library. If you wish to create your own national translation you should consult the blank translation file **PhotoFolio.ct** that resides in the `catalogs` directory of the release archive.

If you have finished creating a translation table and want to share it with others, send it to us for everyone else to use.

Reporting bugs

Please report all bugs!

Bugs can be reported via the bug report form online at http://steeplesoftware.dragonfruit.org/amiga/photofolio/support.html.

We are striving to make PhotoFolio the best program of its kind and we have tried to write a bug free program, but it's nearly (always?) impossible. Some (many?) bugs always make it through the testing.

If you come upon one of those undesirable features which even we and our testers were unable to track down and remove, follow these steps:

1. Read the documentation and make sure you are doing things correctly.
2. Check on the PhotoFolio mailing list to see if others are having similar problems or that there is a known problem.
3. Ensure no *patch* programs are running and causing problems.
4. Make sure you have all required libraries etc and that they are of the required minimum versions. You can use PFSysInfo included in the distribution to assist in this process.
5. Describe your problem as elaborately as possible, explaining **every** step taken to reproduce the problem.
6. State your system and program configuration. It helps to know on which machine there are problems. Include information such as memory size, Amiga model (A1200, A4000, etc.), graphics hardware (ECS, AGA, etc.), CPU type (MC68020, MC68040, PPC etc.).

Contacting Steeple Software

For the latest patches, upgrades and information on PhotoFolio, check the PhotoFolio home page at http://steeplesoftware.dragonfruit.org/.

To join the PhotoFolio users mailing list, point your browser at http://groups.yahoo.com/group/photofolio or send an email to photofolio@yahoogroups.com.

It is always satisfying to receive feedback from others (good or bad) and to receive requests for enhancements.

Authors

If you wish to contact Steeple Software use the address below.

Email: steeplesoftware@dragonfruit.org

To contact the author’s directly, use the following contact details:

Steve Quartly
1280 Stevens St
Mundaring, Perth,
Western Australia 6073
Australia
Email: steve@quartly.net

Paul Huxham
Email: paulhuxham@yahoo.com

Developer systems

System 1	**System 2**
Amiga 3000/040	Amiga 4000/060
Kickstart 3.1, OS 3.5	Kickstart 3.1, OS 3.9
14 Meg RAM	42 Meg RAM
1 Gig Hard Disk	9 Gig Hard disk
CyberVision 64/3D	CyberVision 64/3D
SAS/C 6.55	SAS/C 6.58

Acknowledgements

Steve and Paul would like to extend special thanks to a great many people who have helped to bring PhotoFolio to release state. The list below is by no means complete or in any particular order and we hope you will forgive us if we forgot you.

from **Steve Quartly**

I spend many long hours in front of my computer screen writing software. My wife Debra, whom I adore, has not complained once. In fact she encourages me to write software. Many times she has said to me... "I'd rather you wrote software, than spent your time and our money at the pub every night." To you Debra, I thank you for your patience and love.

from **Paul Huxham**

I want to thank my wife Tuyen, who reluctantly watches me wander off to write more code, or run off to press enter to compile something during a movie. You put up with a lot my dear!

Stefan Stuntz

The author of MUI. Wow, what a great programming interface. PhotoFolio wouldn't be half of what it is without MUI. Thanks Stefan.

This application uses **MUI – MagicUserInterface** © Copyright 1992-97 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 Muünchen
GERMANY

Support and online registration is available at http://www.sasg.com.

Timm S. Muëller

The author of mysticview.library, guigfx.library and render.library. Without guigfx.library and render.library, PhotoFolio would be limited to CyberGraphX users only. Our thanks go to Timm for these **very** useful libraries. PhotoFolio supports MysticView internally, this was also made possible by Timm who wrote MysticView and made the library available to users. After endless sleepless nights and a truck load of e-mails, we got it working. Thanks a heap Timm!

This application uses **MysticView** by Timm S. Muëller.

mysticview.library is the third abstraction layer on top of guigfx.library and render.library.

mysticview.library provides a single display class running asynchronously in the background. It can be supplied a variety of attributes.

This library is freeware.

Requirements:

- OS 3.0 (v39)
- 68020, minimum suggested: 68030/40
- guigfx.library v16 (dev/misc/guigfxlib.lha)
- render.library v30 (dev/misc/renderlib.lha)

Supports:

- Higher CPU, FPU
- OS 3.1 (v40)
- CyberGraphX, Picasso96

Dieter Muëller

Release advice and support.

Schatztuhre

Distributing PhotoFolio and continuing support of the Amiga platform

PhotoFolio owners

Without you, PhotoFolio wouldn't exist.

Our translators

Oliver Esberger, Lucca Longone, Vit Sindlar, Victor Gutiérrez, Gérard Leyne, David Ceulemans, Daniel Sternik

Beta testers

Luca Longone, Kevin Twyman, Oliver Esberger, David Ceulemans, Andreas Kleinert, Dieter Muëller, Vit Sindlar

Natural sorting algorithm

Sorting algorithm for *natural order* comparisons of strings in C, Copyright © 2000 by Martin Pool mbp@humbug.org.au

PPM loader

The ppm loader in imageio.library contains code based on the pbm utility library.

pbm utility library

Copyright © 1988 by Jef Poskanzer.

Image processing code

The image processing code in imageprocess.lbrary is based on:

PBM – Copyright © 1988 Jeff Poskanzer
FBM – Copyrigth © 1989, 1990 Michael Mauldin

Tip of the day

Tip of the day (MUI custom class) is freeware software, but is copyright © 1999, 2000 Marcin Orlowski carlos@amiga.com.pl.

PhotoAlbum

PhotoAlbum is Copyright © 1996-1998 by Helmut Hoffmann

Also

- Amiga Inc - Amiga
- Haage and Partner - Amiga platform support
- Olaf Barthel - So much excellent source code, OS 3.5, OS 3.9 and more

Installation

PhotoFolio uses the standard operating system installer for installation. If you are unsure how to use the Amiga Workbench and operating system interface, please refer to your Amiga manual.

PhotoFolio makes extensive use of MUI. Refer to the MUI documentation for details of what MUI provides and how to use the interface. (Some specific notes on aspects of MUI used intensively in PhotoFolio are in user interface notes).

Developer debugging tools are designed for program development and it is not recommended that they are run during any normal program use.

During installation, if PowerPC modules are available and you have a PowerPC, an option to install the PowerPC module appears.

Installation proceeds in the following order

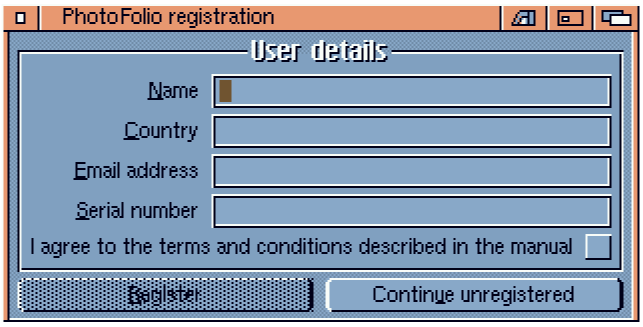
- MUI - User interface (if installation of MUI is required you will need to reboot following MUI installation and run the PhotoFolio installation script again to continue)
- listtree.mcc - MUI class
- NList.mcc and NListView.mcc - MUI class
- jpeg.library and preferences program - jpeg image format support
- proof.library - proof management
- imageio.library and preferences program - image loading and saving
- imageprocess.library - image resizing and manipulation
- guigfx.library and render.library - User interface image display
- PhotoFolio - Application

Optionally installed items

- exif.library - Extended image information
- pfDirectoryView.mcc - An alternate directory view window
- pfExportHTML - Export proofs as HTML pages
- VLab.mcc - VLab image grabbing
- mysticview.library - MysticView viewer support
- url.library - Web and mail links
- Documentation
- Sample ARexx scripts
- Language catalogs

Registration

When PhotoFolio is run after installation a registration window appears.



You need to enter:

1. Your name
2. Home country
3. Email address (if you have one)
4. PhotoFolio serial number
5. You must agree to the terms and conditions in the manual (PhotoFolio.readme or online AmigaGuide help file, contained in the distribution)

The registration window has context sensitive help (if you chose to install the help files during installation). Position the mouse pointer over the item you require assistance with and press the `Help` key.

When all the required information is entered, the register gadget becomes active. Clicking register stores your registration information so that you do not need to enter it again the next time you run Photofolio.

Continuing unregistered allows you to run PhotoFolio without entering the required registration information but with certain restrictions.

The only times you need to reregister PhotoFolio are:

- Reinstallation
- Upgrading to a new version

If you have an internet connection Steeple Software would appreciate it if you would take the time to register PhotoFolio at the online registration page. This only needs to be done once and assists us with the future development of PhotoFolio as well as providing us with a way of contacting you regarding updates. All information entered is treated with strict confidence and is not stored on publicly accessable servers.

User interface notes

This information is not a replacement for the MUI documentation but is intended to attempt to point you in the right direction if you are having some problems getting MUI to do what you want.

- The appearance and functionality of MUI can be changed dramatically from the MUI settings window. If you find that some things do not appear or function as described in this document, try using the MUI default settings (after a clean MUI installation) for all MUI applications.

- PhotoFolio **can** run on any Amiga screenmode. This is accomplished through MUI, not directly with Photofolio (although you can access the MUI settings window from within PhotoFolio). A screenmode has to be created to run a MUI appplication on using PSI in MUI settings. This screenmode is then set as the screenmode for PhotoFolio to use.

  Running PhotoFolio on a screen other than Workbench prohibits the dropping of Workbench icons on PhotoFolio.

- MUI allows each application to be assigned a commodity key. PhotoFolio uses a commodity key to bring PhotoFolio to the front and make it the active application. Assign a commodity key for PhotoFolio in MUI settings if you require this functionality.

- PhotoFolio uses **context menus**, which appear when you use the right mouse button with the mouse pointer positioned over an object (such as a proof, gadget or window area) in the **active** window. To access the regular Amiga menus, move the mouse pointer near to the top of the screen or away from where context menus appear and use the right mouse button.

- You can resize MUI windows. If you want MUI to remember your window sizes and positions you need to snapshot the window by:

  1. Using the MUI window snapshot gadget in the windows' border (available if enabled in MUI settings).
  2. Enable remember window size and positions in MUI settings.

  PhotoFolio windows initially appear at a reasonable position and size.

- Drag and drop operations are visually different for Workbench and MUI operations. The way MUI displays a drop area can be configured in MUI settings.

  1. MUI objects (proofs etc) can be dragged onto other PhotoFolio windows. MUI displays a drop area (usually as a dotted line around the droppable area) as the dragged object passes over a droppable area.
  2. Dragging and dropping a Workbench icon onto a MUI application will not display the MUI drop area.
  3. MUI objects in PhotoFolio (proofs etc) **cannot** be dropped onto the Workbench **or** other MUI applications.

Workbench and Shell

PhotoFolio can be started from the Workbench or Shell.

From the Workbench, PhotoFolio is started by double clicking on the application icon **or** on a project icon.

Only one copy of PhotoFolio can be running at any time. If you attempt to run another copy of PhotoFolio, the existing PhotoFolio comes to the front. (If iconified it uniconifies itself first and then comes to the front.)

Double clicking a project with PhotoFolio already running, loads the project into the existing PhotoFolio. The same applies for paths or projects specified in a Shell.

If multiselected PhotoFolio project icons are double clicked (shift selecting multiple projects and then double clicking the last), only the first selected project is loaded. Other multiselected projects are ignored.

The behaviour of PhotoFolio can be changed by adding tool type entries to it's Workbench icon or by specifying additional command line parameters when running from the Shell.

Supported keywords

PATHNAMES

(Shell only)

Any number of paths can be specified from the Shell and PhotoFolio opens each one into its own browse window and adds the paths to the path list. Invalid paths are ignored. Specifying a path from the Shell overrides the default load path or project.

Items added from shell are added to the **Shell** folder in the path list.

PROJECT/PROJ

(Shell only)

Specifies a project to load into PhotoFolio. If a project **and** pathnames are specified, the project is loaded first and then the specified pathnames are added.

SPLASH

When added to the PhotoFolio program icon, a splash screen is shown while PhotoFolio is initializing. This can also be changed from the settings window.

SETTINGS=path/filename

If added to the PhotoFolio program icon, indicates a settings file to load at startup. If added to a PhotoFolio project file's icon, indicates a settings file to load when that project is loaded from the Workbench or by drag and drop. (This allows per project settings files using save settings as.)

GUIGFX=(YES|NO)

Forces PhotoFolio to use guigfx.library regardless of whether CyberGraphX is installed or not. Under normal circumstances, this setting should be left disabled.

NOTIPS

Forces PhotoFolio not to open the tip of the day MUI custom class, even if it is installed.

For example

```
PhotoFolio "Images24:rendered" "Pictures:Images/sunset.jpg"
```

The first path is loaded into a separate browse window and added to the path list. The last path is ignored because it contains a filename. If however */sunset.jpeg* is a directory then it is loaded into another new browse window.

ARexx

PhotoFolio sports a fully equipped ARexx interface. Commands are provided for the creation and manipulation of browse windows, proofs, show windows, projects, program settings and more.

The full ARexx command set is documented in the `PhotoFolio_ARexx.guide` file contained in the Documentation directory of the release archive. Some example scripte are also included in the ARexx directory of the release archive.

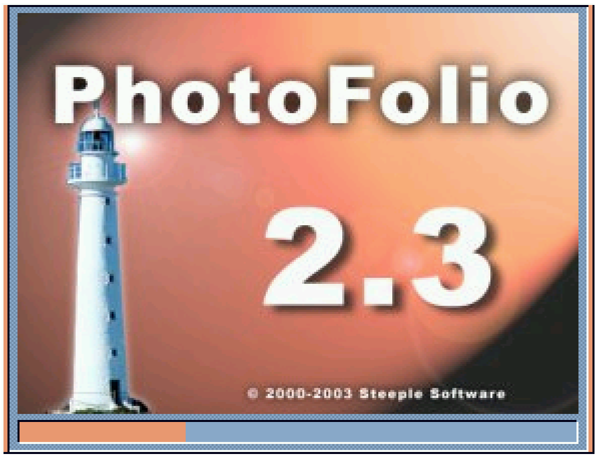
PhotoFolio allows ARexx scripts to be added to the main and browse window menus from the settings window.

ARexx scripts can also be dragged and dropped from the Workbench onto the main window, browse windows and to the ARexx panel of the settings window. Scripts dropped on to the main and browse windows are executed in the context of the window they are dropped on. Dropping onto the settings window adds the script to the list.

How to use PhotoFolio - A quick guide

Double click the icon to run PhotoFolio

When PhotoFolio is run, a splash screen opens showing the progress of intializing and setting up PhotoFolio. The splash screen can be disabled from the settings window or by editing the tooltype in the PhotoFolio program icon.



After intialization is complete the splash window closes and the PhotoFolio main window opens containing the path list. This path list contains any numbers of directory paths that you may want to view. When the mouse is over the path list, use the right mouse button to show a context menu that has options to add paths to the path list:

- Add one path
- Add all paths
- Add all paths recursively

Once paths have been added to the path list, you can view them by:

1. Double clicking the path.
2. Position the mouse over the required path and select Open in new window from the context menu.
3. Select new browse from the main menu and then drag the path into that window.
4. Select the path by single clicking it and then click the magic wand in the toolbar.

The browse window can show the proofs full size, categorise them and sort them. Drag proofs to other browse windows, move, copy, rename or delete them.

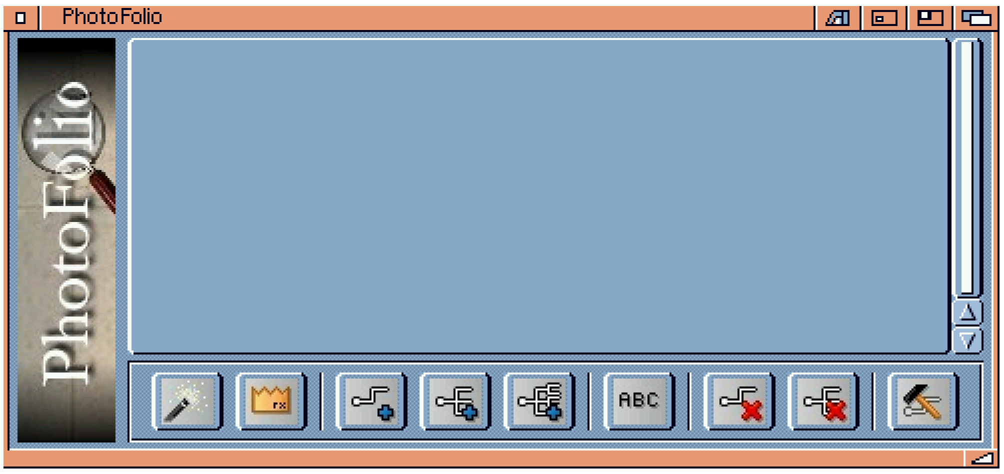
When the path you have selected contains images, the loading of the browse window will be slow, because PhotoFolio has to load the images, scale them and generate proofs. Once done, you can save the proofs. When proofs are saved, they are saved to the proof destination directory. The next time you wish to view these images, don't load the original image path, load the path containing the proofs. The proofs contain a link to the original image and you can then perform all the operations on the proof as if it was the original image. Loading is also a lot faster.

If you run low on memory and can't load all the proofs, PhotoFolio stops loading and displays a set of paging gadgets at the bottom of the browse window, allowing you to step back and forward through pages of proofs. Moving from one page to another loads another page of proofs.

If you have a lot of images to browse, it is better to *make* proofs for them before opening a browse window. To do this, position the mouse over the path you want to make and then select *make path* from the context menu. Or, select *make all paths* from the context menu to make them all. A make window opens. From here you can set parameters and default categories and then start making. This is quicker than opening a browse window because PhotoFolio doesn't have to display the proofs as they are created. When the image path is made, the path the proofs are stored in is added to the **Proofs** folder in the path list. This path can be opened in a browse window to view the created proofs.

Main window

PhotoFolio's main window consists of a path list and an optional toolbar. Path list entries are opened into browse windows. These paths should contain images and/or proofs you wish to view or catalogue.



If you are unsure of what each gadget in the toolbar does, position the mouse over the gadget and a help bubble appears.

Adding and removing paths and indeed any operation on the path list is done by the use of the main windows path list menu or the path list context menu. The context menu contents vary according to what the mouse is over at the time the right button is pressed and the contents of the path list.

The path list can contain folders; that can contain other folders and/or paths. Drag and drop any path or folder to any other path or folder. Double clicking a path list folder adds a path to that folder.

Paths that already exist in the path list are not added again.

Folders are created automatically for default operations:

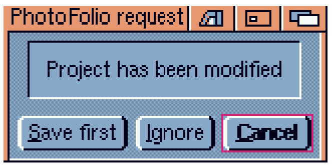
1. *Add path* - Add path commands
2. *Shell* - From Shell launch
3. *Default path* - In settings
4. *Workbench icon* - From dropped icons
5. *Browse window* – Paths added from set browse path
6. *ARexx* - From ARexx interface
7. *Proofs* -If enabled, after make proofs

A path can be opened in a browse window several ways:

- From the context menu item open in window with the mouse pointer over the path to display.
- Double click on the path to display.
- Drag and drop a path to an already open browse window.

- Select a path and click the magic wand gadget in the toolbar.

When the current project is about to be modified in a major way (e.g. erasing all paths in the current project); if it hasn't been saved, you are given the choice to *save first*, *ignore* or *cancel*.


PhotoFolio request

Project has been modified

Icons are saved along with project files.

Saving project icons can be disabled in settings.

ARexx scripts can be dragged and dropped from the Workbench onto the main window to execute the script in the main windows context.

Menus

The main window’s menu contains the following items that operate on the project, entries in the path list, ARexx scripts and program settings.

Project

The *project* menu contains items that operate globally on the current project.

About…

Opens the program about window containing version and other information.

About MUI…

Opens the about MUI window.

New

Erases the current PhotoFolio project. If modifications to the current project are unsaved, a requester opens prompting to *save first*, *ignore* or *cancel* before removing all entries in the path list

Open…

Loads a previously saved PhotoFolio project from an ASL requester. All current entries in the path list are erased.

If a project file is dropped from the Workbench onto the path list, it is loaded.

Recent
Projects previously loaded and saved appear listed under this menu. Selecting one of the recent files loads that project. The most recently accessed projects appear at the top of the list.

The number of files in the list can be changed in the settings window.

Append…
Appends a previously saved PhotoFolio project to the current project. The paths in the project are added to the current entries in the path list.

Save
Saves the current contents of the path list to the last loaded, or last saved PhotoFolio project file. If the project has not yet been saved an ASL requester prompts for a path and filename.

Save as…
Opens an ASL requester, prompting for a path and filename to save the current project as.

Information…
Opens the project information window, which maintains information about the current project.

Iconify
All PhotoFolio windows are closed and a PhotoFolio icon appears on the Workbench.

If PhotoFolio is loading when iconify mode is entered, PhotoFolio pauses all operations until it is uniconified.

Double clicking on the iconified PhotoFolio icon reopens all PhotoFolio windows and continues operating.

Quit PhotoFolio
If confirm quit is enabled, a requester appears asking for confirmation before quitting.

Window
The *window* menu contains items that open new windows and manipulate the error window.

New browse…
Opens a new empty browse window.

New show…
Opens a new empty show window. The type of show window that is opened depends on the default viewer setting.

New information…
Opens a new empty information window.

New Directory View…
If the directory view module has been installed correctly an empty directory view window is opened.

Path palette
Opens and closes the path palette window. The tick mark (√) indicates the current status of the path palette window (open or closed).

Error window…
Opens and closes the error window. The tick mark (√) indicates the current status of the error window (open or closed).

Browse
The submenu contains a list of the currently open browse windows listed by their displayed path. Selecting an item brings that window to the front and makes it active. If there are no browse windows open this item appears ghosted.

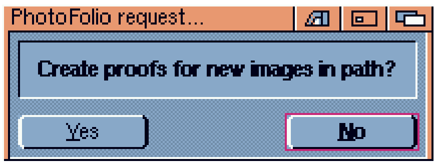
Show
The submenu contains a list of the currently open show windows listed by their displayed filename. Selecting an item brings that window to the front and makes it active. If there are no show windows open this item appears ghosted.

Path list
The *path list* menu contains items that operate on entries in the path list.

Make path…
Opens a make window allowing the creation of proofs for the selected entry in the path list without having to open a browse window.

Make all paths…
Opens a make window allowing the creation of proofs for all entries in the path list without having to open a browse window.

Create proofs for new images…
Generates proofs (and saves them to disk) for new images in the path that currently have no proof. When new images are added to a path that has already had proofs made, this function can build proofs for the newly added images instead of having to regenerate the entire path.



Create proofs for new images in path?

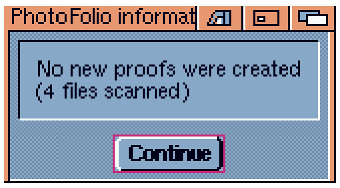
A progress window opens while the list is scanned for images that have no proofs. Proofs are then generated for images found with no proofs.

There are a few rules for the creation to work correctly:

1. The path selected for processing **must** be a path that contains proofs.
2. The path to look for images in is taken from the first proofs encountered during the scan.
3. Proofs that have been moved (either by move, drag and drop or another application) will either be regenerated or not processed.

If the original generated proof path has been distrubed it is likely that proofs wil be created that are not required or they will be created in the wrong place.

If the process is cancelled, images that have not had proofs generated will remain in the path. A report window opens when the procedure is completed or cancelled indicating the number of proofs created and processed.


PhotoFolio informat

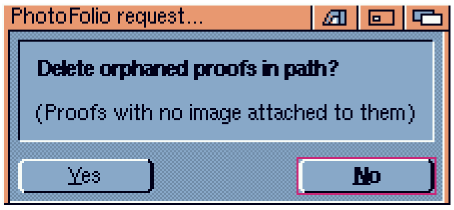
No new proofs were created
(4 files scanned)

Create proofs for new images in all…

Performs create proofs for new images on the selected item in the path list **and** on all folders and entries recursively in the selected item *in the path list*. Directories that do not appear in the path list are not processed.

Delete orphaned proofs…

Deletes proofs that have no image associated with them. This can occur if image files are manipulated (moved etc) from within another application.

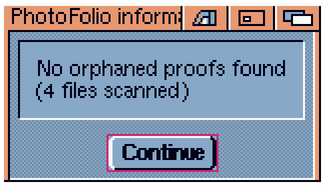

PhotoFolio request…

Delete orphaned proofs in path?

(Proofs with no image attached to them)

A progress window opens while the list is scanned for orphaned proofs which are then deleted.

If the process is cancelled, any proofs that have been deleted already remain deleted. A report window opens when the procedure is completed or cancelled indicating the number of proofs deleted and processed.



No orphaned proofs found
(4 files scanned)

Delete orphaned proofs in all…
Performs delete orphaned proofs on the selected item in the path list **and** on all folders and entries recursively in the selected item *in the path list*. Directories that do not appear in the path list are not processed.

Add one path…
Opens an ASL requester allowing the selection of one path to add to the path list.

Add all paths…
Opens an ASL requester allowing the selection of a path that has all directories in it added to the path list. This is **not** a multi-select requester. Select the path to add and PhotoFolio scans that path for other directories inside it and adds them to the path list. This is a non-recursive function.

Add all paths recursively…
Opens an ASL requester allowing the selection of a path that has all directories in it added recursively to the path list. This is **not** a multi-select requester. Select the path to add and PhotoFolio scans that path recursively and adds all directories inside it to the path list. This is a recursive function.

Clear all made
As proofs are generated using the make window, PhotoFolio changes the colour of the created path in the path list to white so you can tell at a glance which paths have been made. This changes the status of all path entries back to not made (black).

Remove all paths/folders
Removes all the paths and folders from the path list. If the project is unsaved, PhotoFolio asks for confirmation before removing all entries.

Sort alphabetically
Sort all the entries in the **selected** path list folder alphabetically.

ARexx
The ARexx menu contains items that allow the execution of ARexx scripts.

Run ARexx script…
Opens an ASL requester and executes the selected ARexx script.

ARexx scripts
The ARexx scripts selected for the main window in settings are displayed here. Select the ARexx script from the menu to execute it. Keyboard shortcuts are automatically created for the first 10 scripts (`Left Amiga 0-9`).

Settings

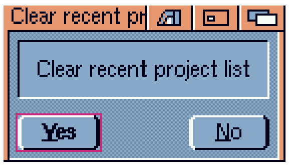
The *settings* menu contains items that configure the operation and appearance of PhotoFolio.

Settings…

Opens the PhotoFolio program settings window.

Clear recent project list…

Removes all entries from the recent project list. If there are items in the list, you are prompted before they are erased.



Clear recent pr

Clear recent project list

This menu item appears ghosted if there are no entries in the recent project list.

MUI…

Opens the standard MUI preferences window.

Load settings…

Opens an ASL requester allowing selection of a previously saved settings file to load.

Save settings as…

Opens an ASL requester allowing selection of a file to save the current program settings as.

Save settings as default

Saves the current program settings in the file `PROGDIR:PhotoFolio.prefs`.

Help

The *help* menu contains items that might assist you with using PhotoFolio.

Online AmigaGuide…

Opens the `PhotoFolio.guide` help file (if installed).

Show tip of the day…

Shows the tip of the day window. If the tip of the day MUI class is not installed, this menu item does not appear.

Context menus

Path list context menus appear when the right mouse button is pressed while the mouse pointer is over entries in the path list. The contents of the context menu vary depending on the item the mouse was over when the context menu was activated.

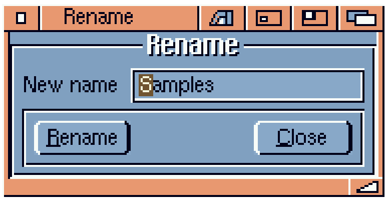
Most of the menu items perform the same functions as items in the main windows path list menu. Only those items that have not previously been discussed are listed here.

The title of a path list context menu displays the entry’s' name so you can be sure which entry you clicked on.

Open in new window…
Opens a new browse window and displays the path that the mouse pointer was over in it.

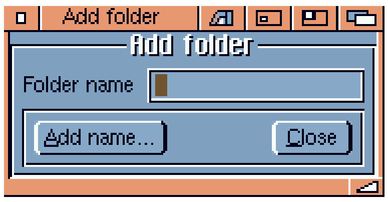
Open in Directory View…
If the directory view module has been installed correctly, the path the mouse pointer is over is opened in a directory view window.

Rename…
Changes the *displayed* name of a path or folder. If a path is renamed, the entry displayed is changed even though the actual path loaded remains the same.


Rename

New name: Samples

Add folder…
Adds a new folder to the path list. Folders are added in the currently selected folder.


Add folder

Folder name:

Remove path/folder
Removes the path or folder that the mouse pointer is over from the path list.

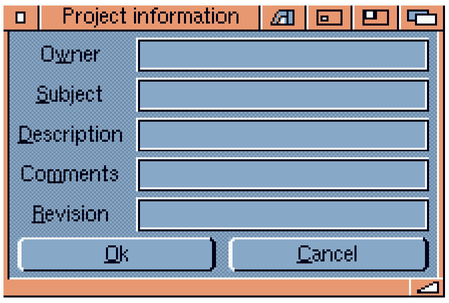
Clear made
Marks the path the mouse pointer is over as being not made.

Retarget…
Allows retargeting of all proofs in the path under the mouse.

Show path…
Shows all the files in the selected path in a show window in sequence.

Project information window

Maintains information about the current project.


Project information

Owner	
Subject	
Description	
Comments	
Revision	

All of the gadgets can contain any alphanumerical characters. If you are maintaining many different projects, the information here may be used to jog your memory about certain aspects of the project.

This window can be set to automatically open when a project is loaded.

Proofs

A proof is a representation of an image. Right clicking on a proof shows the proof context menu.

Proofs can be selected by single clicking on them and deselected by single clicking them again. Selected proofs are indicated with a recessed border (the exact style and size of this border can be changed in MUI settings).

To operate on selected proofs in a browse window, right click anywhere in the blank space around proofs to show the selected menu.

Proofs can be dragged and dropped. For example to move a proof from one location to another, drag and drop the proof from one browse window to another browse window. Proofs can also be dragged and dropped onto existing show and information windows. MUI indicates a drop is allowed when the mouse pointer moves on top of a droppable area by highlighting the area (the appearance of the droppable area can be changed in MUI settings).

The way proofs appear in browse windows can be configured in the settings window.

Context menu

The proof context menu appears when the right mouse button is pressed while the mouse pointer is over a proof.

Information…

Opens an information window, showing information on the proof. Proofs can also be dragged and dropped onto existing information windows.

Show…
Shows the image the proof was created from in a show window. The default viewer can be configured in the settings window. If the proof's associated image cannot be found an error is displayed. Proofs can also be dragged and dropped onto existing show windows.

Copy…
Copies the image to a new path. An ASL requester requests the destination directory. When copying a file, the date stamp and file comment of the source file are copied to the destination file. If the file exists in the destination directory a requester opens confirming the overwriting of the file.

Copy checks if the destination is write protected and if so asks the user to *retry* or *cancel*.

Move…
Move a proof and/or its associated image to another location using the move window.

Correct gamma
Applies auto gamma correction to the proof image (not the original image). PhotoFolio can be configured so that all created proofs are automatically gamma corrected.

Change attributes…
Opens the change attributes window allowing the attributes of the proof to be changed.

Select to end
Selects proofs from this proof to the last proof in the window.

Select from beginning
Selects proofs from the first proof in the window to this proof.

Save proof
Saves the proof. If you don't want to make an entire directory of proofs, load the images directly into a browse window and then save the proofs you want to keep.

Save image as
Saves the image the proof represents. The available save formats are shown as a submenu under this menu item.

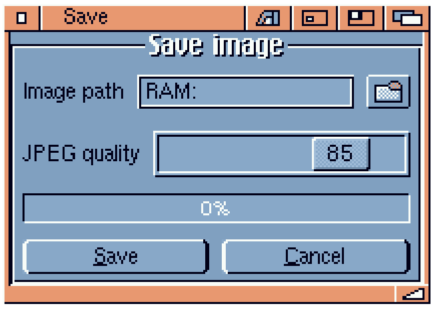


Image path: RAM:
JPEG quality: 85
0%

If a format other than jpeg is chosen, the *jpeg quality* slider does not appear.

The default save path can be configured in the settings window. The default jpeg save quality (if you choose to save in the jpeg file format) can also be configured there.

Open with

Selecting an item in the submenu, opens the proof's image with the selected application. The items that appear here can be configured in the settings window.

If no open with items are defined then this menu item does not appear in the context menu.

Unless the DOS command **Run** is used, the PhotoFolio program will be unable to be used until the application launched has quit.

Retarget proof

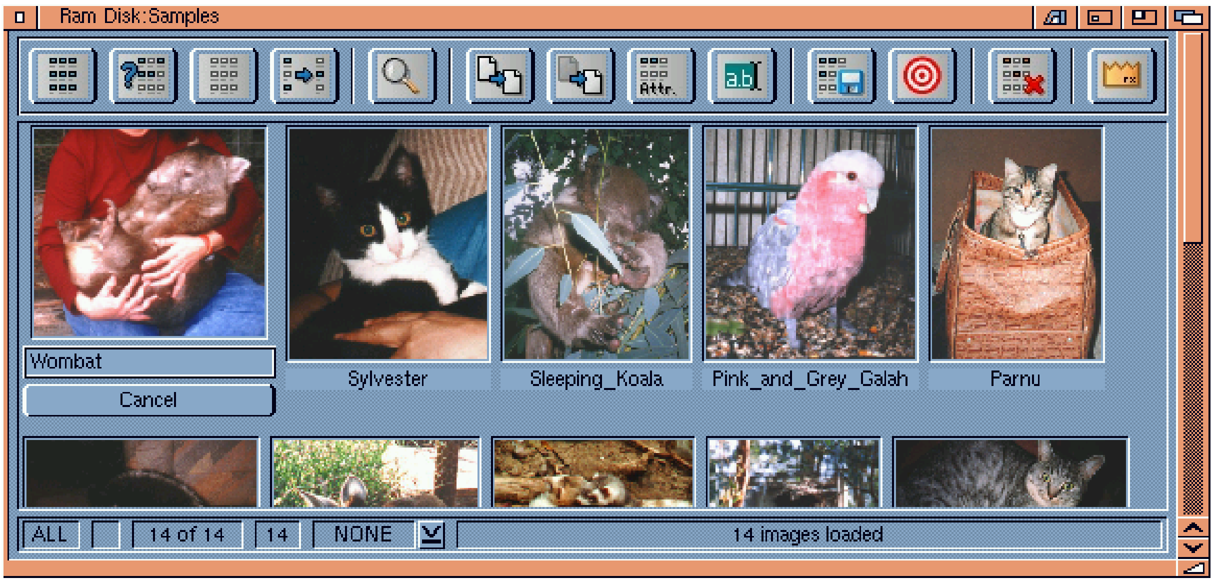
If the location of the original image has changed after a proof is created, this function is used to tell the proof where the image is now located. An ASL requester opens; select the path the image now resides in and click *ok*. The proof is then retargeted to point to the new location.

If an unsaved proof is being retargeted and settings are set to linked then the proof is saved using the retargeted path. If the settings are non-linked, an error message is displayed.

1. Non-linked proofs **cannot** be re-targeted.
2. If a proofs top line of text **is** bold then the proof has been saved. If the top line of text is **not** bold then the proof is unsaved. A red square displayed in the bottom right hand corner of a proof also indicates it has **not** been saved.

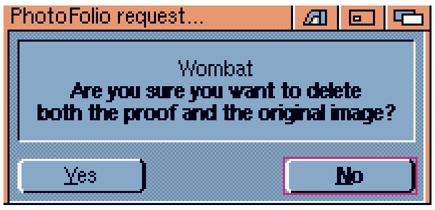
Rename image

The text below the proof is removed and replaced with a string and *Cancel* gadgets. The proof can be left in rename mode while the browse window is used as normal before actually renaming the image. Any number of proofs can be left in rename mode simultaneously. Pressing `enter` in the string gadget renames the image.



Delete…

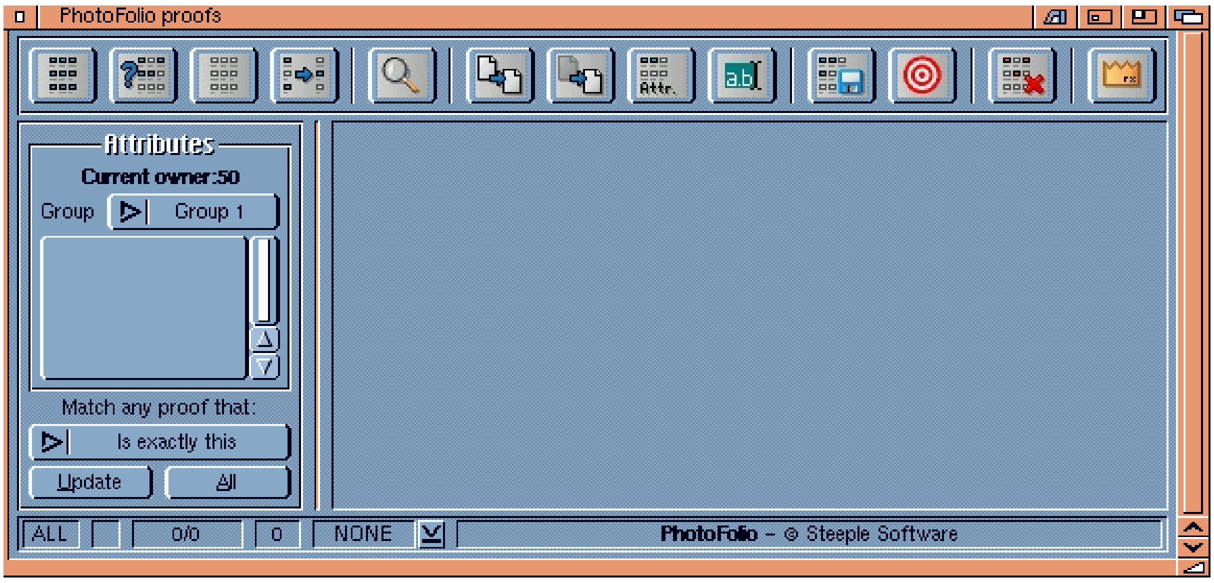
Deletes the proof **and** the image associated with it. A confirmation requester is displayed before anything is deleted.


PhotoFolio request...

Wombat
Are you sure you want to delete
both the proof and the original image?

Browse window

A browse window displays proofs of images.


PhotoFolio proofs

On the right side of the window is a scrollable area where proofs are displayed. On the left side is the attribute selection area, which is used to select which proofs are displayed in the browse window from the loaded proofs. The attribute selection area can be turned on and off, allowing more room for proofs. An optional toolbar can also be displayed in the top of the window.

If you are unsure of what each gadget in the toolbar does, position the mouse over the gadget and a help bubble appears.

When a new path to display is passed to a browse window, PhotoFolio begins by scanning the path and counting how many files it finds. When counting is complete it then attempts to load each file in the path. If an unrecognised file is found (e.g. a text or executable file) then the count is decremented, the file skipped and an error registered with the error window.

PhotoFolio recognises the difference between proofs and images, consequently you can mix the two in the same directory.

The scanning of a directory in a browse window, which establishes what files are to be checked as images and/or proofs, can take a long time for large directories or slow devices. Closing a browse window or clicking *stop* during a scan cancels the loading of the path.

After scanning all the files in the directory to load, PhotoFolio sorts the files in the order specified by the sort mode. Therefore if loading is paused or page mode is entered the display stills show proofs in the correct order.

If a proof is encountered it is loaded and displayed. In this case, the first line of the proof text will be **bold**. This indicates the proof is a saved proof and has not been generated on the fly from an image file. (The font used for text displayed beneath the proof can be changed in the settings window.)

If an image is encountered it is loaded, scaled, converted to a proof and displayed. Notice the first line of the proof text is **not** bold. (Unsaved proofs can also have a red square displayed in the bottom right hand corner of them in case you are displaying proofs without a text description.) This newly created proof is not stored anywhere, it exists in memory. If it is not saved before closing the browse window or before a new path is dropped onto the browse window, the proof is lost.

Closing a browse window closes all attached windows (e.g. information windows). If you need an information window to stay open after the browse window is closed, remember to detach it first.

Proofs can be *selected* by single clicking on them and *deselected* by clicking on them again.

If proofs are hidden using hide questionable they do not participate in select/deselect operations.

Proofs can be sorted using the sort menu items or the small popup gadget next to the sort status line in the bottom of the browse window.

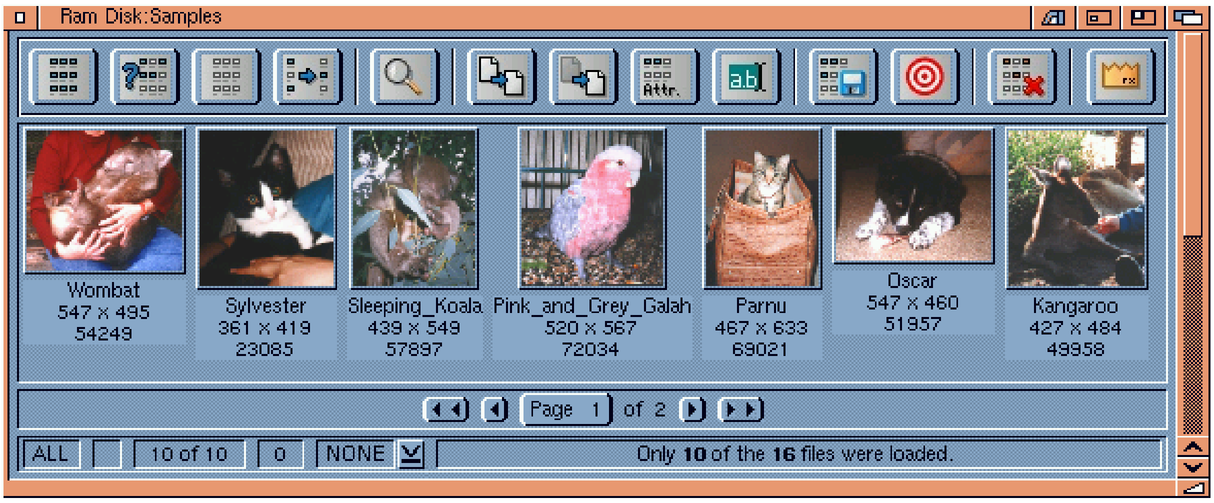
Double clicking a proof shows the image in a show window. The viewer used can be configured in the settings window. Double clicking an already selected proof does not affect its selected state.

The save status of a proof is indicated the following ways.

- A proof **with** bold text has been saved.
- A proof **without** bold text has not been saved.
- A proof **without** a red square in the bottom right hand corner has been saved.
- A proof **with** a red square in the bottom right hand corner has not been saved.

The load process can be stopped at any time by clicking the *stop* button in the bottom of the window or by closing the browse window. Loading can also be *pause*d and *resume*d.

If the safety memory limit is reached while loading, the load stops, a low memory error is displayed and page gadgets appear in the bottom of the browse window.



Only **10** of the **16** files were loaded.

PhotoFolio *tries to guess* how many pages of proofs are left. We must stress that it is only a guess because due to the Amigas multitasking ability, the amount of free memory can vary. For example you may launch another application while a page of proofs is displayed hence the free memory is lower than when you started and the page count will increase. Therefore don't count on the fact that a page will contain the same number of proofs all the time!

Right clicking on a proof shows the proof context menu.

Right clicking on an empty area of the browse window shows the selected proofs context menu. This menu contains operations that act on multiply selected proofs.

Images can be dragged and dropped from the Workbench onto a browse window. Dropping an image creates a proof for the image in the browse window (unrecognized file formats are reported to the error window). Proofs created in this way are unsaved and unless they are saved will be lost when the browse window is closed.

ARexx scripts can be dragged and dropped from the Workbench onto a browse window to execute the script in the browse windows context.

Drag and Drop

Proofs can be dragged from one browse window to another. To drag a single proof, make sure it is **not** selected and click and drag on the text below the proof. When a selected proof is dragged, **all** selected proofs are dragged.

To drag more than one proof, select the proofs to drag and then click and drag on the text of any one of the selected proofs. Dragging proofs shows, in the image being dragged, the total number of proofs being dragged.

Proofs can be dropped in the following places:

- Browse windows
- Show windows
- Information windows
- Path list in the main window

Rules to follow when dragging and dropping proofs

1. If a proof is dragged to a **new** browse window; that is an empty browse window with no path; then the window is considered a temporary storage area and the proof is not moved on disk, only in memory.

2. If a linked proof is dragged to a browse window that **has** a path, the proof is moved on disk to the browse windows' path, however the original image that the proof represents is not touched.

3. If a non-linked proof is dragged to a browse window that **has** a path, the proof **and** the original image are moved on disk to the browse windows' path.

4. If an unsaved proof is dragged to a browse window that **has** a path and you have linked set in settings, the proof is saved to disk in the browse windows' path, however the original image that the proof represents is not touched. If you have non-linked set then an error is generated and nothing is moved.

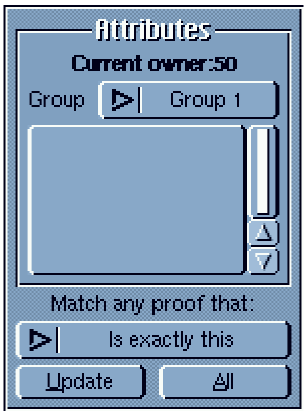
5. If during the move process an error occurs, e.g. not enough disk space or a file with the same name already exists, then the move of that proof is aborted, the error reported to the error window and PhotoFolio continues with the next proof.

6. Proofs **cannot** be dropped onto the Workbench or other MUI applications.

7. Dragging and dropping a proof that is already saved also copies the original proof's file comment and date stamp to the new proof file.

8. How the drop area appears can be configured in MUI settings.

Attribute selection area



The upper portion of the attribute selection area (current owner, group and category list) is explained fully in the change attributes window section.

After selecting categories in the list (use the `shift` key to select more than one), and choosing a *match mode*, proofs that match the category selection can be displayed.

The *match mode* specifies how the selected categories are applied during the selection process.

1. *Is exactly this* – For a proof to be selected, it's attributes must be exactly the same as those selected in the category list.
2. *Contains all selected* – For a proof to be selected, it's attributes must contain at least all the selected categories in the list.
3. *Contains any one* – For a proof to be selected, it's attributes can contain any one or more of the selected categories in the list.

Update applies the category selection to the proofs visible in the browse window. This allows more specific category matches to be applied for proof display.

All redisplays all proofs in the browse window allowing a new selection set to be applied to all proofs in the browse window.

Menus

The browse window's menu contains the following items that operate on the project, new windows, proofs in the browse window, browse window settings and program settings.

Project

The *project* menu contains items that operate globally on the current project.

About…

Opens the program About window.

About MUI…

Opens the about MUI window.

Iconify

All PhotoFolio windows are closed and a PhotoFolio icon appears on the Workbench.

Double clicking on the iconified PhotoFolio icon reopens all PhotoFolio windows and continues operating.

Quit PhotoFolio

If confirm quit is enabled, a requester appears asking for confirmation before quitting.

Window

The *window* menu contains items that open new windows and manipulate the error window.

New browse…

Opens a new empty browse window.

New show…
Opens a new empty show window. The type of show window that is opened depends on the default viewer setting.

New information…
Opens a new empty information window.

Scanners
The sub menu contains a list of image grabbing options that are currently available.

Vlab…
Selecting *VLab* opens a VLab grabbing window. This option appears ghosted if VLab hardware is not found or the VLab software is incorrectly installed.

Error window…
Opens and closes the error window. The tick mark (√) indicates the current status of the error window (open or closed).

Show errors
Displays errors generated from this browse window in the error window.

Close
Closes the browse window.

Browse
The submenu contains a list of the currently open browse windows listed by their displayed path. Selecting an item brings that window to the front and makes it active. If there are no browse windows open this item appears ghosted.

Show
The submenu contains a list of the currently open show windows listed by their displayed filename. Selecting an item brings that window to the front and makes it active. If there are no show windows open this item appears ghosted.

Proof
The *proof* menu contains items that operate on proofs in the browse window.

If proofs are hidden using hide questionable they do not participate in select/deselect operations.

Select all
Selects all the proofs currently displayed.

Pattern select…
Opens a pattern select window, allowing selection of proofs using the AmigaDOS pattern matching system.

Deselect all
Deselects all the proofs currently displayed.

Select unsaved
Selects all the proofs that have not been saved.

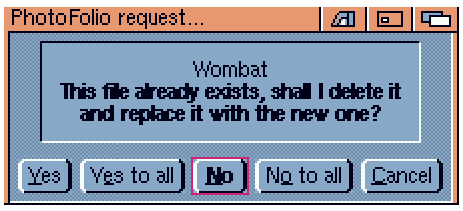
Toggle selected
Toggles all the selected proofs to deselected and vice versa.

Select orphaned proofs
Selects proofs where the image they were created from cannot be found. If the image a proof was created from has been deleted or moved from within another application, the proof will still exist. This command selectes those proofs so that they can be deleted, moved or retargeted.

Create proofs for new images…
Creates proofs for new images in the browse windows path following the same procedure as described for the main window.

Show selected…
Shows all the currently selected proofs in a show window.

Copy selected…
Copies all the currently selected proofs to a new location. If a file already exists in the destination directory, several options are presented as to how to overwrite now and for other files.


PhotoFolio request...

Wombat
This file already exists, shall I delete it and replace it with the new one?

A progress window opens during the copy process.

Move selected…
Move all the currently selected proofs to a new location. If a file already exists in the destination directory, several options are presented as to how to overwrite for this and for other files.

A progress window opens during the move process.

Correct selected gamma…
Corrects the gamma of all the currently selected proofs.

A progress window opens during the correct gamma process.

Change selected attributes…
Changes the attributes of all the currently selected proofs using a change attributes window.

Save selected proofs…
Saves all the currently selected proofs.

A progress window opens during the save process.

Save selected images as…
Saves all the images associated with the currently selected proofs in the chosen format.

At the time of writing, imageio.library supports the following formats. When new formats are added to imageio.library they appear automatically in the submenu.

1. Jpeg (JFIF)
2. IFF ILBM
3. PPM

A progress window opens during the save process.

Save selected decoded images as…
Saves the images the proofs represent in a browse window that originate from Base64 or UUencoded files (e.g. from email attachments or NNTP servers) as decoded files. The ASL requester allows selection of a destination directory. The filename of the saved image is the name stored with the encoded data, not the filename of the proof.

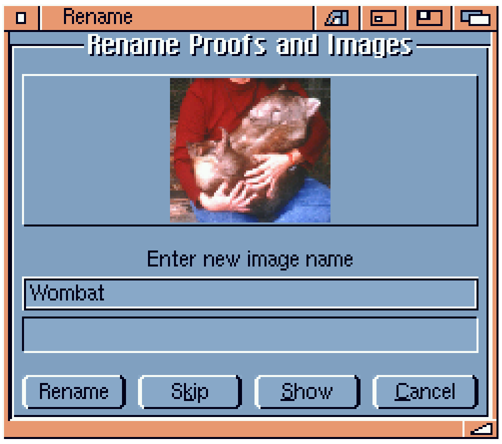
A progress window opens during the save process.

Retarget selected…
Retargets all the currently selected proofs.

Non-linked proofs encountered during retargeting are skipped.

A progress window opens during the retarget process.

Rename selected…
Opens a rename window where the selected proofs can be renamed. The browse window is unable to be used while this requester is open.



Clicking *rename* changes the name of the proof to the entered name. *Skip* discards changes in the string gadget and advances to the next selected proof. *Show*, shows the proof in a show window. *Cancel* closes the window; proofs renamed before cancelling remain renamed.

The progress bar beneath the string gadget indicates how many images have been processed out of the total number of selected proofs.

The displayed proof image has a context menu that contains *show*, to show the proof in a show window.

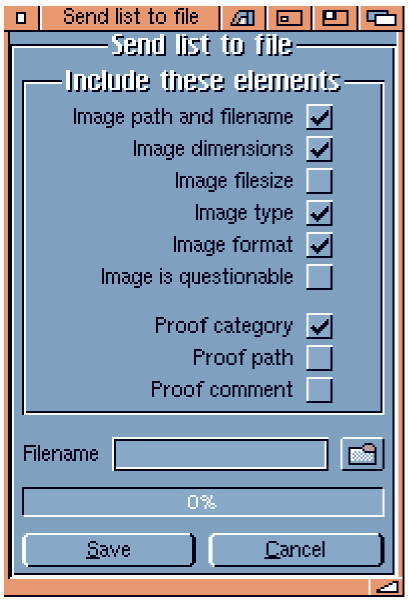
Delete selected…

Deletes all the currently selected proofs.

A progress window opens during the delete process.

Send list to file…

Exports an ASCII list of the browse window contents to a file.



Select the elements to export from the list and enter a filename. Click *save* to store the information, click *cancel* or close the window to abort. A progress indicator shows the process of saving the information.

Export as

Selecting an item from the sub menu exports proofs in the browse window into that format.

Other PhotoFolio modules allow exporting in other formats. See the module's documentation for further information.

Jpeg…

Saves the proof image to disk in jpeg format.

HTML…

If the export HTML module has been installed correctly, saves the proofs in the browse window as a clickable HTML page.

Browse

The *browse* menu contains items that control how proofs are displayed in the browse window. A tick mark (√) against a menu item indicates that the option is enabled.

High quality scaling

Enable high quality scaling in this browse window for newly created proofs.

Hide questionable

Any currently displayed proof that has the hide questionable attribute set is hidden from view in the browse window.

Hide attributes

Hide the attribute selection panel in the browse window allowing more room to display proofs.

No sort

Disables further sorting of proofs in the browse window. If proofs have already been sorted, they will remain in their last sorted order.

Sort alphabetically

Sorts all the currently displayed proofs by filename.

Sort by size

Sorts all the currently displayed proofs by file size.

Sort by type

Sorts all the currently displayed proofs by file type. Filename extenders are only used to determine the file type for sorting prior to loading in a browse window After loading is completed, sorts are performed using the actual image type.

Sort in natural order
Sorts all the currently displayed proofs using a natural sort algorithm. This allows filenames that would normally be separated to appear together. Numbers in filenames are sorted differently than they would be in a regular sort.

Direction
Selecting an item in the submenu sets the sort direction, either *ascending* (up) or *descending* (down). A tick mark (√) against a submenu item indicates that the option is selected.

Ascending
Entries are sorted from lowest to highest. E.g if set to sort alphabetical and ascending, browse entries are displayed sorted a thru z.

Descending
Entries are sorted from highest to lowest. E.g. if set to alphabetical descending, browse entries are displayed z thru a.

Refresh
Removes all the proofs in the browse window (visible or not) and reloads the currently displayed path.

Set browse path…
Set the path being browsed in the current browse window using an ASL requester.

ARexx
The *ARexx* menu contains items that allow the execution of ARexx scripts.

Run ARexx script…
Opens an ASL requester and executes the selected ARexx script.

ARexx scripts
The ARexx scripts selected for browse windows in settings are displayed here. Select the ARexx script from the menu to execute it. Keyboard shortcuts are automatically created for the first 10 scripts (`Left Amiga 0-9`).

Settings
The *settings* menu items configure the operation and appearance of PhotoFolio. Unless otherwise noted, when the settings window is opened from a browse window, the settings being changed are **for this browse window only**.

Settings…
Opens the PhotoFolio program settings window.

MUI…
Opens the standard MUI preferences window.

Load settings…
Opens an ASL requester allowing selection of a previously saved settings file to load.

Save settings as…
Opens an ASL requester allowing selection of a file to save the current program settings as.

Save settings as default
Saves the current program settings in the file `PROGDIR:PhotoFolio.prefs`.

Help
The *help* menu contains items that might assist you with using PhotoFolio.

Online AmigaGuide…
Opens the `PhotoFolio.guide` help file (if installed).

Show tip of the day…
Shows the tip of the day window. If the tip of the day MUI class is not installed, this menu item does not appear.

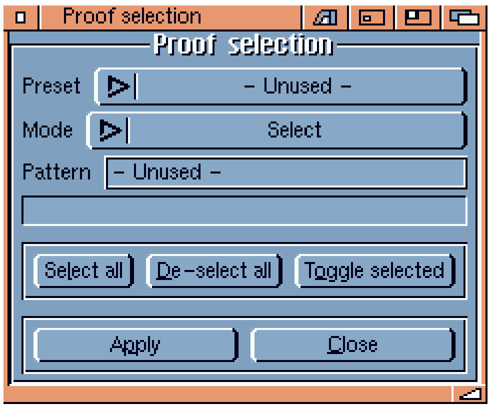
Context menus

Browse window context menus appear when the right mouse button is pressed while the mouse pointer is over blank space in the browse window (i.e. not over a proof).

The context menu items perform the same functions as items in the browse windows proof menu.

Pattern select window

This window allows the selection of proofs by their filename using the AmigaDOS pattern matching system.



The pattern select window applies to the browse window that opened it. Multiple pattern select windows can be open if there are multiple browse windows open. The title bar of the window contains the same path as the browse window it was opened from.

The *preset* cycle gadget displays the 10 default patterns that have been previously set up in the settings window. Selecting one of these copies its contents into the pattern string below it.

If the *mode* is **select**, then applying the match string selects matching proofs. If the *mode* is **deselect**, then applying the match string deselects matching proofs.

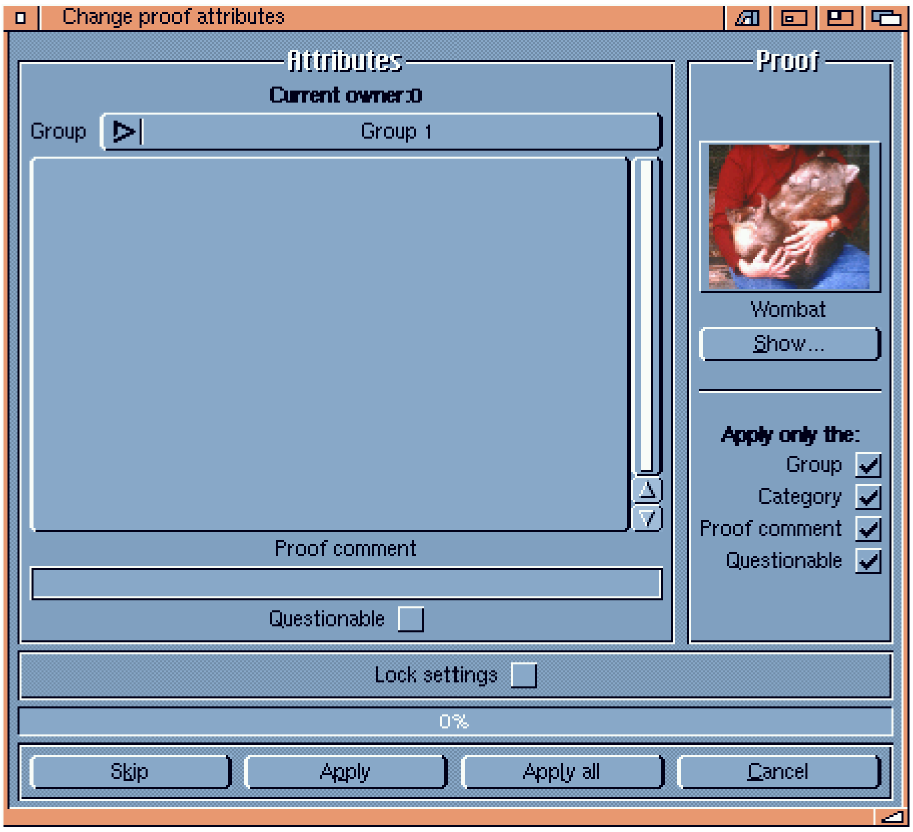
The *pattern* string is the matching string that is applied in the pattern matching routine. Select a preset from the preset cycle gadget above it or type directly into the string gadget.

Select all, *deselect all* and *toggle selected* perform selection operations on the browse window to prepare the window before applying the pattern matching string.

Apply the pattern using the mode to the proofs in the browse window from which this pattern select window was opened. The window doesn't close so the pattern string can be changed and applied again.

Attributes window

The attributes window is used to modify the attributes of one or more proofs.



On the left hand side of the window is the attribute selection area. On the right is the proof display area, which shows the proof whose attributes are being changed, as well as apply boxes.

To change a proof's attributes, select the owner, group, category or categories (while holding the `shift` key), enter a comment and set questionable if required. Next, select which attributes to apply, using the apply boxes below the proof and then use *apply* or *apply all.*

The *current owner* string at the top of the window shows current owner information. The owner can be changed with change owner in the *current owner* or *category list* context menu.

The *group* cycle gadget allows selection of one of the four groups. Group names can be changed or entered using the group cycle gadget's context menu.

The available categories for the selected group appear in the list. There can be a maximum of 22 categories for **each** group. Select all, deselect all, add new categories, save the owner settings, delete a single entry or delete all entries by using the category list menu.

The *proof comment* string gadget adds a comment about the image stored in the proof.

The *questionable* check mark can be used to flag the content of an image. Any images with questionable set can be hidden from a browse window using the hide questionable menu item.

The show gadget or proof context menu allows the image to be viewed using the configured viewer in a show window.

The *apply only the* check marks allow attributes of a proof to be selectively altered. For example, attributes of 200 proofs have already been set, but a general comment has been forgotten. Turn off all the check marks except proof comment. Type in the comment and then click *apply all*. PhotoFolio will only change the selected attributes, leaving all other proof attributes intact.

Lock settings locks the currently set attributes in the window. When **off**, the proof being displayed copies its existing attributes to the attribute selection area. Lock settings **on** prevents this from happening.

Clicking *skip* skips the current proof without changing any attributes.

Apply applies the current attributes to the displayed proof only and *apply to all* applies the current attributes to all the selected proofs.

Clicking *cancel* stops changing attributes and closes the window. Proofs, whose attributes have already been changed, remain changed.

The progress indicator shows how far into the selected proofs the attribute changing process is.

Context menus

A context menu appears when the right mouse button is pressed while the mouse pointer is over an area of the window. The context menu contents vary depending on what the mouse pointer is over when the right mouse button is pressed. Some items appear in more than one menu.

Current owner

Save owner settings
Opens the save owner window allowing changes made to the owner to be saved.

Change owner
Opens a change owner window to select a new owner to load.

Group

Change group names
Opens the group window where group names can be changed.

Categories

New category name
Opens the new category window where categories can be added.

Select all
Selects all of the available categories.

Deselect all
Deselects all of the available categories.

Delete entry
Deletes the category entry under the mouse.

Delete all entries
Deletes all of the available category entries in the list.

Catalogue system

Each proof can be catalogued into 1 of 255 owners, 1 of 4 groups and any one or all of 22 categories. Using these categories in a browse window, proofs can be selectively displayed based on their catalogue information.

Owner
The owner allows the group and categories to be stored in a single file, independent of other group and category configurations. Each proof with attributes set is assigned an owner number that links the proof with the correct owner number.

You could configure the owners so that a friend's proofs can be viewed (using their categories and groups) without modifying your own catalogue configuration.

For example, if you allocated yourself owner 10 and your friend owner 11, you could copy your friend's owner file (`PROGDIR:owners/11.owner`) into your `PROGDIR:owners/` directory and then view your friends proofs using the categories and groups they were catalogued with. Of course, there is nothing stopping you using more than one owner for yourself.

Group
A group is a general heading. There are 4 groups available for every owner. For example, *Textures*, *Renders*, *Watercolours*, *Backgrounds.*

Category
A category is a classification. There are 22 categories for **each** group. For example, *Stone*, *Water*, *Clouds*, *Metal*.

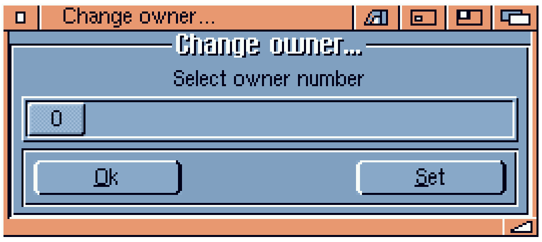
For example, you have allocated yourself owner 10 and you want to catalogue a directory devoted to textures for a 3D rendering package. These image textures may include, a frog, the moon, a marble, a eucalyptus tree, some oranges and a ladies face.

Firstly, you could name a group *Textures*, then create some categories inside that group called, *Animals*, *Stone*, *Nature*, *Food*, *Human*, *Space* and *Backgrounds*.

Now assign the frog image to *Animals* **and** *Nature* **and** *Backgrounds*, the moon image to *Space* **and** *Backgrounds*, the marble to *Stone* **and** *Backgrounds*, the Eucalyptus tree to *Nature*, the oranges to *Food* and the ladies face to *Human*.

Change owner

The change owner window selects an owner to change to.

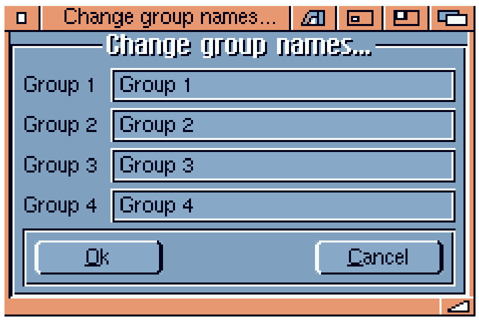


Use the slider to select the owner number to change to and then click *ok* to perform the change. Using *set* instead of *ok* changes the owner but leaves the window open. This feature can be used to look through multiple owners quickly.

To cancel an owner change, close the window.

Change group names

This window is used to change the group names, which appear in the group cycle gadget.

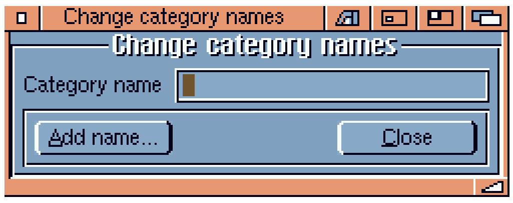


The window contains four string gadgets, one for each group. Enter the group names and click *ok*.

To cancel a group name change, close the window or click *cancel*.

New category

The new category window adds category names to the category list.

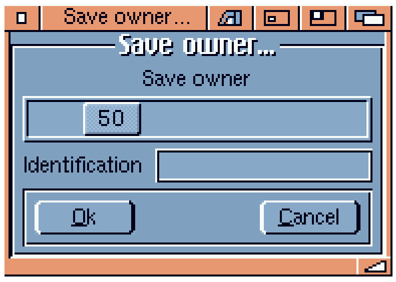


Enter the new category name and click *add name*. The window stays open so more categories can be entered. To finish, close the window or click *close*.

There can be a maximum of 22 categories per group.

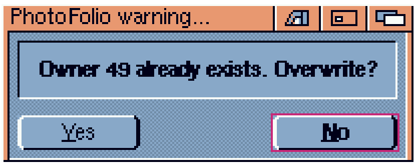
Save owner

The save owner window saves modified owners.



Use the slider to select the owner number to save as, enter a string into the identification gadget, such as your name, then click *ok*.

You are warned if an owner with this number already exists before overwriting an existing owner.

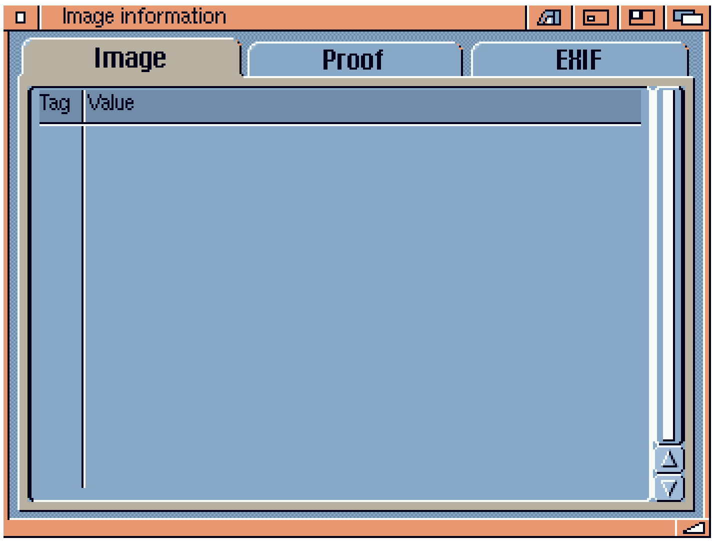

PhotoFolio warning...

Owner 49 already exists. Overwrite?

To cancel a save owner close the window or click *cancel*.

Information window

Shows detailed information about a proof, image, file, directory or device.



Information windows can be opened from:

- Main window - New Information window.
- Browse window - New Information window.
- Proof context menu - Information.
- Show window context menu - Information.

There can be more than one information window open simultaneously. To assist in managing multiple information windows, PhotoFolio uses the concept of *association*. This means the window that opens an information window **owns** that information window. Closing the **owner** or **parent** window closes all attached windows.

For example, if you open an information window from a browse window, that information window belongs to that browse window. Closing the browse window closes the browse window **and** the information window.

To unassociate an information window with its parent window (whoever opened it) use *detach from parent window* in the information window's menu. The information window is then separated from its parent window and operates independently.

Dropping a proof from a browse window onto an information window associates that information window with the browse window the proof is in.

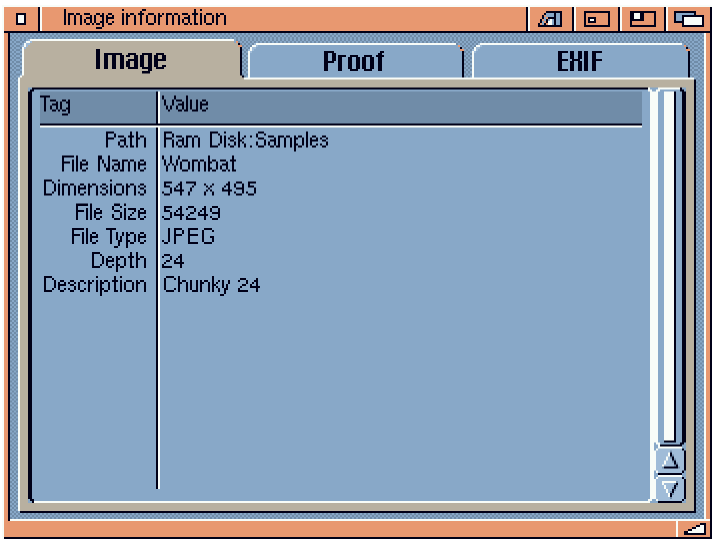
Workbench icons can be dropped onto information windows (including files, directories and devices).

The three tabs in the information window are:

1. Image

   If the file is an image in a recognized format, details about the image are extracted and displayed.

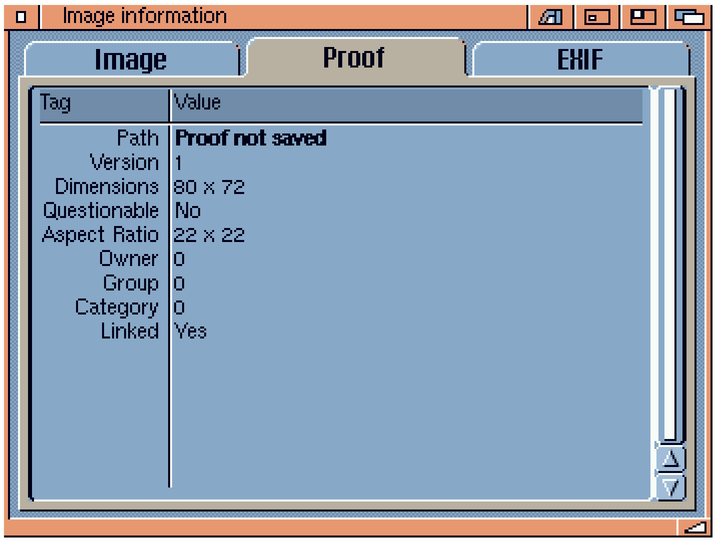
   Other dropped items that are not proofs or images but are files, directories or devices show some information about the dropped item.



Tag	Value
Path	Ram Disk:Samples
File Name	Wombat
Dimensions	547 x 495
File Size	54249
File Type	JPEG
Depth	24
Description	Chunky 24

2. Proof

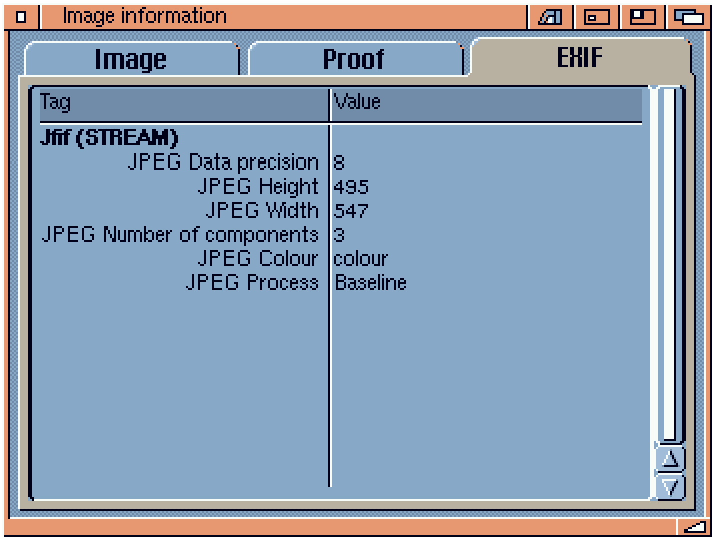
   If the file dropped is a proof, information about the proof and image is extracted and displayed in the *proof* and *image* tabs.



Tag	Value
Path	**Proof not saved**
Version	1
Dimensions	80 x 72
Questionable	No
Aspect Ratio	22 x 22
Owner	0
Group	0
Category	0
Linked	Yes

3. EXIF (requires exif.library)

   Provides additional information from image file formats. The jpeg image format used in many digital cameras stores additional information about the image, such as time, date, focal length etc. If the image contains any recognized information it is displayed here.



Tag	Value
Jfif (STREAM)	
JPEG Data precision	8
JPEG Height	495
JPEG Width	547
JPEG Number of components	3
JPEG Colour	colour
JPEG Process	Baseline

Menus

Information windows can be manipulation using the menu items.

Image information

The *image information* menu contains items that manipulate the information window.

Save information as…
All of the 3 tabs of image information will be saved in ASCII format to the filename entered in the ASL requester.

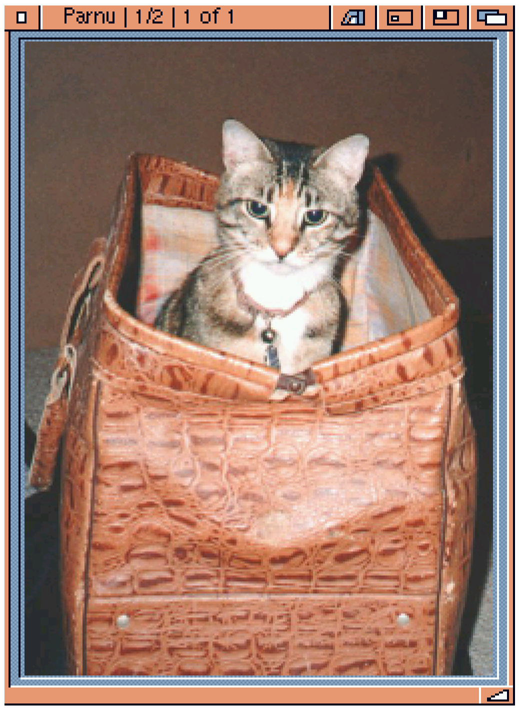
Save information to clipboard
The information displayed in the currently visible window tab will be copied to unit 0 of the system clipboard.

Detach from parent window
Separates this information window from the window that opened it (or it was attached to) preventing it from closing when the parent window closes. Dropping a proof after using this menu item will reattach the information window to the browse window the proof is in.

Close
Closes the information window.

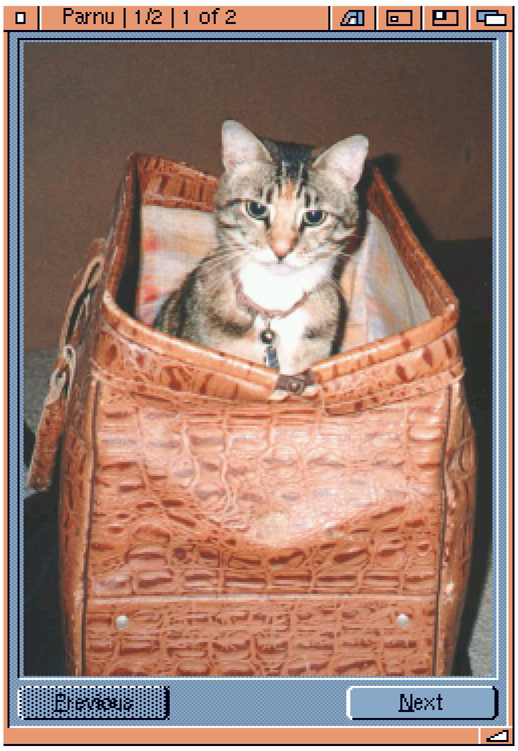
Show window

Images shown using the internal viewer use this window.



The window automatically resizes itself for each image displayed. Scroll bars are used to move around an image that doesn't fit on the screen. If the image can fit entirely on the screen, the scroll bars are automatically removed.

If more than one image is queued to be shown in the window then next and previous gadgets are displayed in the bottom of the window.



Dropping Workbench icons onto the show window attempts to load the dropped files.

Clicking *next* or pressing the `spacebar` shows the next picture in sequence, clicking *previous* or pressing `backspace` shows the previous picture in the sequence.

To find out what image will display if you click the *next* or *previous* gadgets, position the mouse over the *next* or *previous* gadgets and a help bubble appears with the name of the image to be shown.

If the first image in a sequence is displayed, the *Previous* gadget appears ghosted. If the last image is displayed, the *Next* gadget appears ghosted. Pressing the `spacebar` while the last image in sequence is being displayed closes the show window.

Menus

The show window's menu contains the following items that operate on the image visible in the show window.

Show

The *show* menu contains items that operate on the visible image. Menu items appear ghosted when no image has been loaded,.

A tick mark (√) against a *Show* menu item indicates the current scale mode of the show window.

Information...
Opens an information window for the image.

Reload
Reloads the currently displayed image. Useful after processing an image to return to the original image.

Show auto
Display the image so it fits into the available screen space. A tick mark (√) indicates this item is active.

Show full
Display the image full size where one screen pixel equals one image pixel. The shortcut key is `1`. A tick mark (√) indicates this item is active.

Show half
Display the image half of the original image size. The shortcut key is `2`. A tick mark (√) indicates this item is active.

Show quarter
Display the image one quarter of the original image size. The shortcut key is `4`. A tick mark (√) indicates this item is active.

Show eighth
Display the image one eighth of the original image size. The shortcut key is `8`. A tick mark (√) indicates this item is active.

Save original size as
Selecting an item in the submenu saves the image as a new file using the *original image file* as the source for the save. The original image will be loaded **full** size into memory and then saved to disk under the new name. If an image has been processed and this command is used, any processes applied will **not** be saved.

Save loaded size as
Selecting an item in the submenu saves the image as a new file using the *loaded (or visible) image* as the source for the save. The visible image will be saved to disk under the new name. This allows an image to be reduced in size or processed and then saved.

Saving an image will remove any extended image information contained in the original image (such as EXIF from digital cameras).

Process
Image process operators can be applied to the image currently being viewed in the show window. It is important to realize that operators are applied to the **visible** image. Pressing next or previous discards the current image as well as any operators applied to it. If the image has been loaded at a smaller size than the original, the operator is applied to the

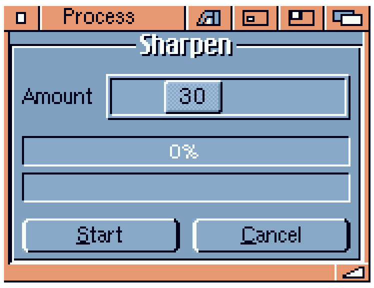
smaller (visible) image, not the original size. Reload can be used to return to the original image before operators were applied.

The operators are provided by imageprocess.library and appear as submenu items under this menu and the process context menu. Additional operators become available as features are added to imageprocess.library. At the time of writing, the following operators are available:

- Rotate +90
- Rotate –90
- Rotate 180
- Flip X
- Flip Y
- Auto gamma
- Gamma
- Sharpen
- Negative

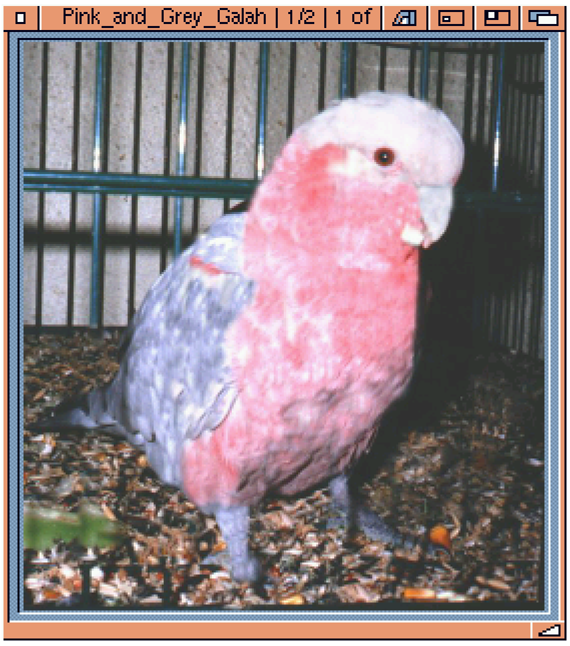
Selecting a process performs that process on the image visible in the show window. A progress window opens during the application of the process operator.

If the operator requires additional parameters (e.g. the sharpen operator requires the amount of sharpening to apply), a requester opens prompting for the required information. Clicking *start* then applies the operator.

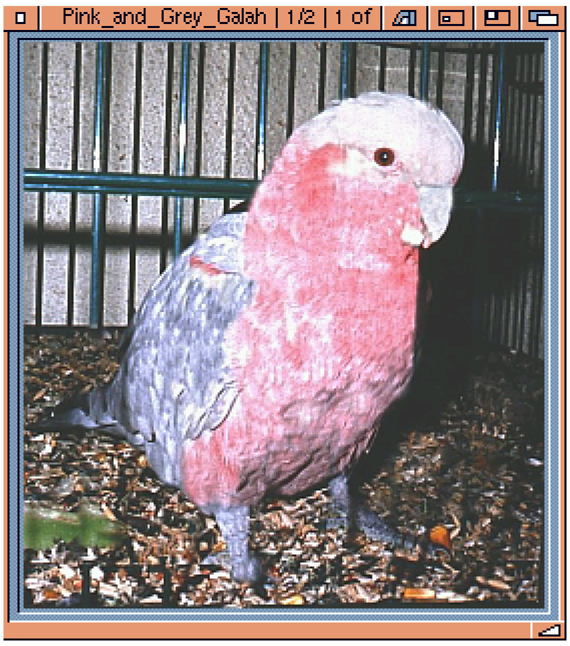


Cancelling an in progress operator can sometimes leave the visible image partly processed. The reload command can be used to return to the original image before processing.

Here is a before and after view of the sharpen operator being applied to an image.



Before



After

Close
Closes the show window.

Context menu

The context menu appears when the right mouse button is pressed with the mouse pointer over the image. The title of the menu consists of the filename of the image, the number of the image in the sequence in brackets `[]` and an indication of the size of the image being displayed:

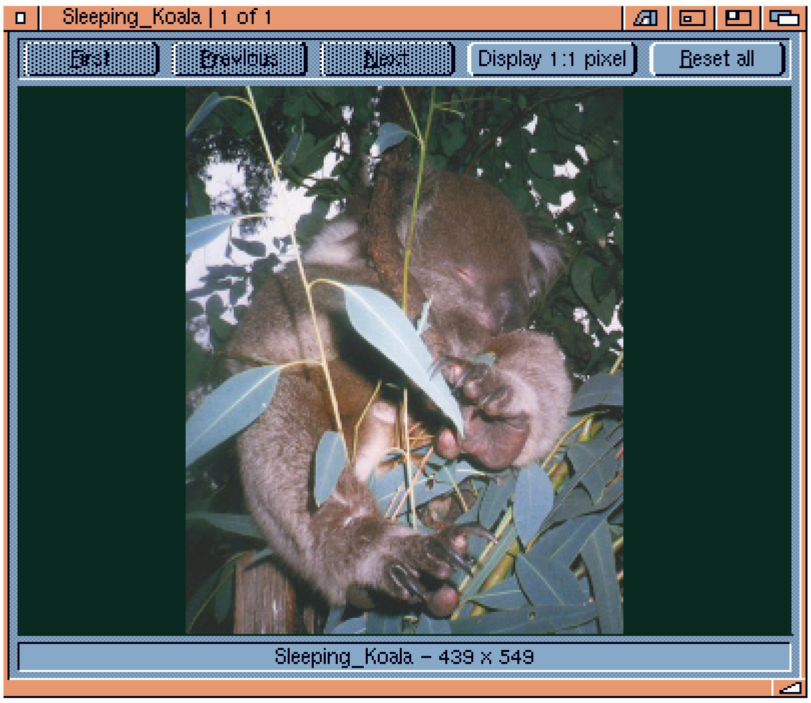
- `1/1` = Full size
- `1/2` = Half size
- `1/4` = Quarter size
- `1/8` = Eighth size

A tick mark (√) against a *Show* menu item indicates the current scale mode of the show window.

The menu items in the context menu duplicate the functionality of the regular window menu.

Mysticview window

PhotoFolio internally supports MystiView.library to view images.



The default MysticView setup can be configured in the settings window.

With show picture info enabled, picture information is displayed underneath the image.

With show buttons enabled, the top of the window contains a button bar with controls for navigating multiple images.

If more than one image was selected to be shown, the *first* gadget moves to the first image in the list, the *previous* gadget to the previous image in the list and *next* to the next image in the list.

Display 1:1 pixel controls how the image is scaled to fit the available space.

Reset all resets the move, rotation and zoom values to their defaults.

Menus

Some of the menu items perform the same functions as their equivalent controls.

Picture

The *picture* menu contains items that operate on the image being shown.

Next

If more than one image was selected to be shown, moves to the next image in the list.

Previous

If more than one image was selected to be shown, moves to the previous image in the list.

First
If more than one image was selected to be shown, moves to the first image in the list.

Move

Left
Scroll the image to the left.

Right
Scroll the image to the right.

Up
Scroll the image up.

Down
Scroll the image down.

Centre
Position the image so that the centre of the image is in the centre of the window if possible.

Zoom

In
Zoom in on the image. Once a one to one pixel ratio is achieved, zooming further increases the pixel size so that the image appears *blocky*.

Out
Zoom out on the image.

Reset
Reset the zoom to one to one pixel ratio (not zoomed).

Rotate

Left
Rotate the image left 30 degrees.

Right
Rotate the image right 30 degrees.

Reset
Reset the rotation value to 0 degrees.

Reset all
Resets the move, rotation and zoom values to their default state.

Options

The *options* menu contains items that allow MysticView settings to be adjusted. The default settings can be configured in the settings window.

Scale with aspect

If enabled, indicates that the image is to be scaled to fit the window size.

Display 1:1 pixel

Indicates that the image is displayed actual size.

Show buttons

Toggles the button bar in the top of the MysticView window on and off.

Show picture info

Toggles the picture information bar in the bottom of the MysticView window on and off.

Show arrows

When enabled, draws arrows at the picture's borders.

Show picture in picture

When enabled, displays a thumbnail of the image in the top left hand corner of the image.

Show cursor

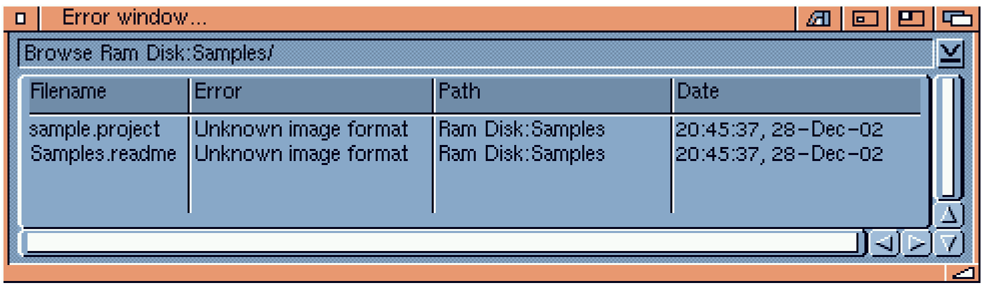
When enabled, activates a crosshair on the picture, indicating the current cursor position.

Mouse drag picture

Enables scrolling of the image, by clicking and dragging the mouse.

Error window

The error window displays information about problems that occur while processing files, images and proofs.


Error window...

Browse Ram Disk:Samples/

Filename	Error	Path	Date
sample.project	Unknown image format	Ram Disk:Samples	20:45:37, 28-Dec-02
Samples.readme	Unknown image format	Ram Disk:Samples	20:45:37, 28-Dec-02

Any errors that occur while PhotoFolio is running are reported to this window. It contains a list of the files which had errors, a description of the error, the path the files came from and the date/time the error occurred.

Errors are reported to this window regardless of whether the window is visible or not. The error window can be opened and closed from any window that logs errors in the error window.

The error window is global to all windows of PhotoFolio.

Errors are logged on a *which window created the error* basis. For example, if you open a browse window, files that aren't images are logged as errors. They are listed under the browse windows title in the error window.

The popup gadget in the error window allows selection of which *window* of errors to look at. All errors can be shown in one list or errors that are unattached to any window can be viewed separately.

Browse windows have a show errors menu item to view the errors that were generated from that browse window.

Normally, closing a browse or other window that has generated errors removes those errors from the error window. Selected errors can be manually *detached* from a window's title so that they don't disappear when the window is closed. *Remove window errors* removes all errors associated with a window.

Errors generated during a proof make are not removed when the make window closes. Instead the errors remain logged under the date and time the make was started.

Menus

Errors (which are essentially just unrecognised/corrupt image files) can be manipulated as though they were files using the windows menu and context menu.

Project

The *project* menu contains items that manipulate the entire error list.

Hide error window

Hides the error window from the screen. Errors are still logged even if the error window is not visible. The window can be set to automatically open whenever an error is logged.

Remove window errors

For windows that generate errors and leave them behind when they are gone (e.g. the make window), this allows all of the errors attached to that window title to be removed.

Save error list...

Opens an ASL requester prompting for a filename to save the entire error list into as an ASCII file.

Edit

The *edit* menu contains items that allow editing of the contents of the error list.

Select all
Selects all the visible items in the error list.

Deselect all
Deselects all the visible items in the error list.

Toggle selected
Toggles the select state of all the visible items in the error list. Selected items are deselected and vice versa.

Selected
The *selected* menu contains items that operate on all of the selected items in the error list.

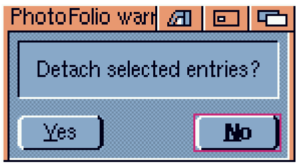
Copy...
Opens an ASL requester prompting for a destination directory to copy the file the error represents to.

Move...
Opens an ASL requester prompting for a destination directory to move the file the error represents to.

Save...
Opens an ASL requester prompting for a filename to save the selected items into as an ASCII file.

Detach...
Separates the selected items from their window title. This prevents them from being lost when the window that created them closes. They appear under the *unattached* window title.

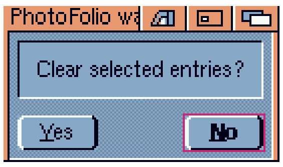
A confirmation requester appears before the selected errors are detached.


PhotoFolio warn

Detach selected entries?

Clear...
Removes the selected entries from the error list.

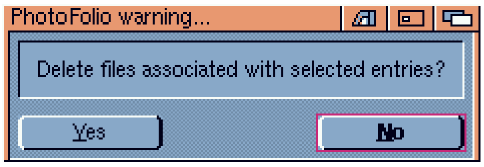
A confirmation requester appears before the selected errors are cleared.


PhotoFolio wa

Clear selected entries?

Delete...
Deletes the files associated with the selected items in the error list.

A confirmation requester appears before the selected errors are deleted.


PhotoFolio warning...

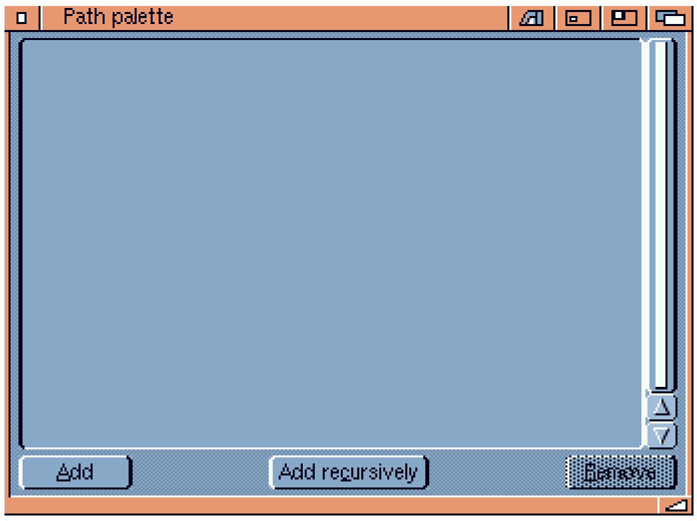
Delete files associated with selected entries?

Context menu

Items in the error list context menu perform the same functions as those in the selected menu but they operate on the right clicked item and do not ask for confirmation before executing.

Path palette window

The path palette window is a storage area for paths.



Paths can be added to this window by clicking *Add* or *Add recursively* and selecting a path in the ASL requester. Workbench directories and devices can be dropped onto the window to add them to the list.

The contents of the path palette window are remembered when PhotoFolio is quit and restored when PhotoFolio is restarted. The window can also be set to open automatically when PhotoFolio is run.

Individual paths can be removed by selecting an item and clicking *Remove*.

Path palette entries can be dragged and dropped to:

1. Main window (inserts the path into the path list)
2. Browse windows (loads and displays the path)
3. Show windows (shows all images in the path)
4. Information windows (shows information about the path)

5. Directory view windows (loads the path)

Double clicking a path palette entry opens the entry into a new browse window.

Menus

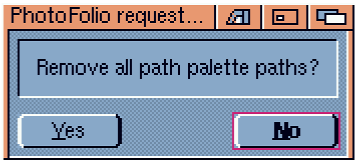
Path palette entries can be manipulated from the window's menu.

Path palette

The *path palette* menu contains items that manipulate the path palette entries.

Remove all

Removes all the entries in the path palette window after prompting for confirmation.



PhotoFolio request...

Remove all path palette paths?

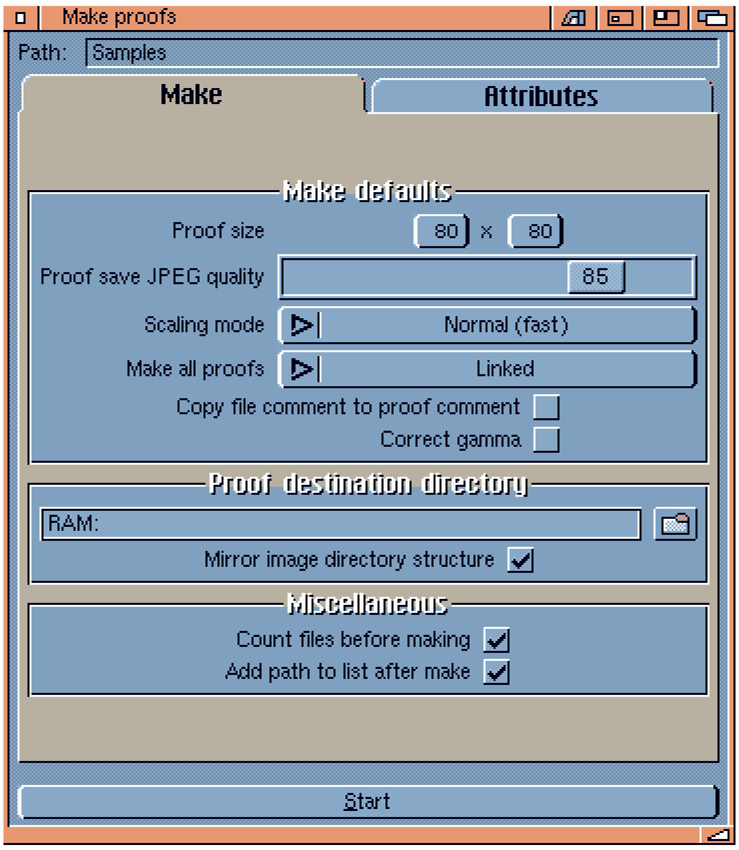
Close

Closes the path palette window.

Make window

The make window is used to create proofs for directories of images without displaying them or using a browse window. This has the advantage of being quicker and using less memory.

The *make* group tab contains settings for the creation of the proofs. The default settings for make are copied from the settings window.



Make proofs

Path: Samples

Make defaults

Proof size 80 x 80

Proof save JPEG quality 85

Scaling mode Normal (fast)

Make all proofs Linked

Copy file comment to proof comment ☐

Correct gamma ☐

Proof destination directory

RAM:

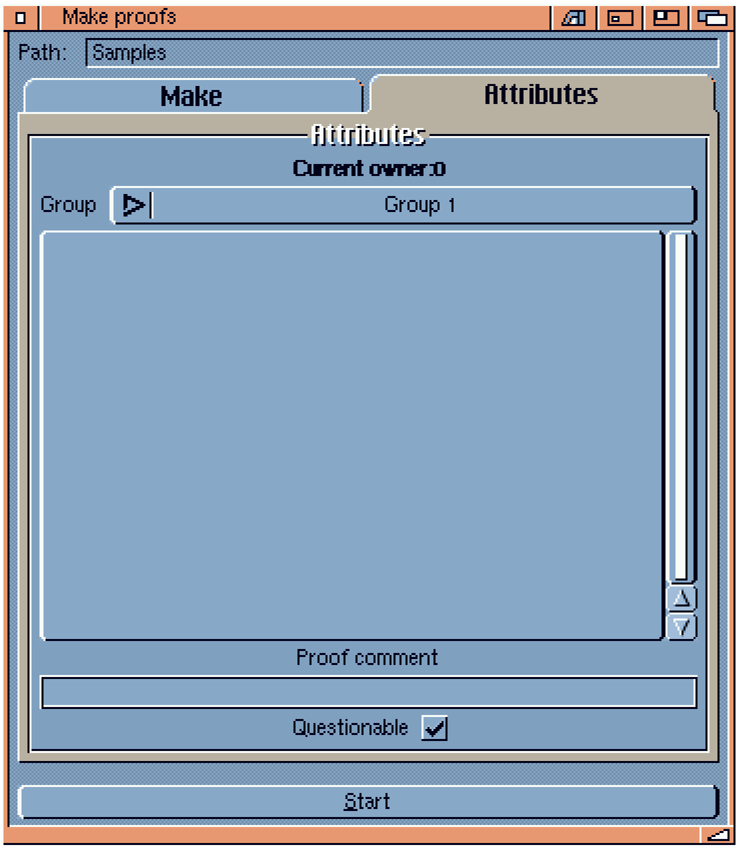
Mirror image directory structure ✓

Miscellaneous

Count files before making ✓

Add path to list after make ✓

The *attributes* group tab contains the attributes that are given to the created proofs.



As a path in the path list is made, PhotoFolio highlights it to make it easy to see which paths have and haven't been made.

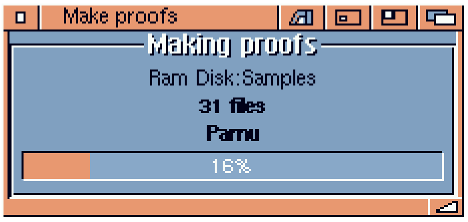
The *start* gadget at the bottom of the window starts the make process. Closing the window cancels the make.

The default make settings can be configured in the settings window.

The error window can be opened from the *error window* menu item in the make window.

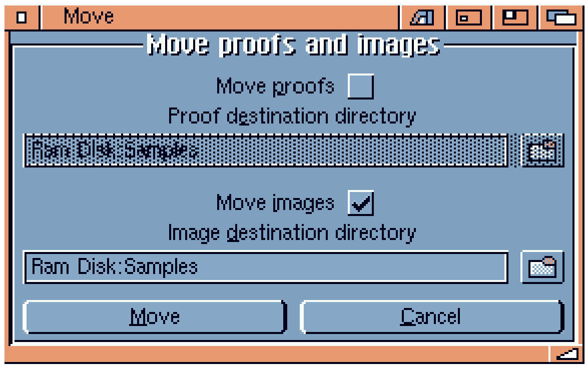
If add to path after make is enabled, the proof directory is added to the path list **Proofs** folders after the path has been processed.

During the make process, a progress window opens. The menu item *error window* shows errors from this make window.



Move window

Allows entry of the destination paths and selection of what to move (proofs and/or images). Paths entered into the window are remembered and are displayed the next time the move window is opened.



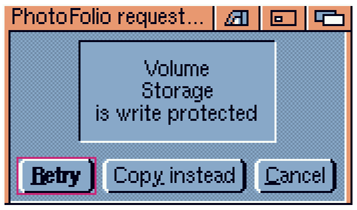
If *move proofs* is enabled, the proof path string gadget becomes active allowing selection of a destination directory for proofs. If left disabled, proofs are **not** moved.

If *move images* is enabled, the image path string gadget becomes active allowing selection of a destination directory for images. The date stamp and file comment of the source file are copied to the destination file. If left disabled, images are **not** moved.

Clicking the *move* gadget starts the move process. The browse window cannot be used during a move and it is not updated visually until after the move is finished.

A progress window opens during the move process.

Move checks if the source is write protected and if so asks the user to *retry*, *copy instead* or *cancel*.



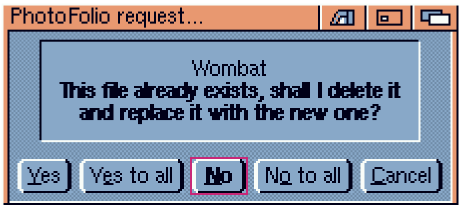
Volume
Storage
is write protected

There are several rules when moving proofs and images:

- The proof destination path **cannot** match the current proof path.
- The image destination path **cannot** match the current image path.
- If the proofs are linked, the proof destination path **cannot** match the image destination path.
- If moving images, an image destination path **must** be supplied.
- A proof destination path is optional depending on whether the proofs are linked or non-linked.

If problems are encountered as proofs and images are moved the following measures are taken:

- If the proofs are non-linked the proof destination path is ignored and the proofs stay with the image.
- When moving proofs only and PhotoFolio encounters a non-linked proof, it is **not** moved and the error window opens displaying the error.
- When moving proofs only, and PhotoFolio encounters an unsaved proof it is **not** moved and the error window opens displaying the error.
- When moving images only, the associated proof is automatically retargeted to point to the new location.
- If a linked proof of the same name already exists in the destination path, the proof is **not** moved and the error window opens displaying the error.
- If an image of the same name already exists in the destination path, a requester opens asking to overwrite the existing image. If you answer *yes* or *yes to all*, the linked proof associated with that image is also overwritten (if one exists).

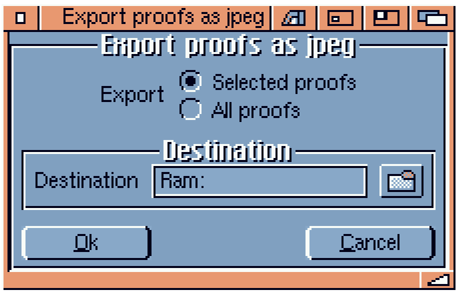

PhotoFolio request...

Wombat
This file already exists, shall I delete it and replace it with the new one?

- If a non-linked proof of the same name already exists in the destination path, the proof is moved and renamed.
- If a linked proof is moved to a new path, it is removed from the current browse window.

Export as jpeg

The *export as jpeg* window allows the proof sized images in a browse window to be exported as jpeg files.


Export proofs as jpeg

Export proofs as jpeg

Export (● Selected proofs) ○ All proofs

Destination

Destination: Ram:

Default settings for export as jpeg are configured in the settings window.

If any proofs are selected in the browse window, the default *export* mode is *selected*. When no proofs are selected, the default *export* mode is *all* which exports all proofs in the browse window.

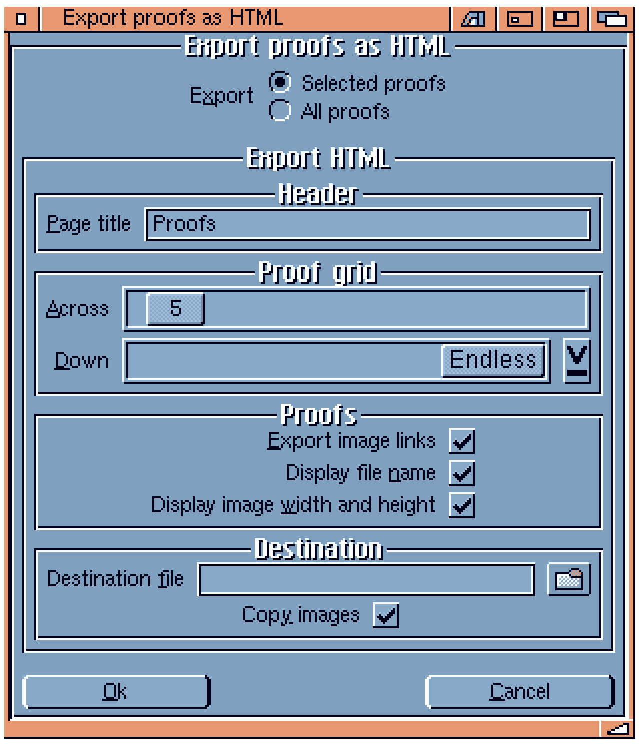
Select a destination directory and click *ok* to start exporting.

No files are overwritten by this process. If a file of the same name already exists in the destiantion directory then the export of that file is skipped.

Export as HTML

The *export as HTML* module allows proofs in a browse window to be exported as HTML pages with optional links to copies of the original images.

The module adds *HTML* to the export proofs as menu in browse windows and an export HTML settings group to the settings window.


Export proofs as HTML

Export proofs as HTML

Export [✓ Selected proofs] All proofs

Export HTML

Header

Page title: Proofs

Proof grid

Across: 5

Down: Endless

Proofs

Export image links ✓

Display file name ✓

Display image width and height ✓

Destination

Destination file:

Copy images ✓

Default settings for export HTML are configured in the settings window.

Even though the settings for this module can be configured from any settings window, they are global settings and changing the module's settings in one window affects **all** other windows.

If any proofs are selected in the browse window, the default *export what* mode is *selected*. When no proofs are selected, the default *export what* mode is *all* which exports all proofs in the browse window.

The module's default settings are displayed when the export HTML window is opened.

Clicking the *export* gadget starts the process. The current settings are also copied to the global module settings as the new default settings. Clicking cancel loses changes made to the local settings in the export window

Once complete, the resulting directory structure is as follows (in this example we assume the destination file is `ram:steeple.html`, there are 30 proofs and the proof grid is 5 x 5):

- An entry HTML file `steeple.html` is created containing 25 proofs.
- A second HTML file `steeple2.html` is created containing 5 proofs.
- A `ram:proofs` directory is created containing the proof images.
- A `ram:images` directory is created. This directory is only populated with images if enabled in settings.

Restrictions

- Export HTML does not allow the link of an exported proof's image to point into another filesystem (all links are relative). To allow the viewing of images in the exported HTML file, the exported images are copied to the destination directory into the `image` subdirectory.
- There must be enough disk space to accomodate all necessary files otherwise the export will fail.

Menus

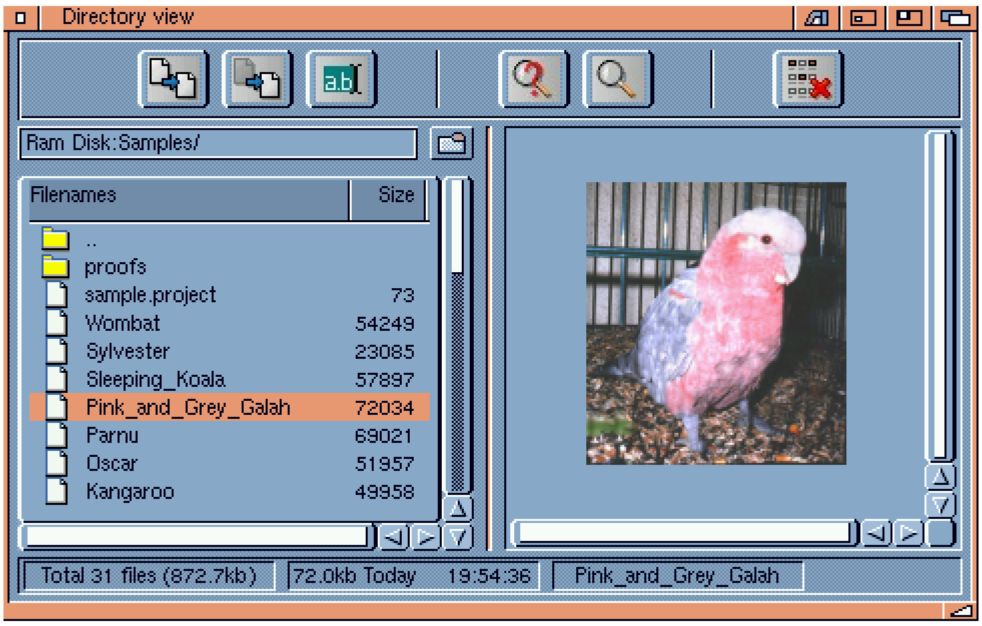
Version…
Displays version number information for the export as HTML module.

Close
Closes the export as HTML window.

Directory view

The *directory view* module allows browsing of files using a directory list with a preview pane.

When installed correctly, the module adds *new directory view* to the window menu in the main window and *open in directory view* to the path list context menu.



Ram Disk:Samples/

Filenames	Size
..	
proofs	
sample.project	73
Wombat	54249
Sylvester	23085
Sleeping_Koala	57897
Pink_and_Grey_Galah	72034
Parnu	69021
Oscar	51957
Kangaroo	49958

Total 31 files (872.7kb) | 72.0kb Today 19:54:36 | Pink_and_Grey_Galah

The toolbar at the top of the window has (from left to right) copy, move, rename, information, show and delete gadgets. These are the same visually as for browse windows. If you are unsure of what each gadget in the toolbar does, position the mouse over the gadget and a help bubble appears.

The displayed path is shown in the string gadget above the file list and the associated popup gadget opens an ASL requester allowing selection of a different path to display.

Selecting an entry shows it in the preview pane (if it is an image). Another entry can be selected at any time and loading is aborted and started on the new selected entry.

The context menu in the preview pane allows selection of the preview size.

Double clicking an entry in the file list:

- If an image, shows it in a show window.
- The **..** entry (at the top of the file list) opens the parent directory into the file list.
- If a directory, opens that directory into the file list.

Files (images or not) can be copied, moved, renamed and deleted using the toolbar or context menu on the entries.

Image information can be obtained as well as showing the selected image in a show window.

Menus

Show all images in path…

Shows all images in the displayed path in a show window.

Version…

Displays version number information for the directory view module.

Close
Closes the directory view window.

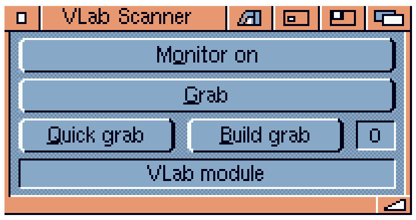
Context menus

The context menu contans items that duplicate the toolbar gadgets functions.

Vlab window

The VLab window is only available if a VLab card and the associated VLab driver libraries and classes are installed correctly.

This window allows image grabbing with a VLab card. Images can be grabbed and sent to a browse window.



To display an input monitor window, click the *monitor on* gadget. The speed the monitor updates depends on how busy your system is and how fast your CPU is.

Clicking *grab*, grabs an image from the VLab card and sends it to the browse window. The speed of this process is affected by VLab.mcc class settings. For the fastest possible grab, turn *VMem* off in VLab preferences in MUI settings. You will need to have enough available memory to do this. The conversion process from the VLab format increases the time taken to display the grabbed image as a proof.

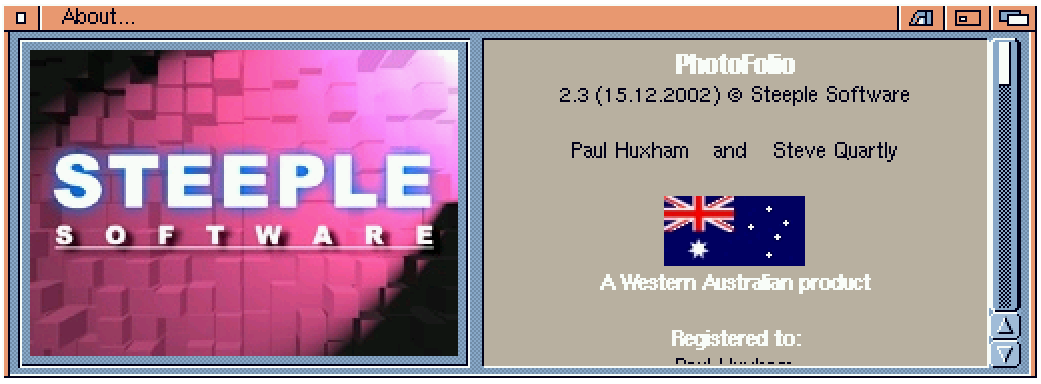
Once the image is sent to the browse window, it can be shown (scaling is disabled), deleted or saved. If save proof or save selected proofs is selected, PhotoFolio detects that the image is a VLab grab and saves the image before it saves the proof. Once saved, the grab becomes a normal image and its proof operates like all others. The image is saved in the jpeg format.

Due to the slow speed of the grab function you can use *quick grab*. When quick grab is used, an image is grabbed from the VLab card, however no conversions are performed and the image is not sent to the browse window immediately. The image is stored internally and when you have finished grabbing, use build grab to send the grabbed images to the browse window. If *VMem* is turned on in VLab.mcc class MUI settings, this uses very little memory. A counter to the right displays the number of grabs currently stored.

Build grab sends stored grabs (grabbed with quick grab) to the browse window. This is the slow part of the process, so if a lot of images have been quick grabbed, have a coffee! Each time an image is built the counter on the right decrements.

About window

This window shows general information about PhotoFolio.


About...
STEEPLE
SOFTWARE
PhotoFolio
2.3 (15.12.2002) © Steeple Software
Paul Huxham and Steve Quartly
A Western Australian product
Registered to:

Most images and underlined text can be clicked to jump to an internet homepage via url.library (if installed).

Settings window

The settings window is used to set the user preferences for PhotoFolio.

Settings are grouped together in common blocks in a list down the left hand side to make finding and adjusting settings easier.

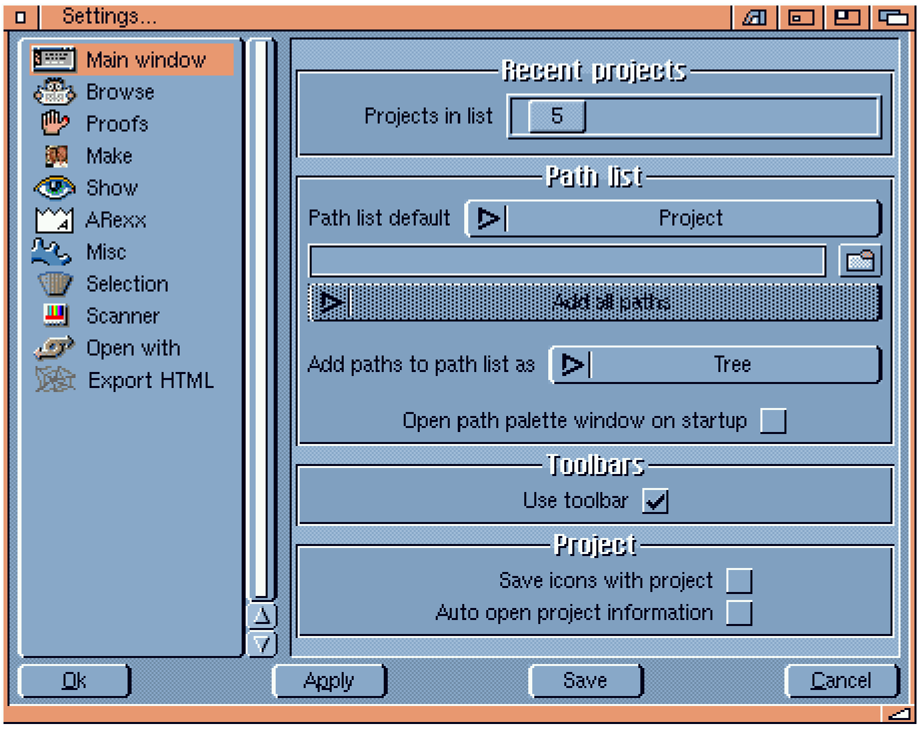
Clicking *ok* accepts all the settings' changes and closes the window. Any visual updates that are needed are then made. *Apply* performs a similar function to *ok* except that the settings window remains open. *Save* accepts all the settings' changes and saves the current settings as the default settings.

Clicking *cancel* abandons any changes (not yet applied) and closes the window.

The settings window shows settings that are applicable for the window from which the settings window was opened. For example, main window settings are not shown if the settings window is opened from a browse window.

Main Window

Configures the main window. This settings group is only visible if the settings window is opened from the main window.



Projects in list 5

Path list default Project

Add paths to path list as Tree

Open path palette window on startup ☐

Use toolbar ✓

Save icons with project ☐

Auto open project information ☐

Projects in list

Setting this to 0 disables the storing of recent projects. If the new number is lower than the previous settings, then the list is cropped to the new size. The clear recent project list item in the settings menu clears all stored entries.

Path list default

The cycle gadget offers two choices:

1. PhotoFolio project
2. Path

Either a path **or** a PhotoFolio project can be automatically loaded on program startup.

The string gadget should contain either a valid path or valid PhotoFolio project file.

Add one/all/recursively

Determines the way the default path is added to the path list. It is only used if the default path is an image path.

Add paths to path list as

Indicates that paths added to the path list should be added in *tree* or *flat* mode. For example adding `Ram:pictures/nature` in *tree* mode creates a `Ram:` folder, inside that a `pictures` folder and inside that a `nature` folder. If added in *flat* mode, a single entry `Ram:pictures/nature` is created in the path list.

Open path palette window on startup
When enabled, automatically opens the path palette window when PhotoFolio is run.

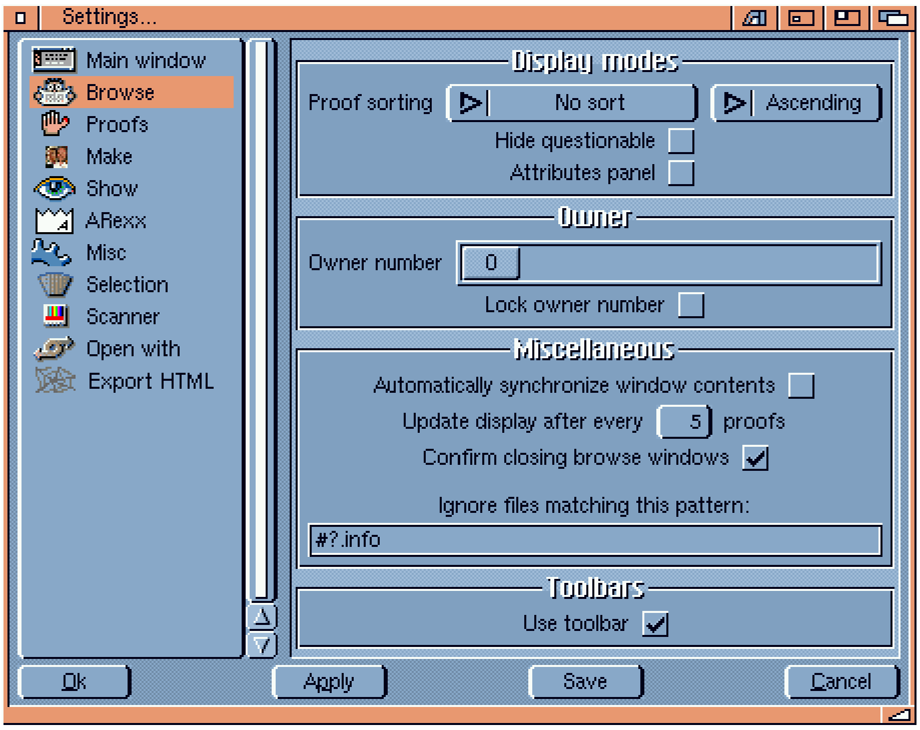
Use toolbar
When enabled, the toolbar appears in the main window.

Save icons with projects
When enabled, icons are saved along with project files.

Auto open project information
When enabled, the project information window opens automatically whenever a project is loaded.

Browse

Configures browse windows.



Display modes

Proof sorting: No sort | Ascending

Hide questionable [ ]

Attributes panel [ ]

Owner

Owner number: 0

Lock owner number [ ]

Miscellaneous

Automatically synchronize window contents [ ]

Update display after every 5 proofs

Confirm closing browse windows [✓]

Ignore files matching this pattern:

#?.info

Toolbars

Use toolbar [✓]

Proof sorting
Proofs displayed in a browse window are sorted in one of 4 ways:

1. Sort alphabetically
2. Sort by size

3. Sort by type
4. No Sort

This can also be changed from the browse window's menu.

Direction

Sets the sort direction, either *ascending* (up) or *descending* (down). E.g. If set to sort alphabetical and ascending, browse entries are displayed sorted a thru z. When set to alphabetical descending, browse entries are displayed z thru a.

Hide questionable

When enabled any proof that has the questionable attribute set is hidden from view in the browse window.

This can also be changed from the browse window's menu.

Attributes panel

Toggles the attribute selection area in the browse window on and off.

This can also be changed from the browse window's menu.

Owner number

Sets the default owner for the catalogue system.

Lock owner number

If enabled, when a change attributes window is opened, lock settings is also enabled. This forces proofs to be viewed with this owner number rather than with the owner number they were catalogued with.

Automatically synchronize window contents

When enabled, automatically runs delete orphaned proofs and create proofs for new images on the path **before** it is loaded into the browse window. This affects every path displayed in the browse window; opened from the main window, dragged and dropped, from set browse path and from ARexx commands.

This can be useful if you are continually examining a directory whose contents change often such as a digital camera directory. The proofs can then be on disk for faster loading, but new images and proofs for images that have been removed are taken care of automatically.

Update display after every

Every time a proof is added to the browse window, MUI updates the entire browse window (even the non visible parts).

Increasing this value speeds up display refreshing. It can be set between 1 and 300. This indicates that the display will update every *x* proofs. e.g. if you are loading a directory with 30 proofs, you will not see anything until the first *x* are loaded, and then again after the next *x* are loaded etc.

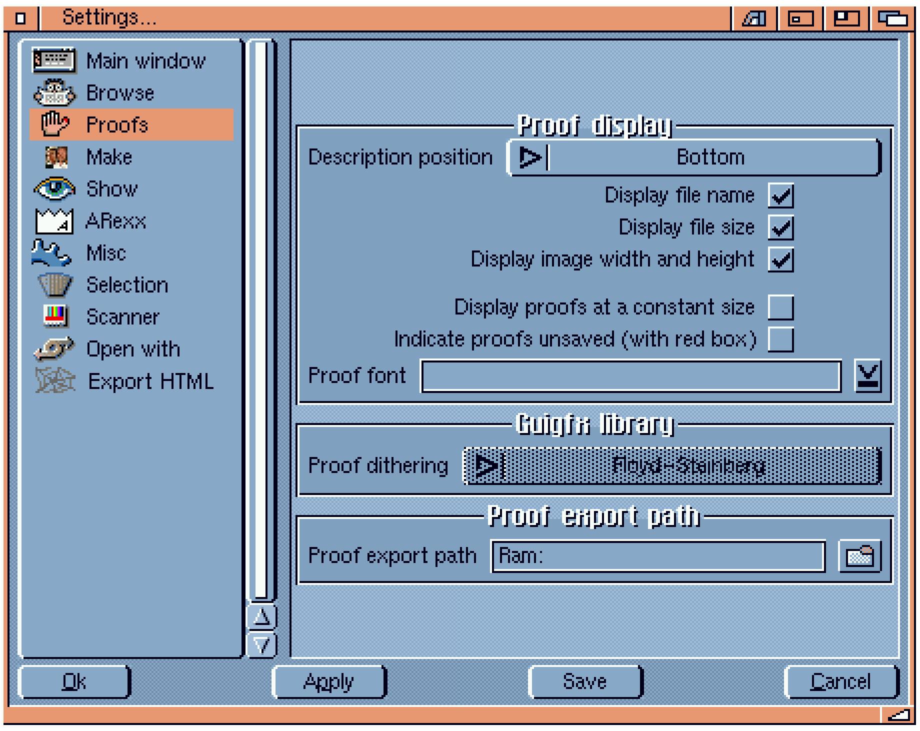
Confirm closing browse windows
When enabled, a confirmation requester appears whenever you attempt to close a browse window.

Ignore files matching this pattern
The string gadget specifies files (using the AmigaDOS pattern matching system) that should be ignored when scanning files to load into a browse window. For example to ignore `.info` files, enter `#?.info`.

Use toolbar
When enabled, the toolbar appears in the browse window.

Proofs

Change the way the proofs appear in a browse window.



Proof display

Description position: Bottom

Display file name ✓

Display file size ✓

Display image width and height ✓

Display proofs at a constant size ☐

Indicate proofs unsaved (with red box) ☐

Proof font

Guigfx library

Proof dithering: Floyd-Steinberg

Proof export path

Proof export path: Ram:

Description position
Sets where and how the text describing the proof appears:

1. *No text* – Turns text off altogether.
2. *Text left*
3. *Text right*
4. *Text top*
5. *Text bottom*
6. *Bubble* - no text is visible, however if the mouse is positioned over a proof, the description appears in a bubble

Proof text attributes
By enabling or disabling these items the lines of text displayed with proofs can be selected.

- Display file name
- Display file size
- Display image width and height

Display proofs at a constant size
When enabled, PhotoFolio draws proofs at the size currently set by the proof size setting in make window settings. This ensures that all proofs are aligned evenly in the browse window, however it takes up unnecessary space and fewer proofs can be displayed in the same area than if this was turned off.

If the proof size settings are smaller than the proof being loaded, the proof is drawn at the proof's actual size.

Indicate proofs unsaved (with red box)
When enabled, PhotoFolio draws a small red box in the bottom right corner of any unsaved proof.

Proof font
The font used for the text displayed beneath a proof. The pop up gadget opens an ASL requester to select a font, or enter a font name and size directly into the string gadget. For example, `topaz.font/11`.

Proof dithering

This is only available if guigfx.library is being used.

There are 4 dithering modes:

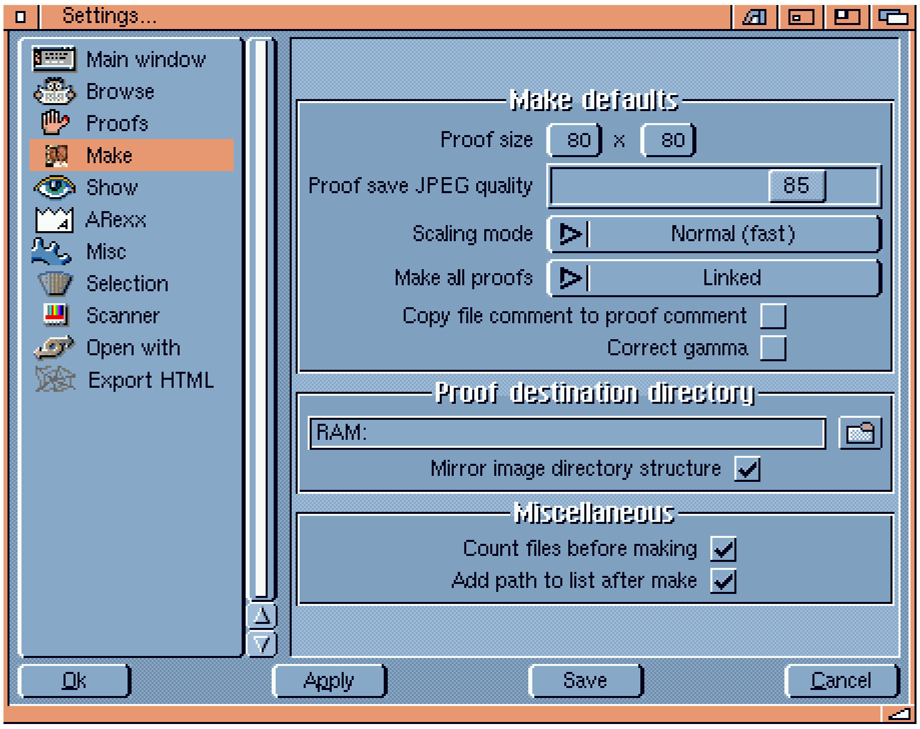
1. None
2. Floyd-Steinberg
3. Random
4. EDD

This setting is for displaying proofs only. If you find your system is slow, set this to *none* and the graphics rendering will be faster, however the quality of the displayed proofs may suffer.

Proof export path
The default path used when exporting proofs.

Make

Determines how proofs are created for browse windows and the make window.



Make defaults

Proof size 80 x 80

Proof save JPEG quality 85

Scaling mode Normal (fast)

Make all proofs Linked

Copy file comment to proof comment ☐

Correct gamma ☐

Proof destination directory

RAM:

Mirror image directory structure ✓

Miscellaneous

Count files before making ✓

Add path to list after make ✓

Proof size
The width and height of the generated proofs.

Proof save JPEG quality
Each proof image is saved internally as a jpeg image using jpeg.library. This setting determines the quality of the jpeg proof image. Qualities range from 1 to 100 with 100 being the best quality. A good balance between size and quality is 85.

Scaling mode
There are 3 different scaling routines PhotoFolio can use to scale the original image to the proof size.

1. *Low quality* scaling (fast)
2. *High quality* scaling (slow)
3. *Cubic interpolation* scaling (slowest)

Low quality scaling is still good, and is *much* faster than the other two methods.

This can also be changed from the browse window’s menu.

Make all proofs
PhotoFolio can store proofs in two different ways, linked and non-linked:

1. A **linked proof** is stored in a totally different place to that of the original image. It is stored in the proof destination directory, in a directory structure that can mirror the original. Proofs can be moved between directories later without the connection to the original images being lost.

2. A **non-linked** proof is stored with the original image. The file name is changed so that the two files can coexist in the same directory. An extender is created to make it easier to pick a proof from a real image. This type of proof **must** reside in the same directory as the image. It should **not** be moved to another directory independently of the image.

Files are **not** overwritten by non-linked proofs. A different name is created to avoid filename conflicts.

Copy file comment to proof comment

If enabled, when creating proofs the image files' comment is copied to the proofs' comment.

Correct gamma

If enabled, automatically corrects gamma on proof creation. Creation is slower but saves time if you usually perform this operation afterwards.

Proof destination directory

This is the directory where PhotoFolio saves created proofs. A directory structure mirroring the original directory structure can be created here. The destination directory is only valid if the proofs are linked.

Mirror image directory structure

When proofs are saved they can be saved directly into the proof destination directory or, if this option is enabled, into a directory structure which mirrors that of the original images'. For example:

The image is stored in: `Graphics:images/pictures/jpegs`
The proof destination directory is: `Ram:proofs`

Without this option enabled, proofs are saved into `Ram:proofs`. When enabled, proofs are saved into `Ram:proofs/graphics/images/pictures/jpegs`

This is useful for creating proofs of directories of images where the directory structure of the created proofs matches that of the original images.

Count files before making

When enabled, PhotoFolio scans the directory and counts the files before commencing a make. This allows PhotoFolio to update progress indicators while a make is in progress. If this is disabled, progress indicators are not displayed.

Add path to list after make

If enabled, once a path has been made the created proof path (if linked) is automatically added to the path list in the **Proofs** folder. This avoids having to manually add the created path to the path list.

Show

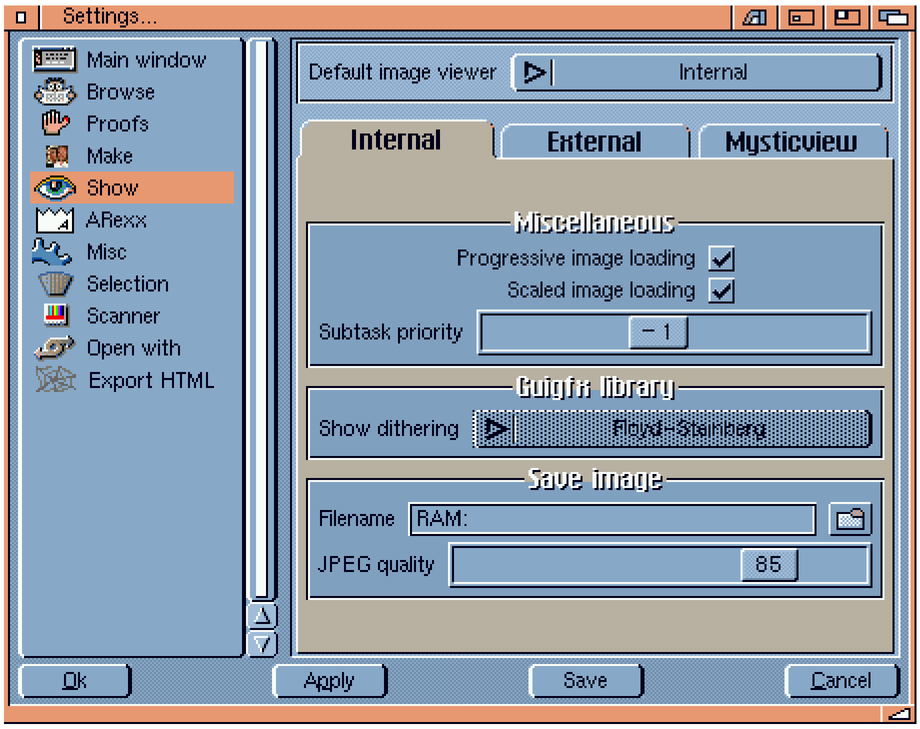
Configures the default viewer and settings for each of the viewing modules.

Default image viewer

The cycle gadget at the top allows selection the image viewer PhotoFolio uses as its default viewer.

1. Internal, use PhotoFolio's internal viewer to view the images.
2. External, use an external viewer to view images.
3. MysticView, use mysticview.library to view the images.

Internal


Settings...

Default image viewer: Internal

Internal | External | Mysticview

Miscellaneous
Progressive image loading ✓
Scaled image loading ✓
Subtask priority: -1

Guigfx library
Show dithering: Floyd-Steinberg

Save image
Filename: RAM:
JPEG quality: 85

Progressive image loading

When enabled, the image being displayed in the show window is seen progressively, as it is being loaded. When disabled, the image won't be seen until loading is complete.

This feature does not function with images loaded using the Amiga datatype system.

Scaled image loading

When enabled, the image being displayed is automatically scaled to fit the screen.

Subtask priority

The show window uses a separate task to load and decode images. This allows other windows in PhotoFolio to continue to function while images are loading. The loading priority of this task can be adjusted, allowing you to decide how much priority you want to give to loading and decoding images. Tweaking this may or may not improve the performance of loading on your machine.

Setting the priority to its maximum value may in fact result in **reduced** performance. If progressive loading is enabled, having the priority set too high will result in the image not showing until loading is completed (because the loading task has a higher priority than the viewing task).

Most users should leave this set to the default value of -1.

Filename

Sets the default path in which to save images when using save as.

Jpeg quality

If the image is saved in jpeg format, this sets the default jpeg quality of the saved image.

Show dithering

This is only available if guigfx.library is being used.

There are 4 dithering modes:

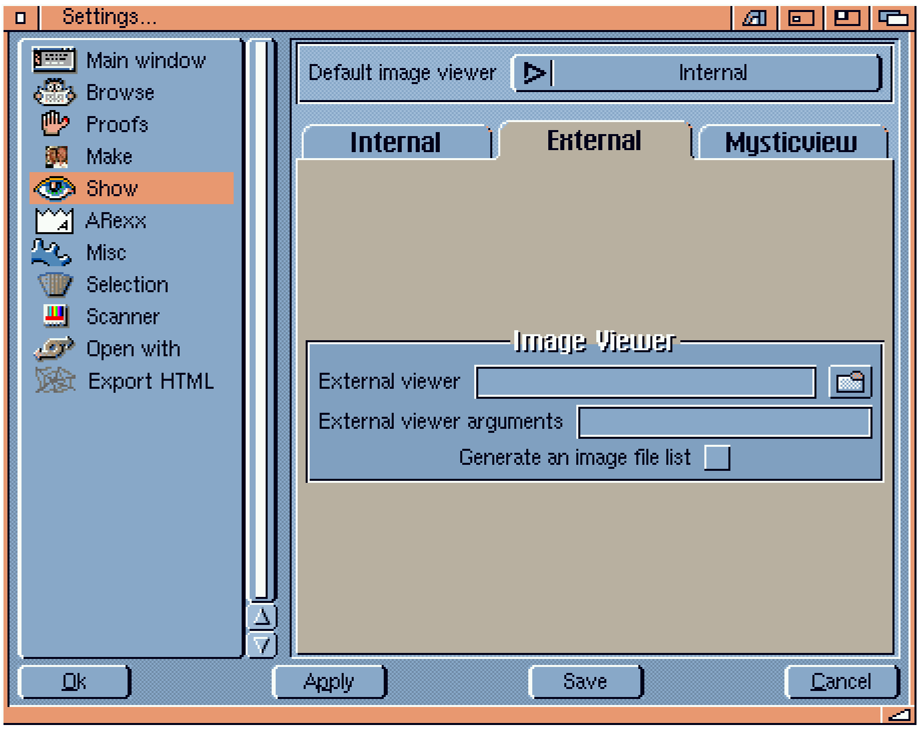
1. None
2. Floyd-Steinberg
3. Random
4. EDD

This setting is for showing images only. If you find your system is slow, set this to *none* and the graphics rendering will be faster, however the quality of the displayed images may suffer.

External

While an external application is running, PhotoFolio cannot be used and all of it’s windows will have wait or busy pointers. When the external application is quit, PhotoFolio will continue execution.

This behaviour can be overridden using the DOS command **Run** or an ARexx script that executes an application using the run command and then sends commands to the running application.



Default image viewer: Internal

External viewer:

External viewer arguments:

Generate an image file list: ☐

External viewer

Enter the `Path:Filename` of the external program used to view images.

External viewer arguments

Enter the arguments to pass to the external viewing program.

When calling the external viewer, PhotoFolio builds an argument string in this manner:

1. The name of the external viewing program.
2. The argument string is appended.
3. Files to be viewed are appended.

The Amiga limits the length of command lines. Consequently, if the number of files appended to the argument string exceeds this limit, the string is truncated and not all of the selected files will be seen.

Generate an image file list

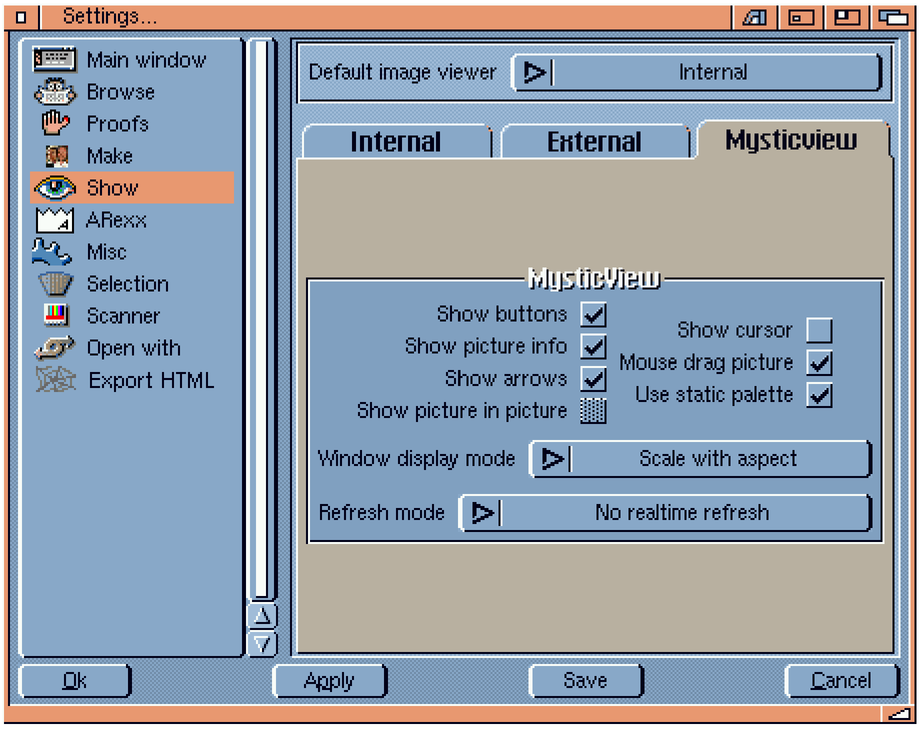
When enabled, PhotoFolio creates a list of images to view in a file. This file is then passed to the external viewer. A `%s` in the external viewer argument string indicates where the file list name is to be placed.

For example:

```
pubscreen=Workbench listfile=%s
```

The external viewer program must support the passing in of images through a file list for this to work. Consult the external viewer's documentation for the correct Shell arguments for passing in file lists.

MysticView



Default image viewer: Internal

Show buttons ✓
Show picture info ✓
Show arrows ✓
Show picture in picture ▨
Show cursor ☐
Mouse drag picture ✓
Use static palette ✓

Window display mode: Scale with aspect

Refresh mode: No realtime refresh

Show buttons
Toggles the button bar in the top of the MysticView window on and off.

Show picture info
Toggles the picture information bar in the bottom of the MysticView window on and off. If enabled, the image filename and width and height are displayed.

Show arrows
When enabled, draws arrows at the picture's borders, when the image is not fully visible in the respective direction.

Show picture in picture
Displays a thumbnail of the image in the top left hand corner of the image. The green box indicates the visible portion of the image.

Show cursor
When enabled, activates a crosshair on the picture, indicating the current cursor position. Scrolling moves the cursor and zooming centers on the crosshair.

Mouse drag picture
Enables scrolling of the image by clicking and holding the left button and dragging the mouse.

Use static palette
When enabled, MysticView uses a static palette instead of a dynamic one (calculated for each picture). This option is meaningless for hicolor and truecolor displays.

A static palette,

- Gives bad results without dithering.
- Produces more stable quality with dithering.
- Triggers MysticView's automatic dithering in most cases.
- Causes less flickering when advancing from one picture to another, because there is no need to clear the display.
- Is not necessarily faster than a dynamic one. This depends on the picture.

Window display mode

Scale with aspect
The image is scaled to fit the window size, maintaining its aspect ratio. If the window is resized, the image is also resized. This guarantees that the image is always fully visible.

Display 1:1 pixel
The image is displayed actual size, regardless of the window size. Each pixel in the image is represented by one pixel on the screen.

Refresh mode

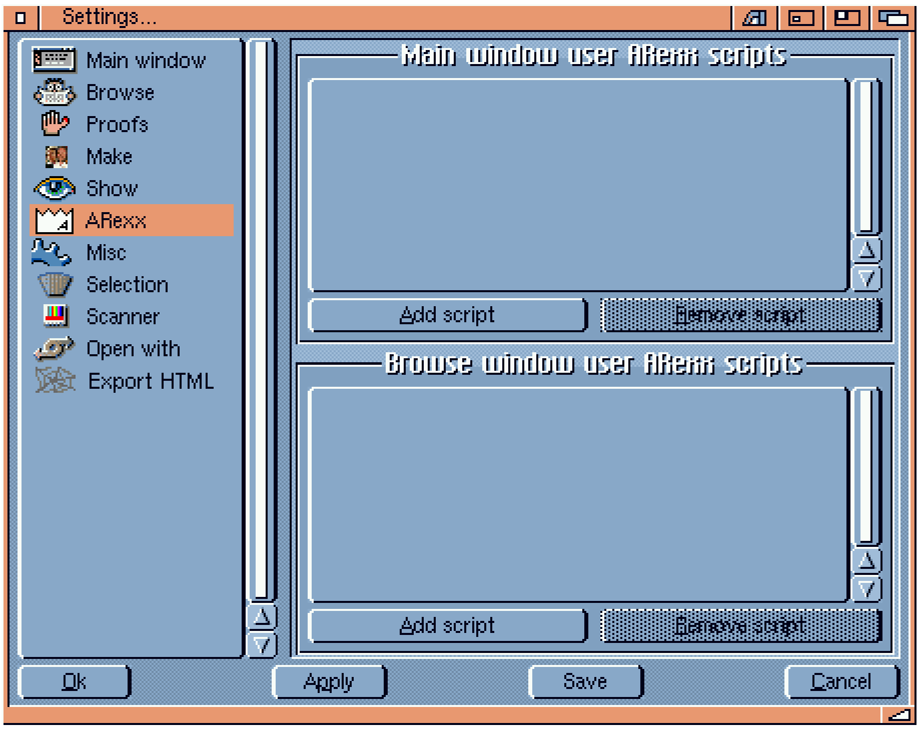
No realtime refresh
No realtime update takes place. The image is rendered in the highest possible quality.

Grid refresh
A 10 x 10 grid is drawn instead of the picture and then the picture is rendered into this grid with the highest possible quality.

Opaque refresh
After a quick preview has been displayed, the image is rendered into the display in the highest possible quality (if necessary). This option increases memory consumption dramatically and should only be enabled on systems with a fast CPU.

ARexx

Set the default ARexx scripts for the main window and browse windows.



The page consists of 2 lists. The top list contains the scripts for the main window and the bottom list contains the scripts for browse windows.

The main window ARexx scripts list can only be edited if the settings window is opened from the main window.

Click the *add script* gadget underneath either of the lists and select an ARexx script from the ASL requester. The script is added to the respective list. After clicking *ok*, *save* or *apply*, the respective window's menus are updated.

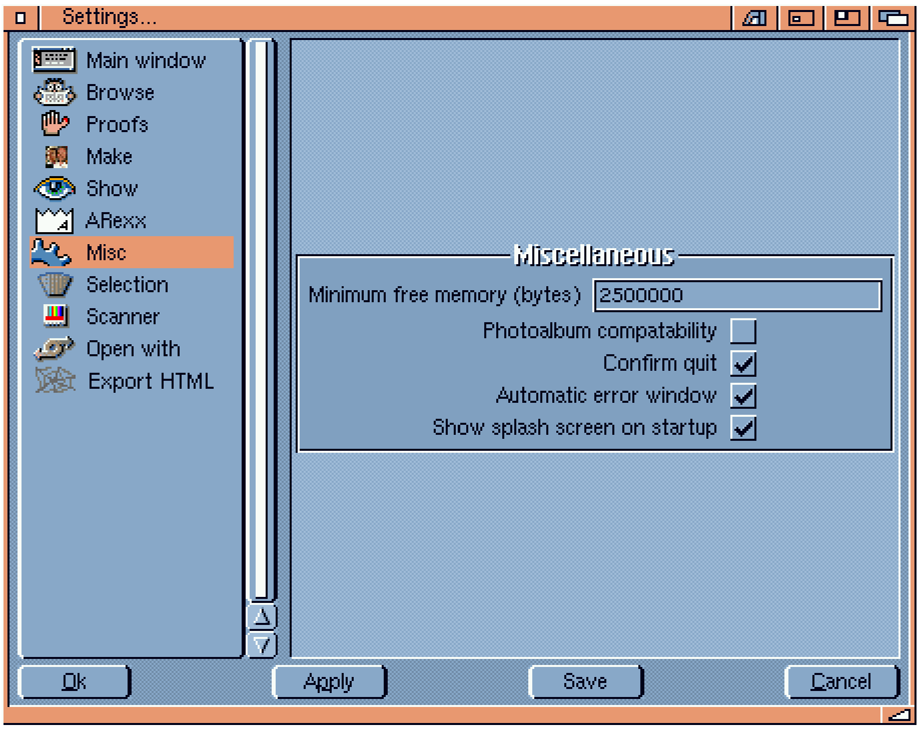
Remove script removes the highlighted script from the list.

Each script list has a context menu to add a script and remove the script clicked on with the right mouse button.

ARexx scripts can be dragged and dropped from the Workbench onto this panel of the settings window to add the dropped script to a list.

Misc

Other settings affecting the global operation of PhotoFolio can be found here.


Settings...

Miscellaneous

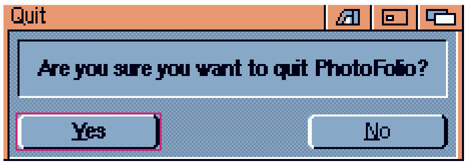
Minimum free memory (bytes) 2500000
Photoalbum compatability ☐
Confirm quit ✓
Automatic error window ✓
Show splash screen on startup ✓

Minimum free memory
If the amount of free memory goes below this setting during a load, PhotoFolio stops loading and displays pagination gadgets.

Photoalbum compatibility
When enabled, PhotoFolio automatically recognises Photoalbum thumbnails stored in the jpeg format and displays them in a browse window.

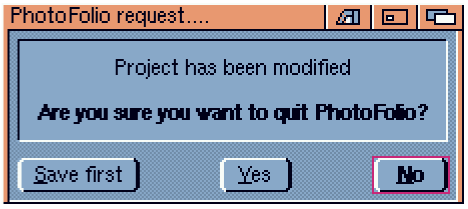
PhotoAlbum thumbnails cannot have their attributes set.

Confirm quit
When enabled and a quit is attempted, a requester opens to confirm quitting PhotoFolio.


Quit

Are you sure you want to quit PhotoFolio?

If the project has been modified, the quit confirmation requester is slightly different.


PhotoFolio request....

Project has been modified

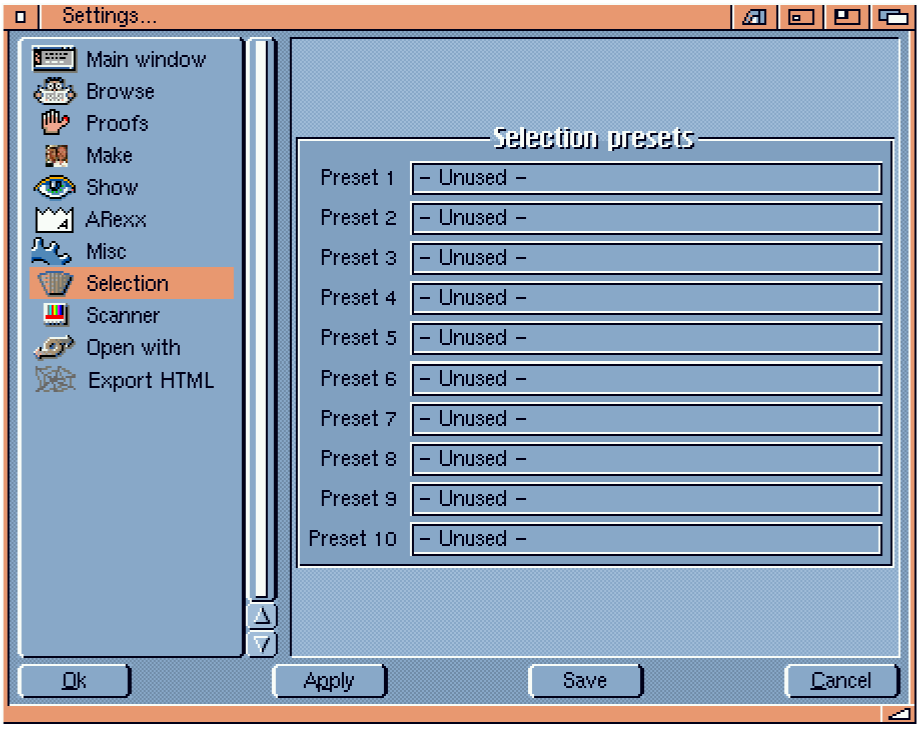
Are you sure you want to quit PhotoFolio?

Automatic error window
When enabled, the error window automatically opens, or if it is already open, is brought to the front, whenever an error occurs. Errors are still added to the error window even with this option disabled.

Show splash screen on startup
When enabled, a splash screen is shown during the PhotoFolio initialization process. This settings is stored as a tooltype in the PhotoFolio program icon. You **must** save the settings, to store any changes made to this setting. Alternatively, you can edit the icon tooltype SPLASH directly from the Workbench.

Selection

The default filename pattern matching strings for pattern select windows are defined here.



Selection presets

Preset 1	- Unused -
Preset 2	- Unused -
Preset 3	- Unused -
Preset 4	- Unused -
Preset 5	- Unused -
Preset 6	- Unused -
Preset 7	- Unused -
Preset 8	- Unused -
Preset 9	- Unused -
Preset 10	- Unused -

Preset 1 - 10
These define 10 default filename pattern matching strings for use with the pattern select window.

These strings **must** conform to the AmigaDOS pattern matching system more frequently used in the Shell and ASL requesters.

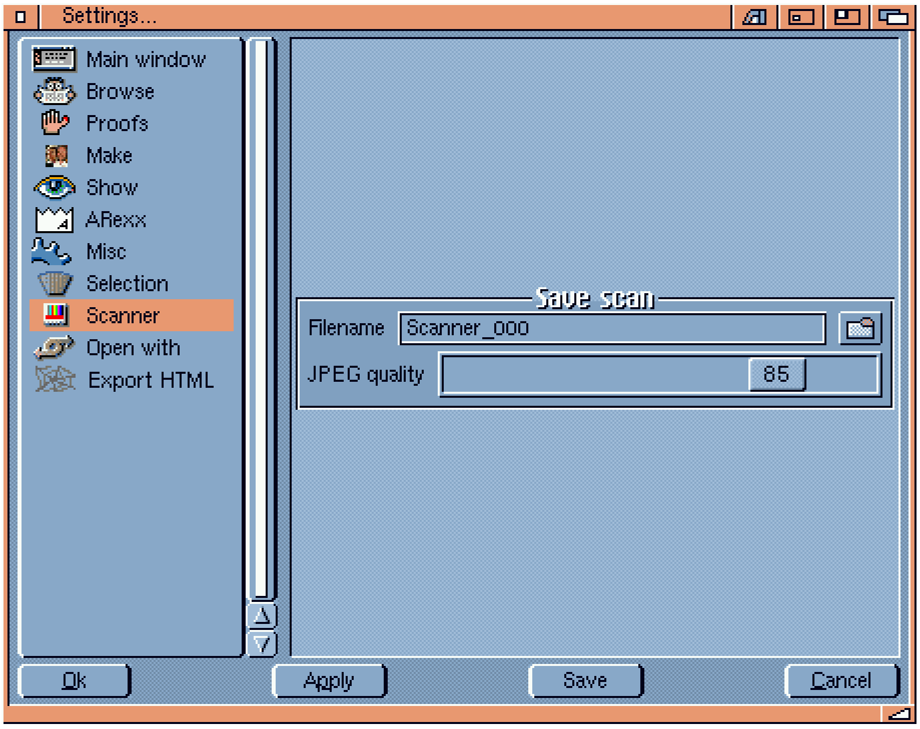
Examples of valid patterns:

`#?.gif`	Match all the files with *.gif* as the last part of their filename.
`#?.(jpg|jpeg)`	Match all the files with *.jpg* or *.jpeg* as the last part of their filename.
`~(#?.(jpg|jpeg))`	Match all the files that **don't** have *.jpg* or *.jpeg* as the last part of their filename.

`#?.(jpg|gif|tif|png)`	Match all the files with *.jpg* or *.gif* or *.tif* or *.png* as the last part of their filename.
`#?sun#?`	Match all the files with *sun* somewhere in the filename.

Scanner

Settings for image grabbing modules can be configured here.



Filename Scanner_000
JPEG quality 85

VLab save filename

Enter the default name to use for saving VLab image grabs. The filename should consist of text characters followed by a 3 digit number e.g. `Graphics:VLab/Grabs/VLGrab000`

PhotoFolio automatically increments the number on the end of the filename each time a grab is saved to ensure batch saved grabs have their own unique filename.

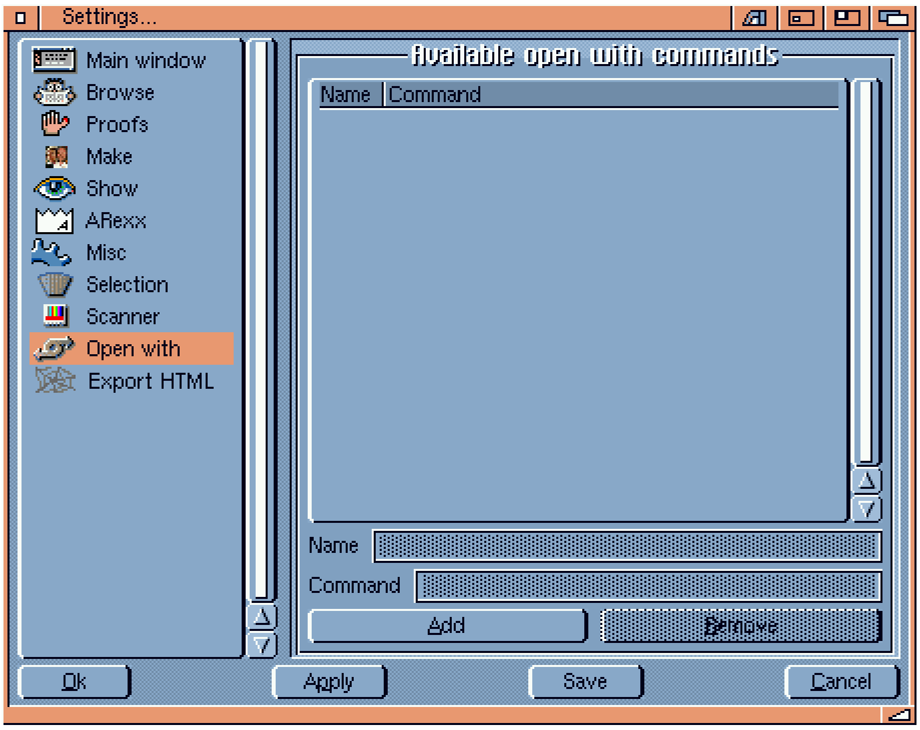
VLab save jpeg quality

The jpeg quality to use when saving a Vlab grab as a jpeg.

Open with

Settings for the proof context menu item open with can be configured here.

While an external application is running, PhotoFolio cannot be used and all of it's windows will have wait or busy pointers. When the external application is quit, PhotoFolio will continue execution.



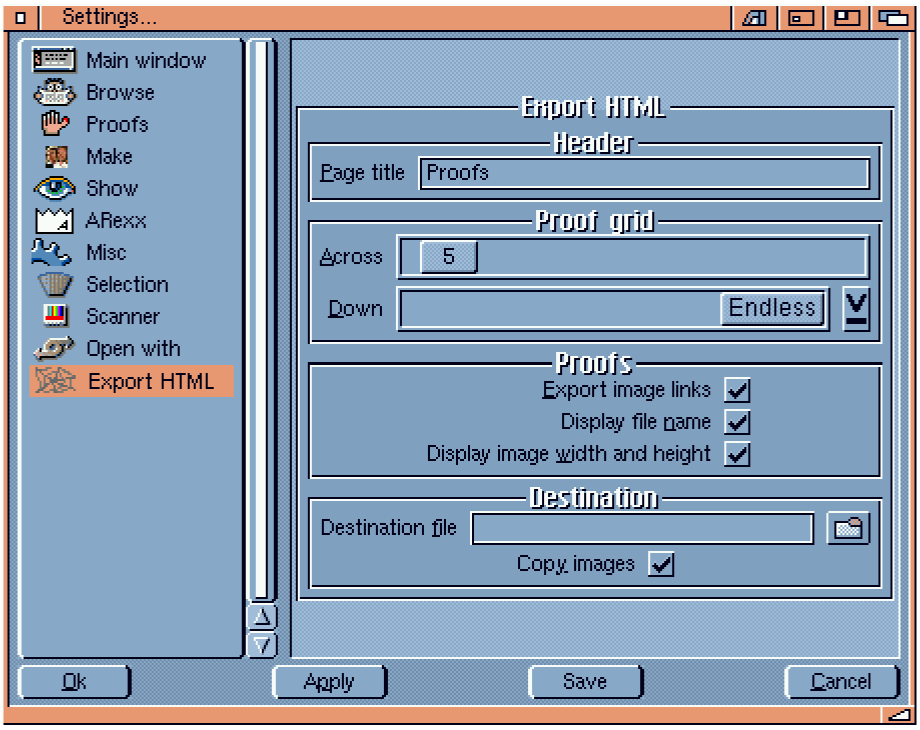
To add items to the list, click *Add*, then enter a *Name* that appears as a title in the context menu and a *Command*, used to perform the open with operation. Place a `%s` in the string where the path/filename of the proof's image should be located.

An example command `SYS:Utilities/Multiview %s` would open the image with Multiview from the Workbench.

Export HTML

If the *Export HTML* module is installed correctly, default settings for exporting proofs as HTML files can be set here.

Even though the settings for this module can be configured from any settings window, they are global settings and changing the module's settings in one window affects **all** other windows.



Export HTML

Header

Page title: Proofs

Proof grid

Across: 5

Down: Endless

Proofs

Export image links ✓

Display file name ✓

Display image width and height ✓

Destination

Destination file:

Copy images ✓

Page title
This string appears in the title bar of the web browser when viewing the HTML file.

Proof grid
Sets the number of proofs across and down that appear in the exported HTML file(s). The down value can be set to *endless* to indicate that the creation should not span across multiple HTML files; instead creating an endless page. The popup gadget contains preset values for commonly used layouts.

Export image links
When enabled, clickable links to the original images are embedded in the exported HTML file. To send the original images to the destination directory as well, enable copy images. (Selecting this option automatically selects copy images as that is probably what was intended.)

Display name
When enabled, displays the image's name underneath the proof image.

Display image width and height
When enabled, displays the width and height of the image underneath the proof image.

Destination
The destination path and filename. The filename is used as a base for the created HTML files and the path part of the filename is where subdirectories containing proofs and images are stored. e.g. `ram:steeple.html`

The path **must** exist before starting the export.

Copy images

If enabled, copies the images to a subdirectory in the destination path. To allow the viewing of the images in the HTML file, image links must also be enabled.