# Commander

**COLLABORATORS**

| | *TITLE* :  Commander | | |
|---|---|---|---|
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | | July 31, 2024 | |

**REVISION HISTORY**

| NUMBER | DATE | DESCRIPTION | NAME |
|---|---|---|---|
| | | | |

# **Contents**

# Chapter 1

# Commander

## 1.1 Commander Documentation

```
 Commander

Introduction
FAQ
Development and Testing

Installation                        Menus
Tutorial

Configuration                       ARexx Errors

Future Enhancements
Known BUGS                          Legalness


New in this version                 Getting Updates
Previous versions                   Locating The Author
```

## 1.2 Introduction

```
Introduction
------------


About Commander
---------------
One of the drawbacks of many programs is the lack of a customizable user
interface. Functions, such as ARexx scripting and plug-ins, often have to
be started via file requesters or multi-level menus. This approach can
slow down your workflow - and often hinder your creative energy.

ImageFX, for example, lacks a customizable (or updated) command
interface. When new modules/hooks scripts are added to the program
(home-made, pd, shareware, or commerical) you're required to select those
functions through a file requester or other means. This can slow down the
creative process. Commander is the solution to that problem.
```

Features
--------
- Can control any program that has an ARexx command set. However, it is
  ONLY available for: ImageFX and there are no plans to develop this
  product further. See ImageFX special features below.

- Extensive customization allows the user to design multiple
  interfaces for all kinds of different approaches to working.
  Just about everything is customizable, including: layout, fonts,
  gadgets, colors, images, etc.

- Openable on any public screen. Can be run multiple times on the same
  screen and/or on different screens, each with a different or same GUI.

- Can control more than one application from one GUI.

- GUIs are designed external to the main program. This minimizes memory
  requirements and increases the interface speed.

- Includes useful examples and tutorials to get you started. Example
  files can be modified to meet your needs, or new interfaces can be
  created.

ImageFX Features
----------------
- Works with any version of ImageFX under WB2.x or higher.

- Full control of ImageFX via it's ARexx command set. Now you can have
  immediate access to all those extra hooks and effects that aren't part
  of the standard ImageFX GUI.

- Commander can start automatically when ImageFX starts. Quitting
  Commander can also quit ImageFX.

- Can be run multiple times on different screens, for multiple
  simultaneous iterations of ImageFX.


What Will Commander Do?
-----------------------
Commander allows you to customize your own GUI for supported
applications, giving you the ability to directly access your favourite
ARexx commands, scripts, and plug-ins in a way that works best for you.

If you use a large screen (1280x1024) you may wish to have numerous
independently movable panels of grouped commands. If you use a smaller
display (640x400) you may prefer to have a page flipping type interface.
Or you may want to have multiple configurations for different kinds of
workflow. Commander can do it all.

What Won't Commander Do?
------------------------
Commander won't help you to be creative, but it will help your creative
flow by avoiding the hinderance of a cumbersome GUI.

Requirements

```
------------
- WB 2.x+
- Text editor
- A supported application (see above)
```

## 1.3  FAQ

```
FAQ: Frequently Asked Questions
-------------------------------
```

Q: I'm not a computer programmer, can I still use Commander to create my own interfaces?

A: Yes. Although Commander does have it's own "language" for defining interfaces (modules), it's not as difficult to understand as a programming language might be. The Commander definition language uses simple concepts, and basic commands to do what it does.

Commander comes with many ready-to-go interfaces, and tutorials on how to modify them to suit your needs. There's also a tutorial and detailed documentation on creating your own custom interfaces.

See the included sample definition files to get an idea of what the definition language looks like.

Technical support is available for free via email. So if you're having problems, we'll be there to help you out.

## 1.4  Development and Testing

```
Development and Testing
-----------------------
```

Commander is developed on an A3000T/060/50MHz with 2 Meg CHIP / 136 Meg FAST memory and Cybervision64/4 Meg at 1280x1024.

It has been beta-tested on an A3000T with 2 Meg CHIP / 12 Meg FAST memory at 640x400, and an A600 with 2 Meg CHIP at 640x400.

Commander has been tested under Kickstart 3.x.

## 1.5  Installation

```
Installation
------------
```

- Copy/Drag the entire Commander directory into your ImageFX main directory.

- The recommended way to get Commander to start automatically when
  ImageFX is started, is to copy the "Startup.ifx" file into ImageFX's

"Rexx" directory. Alternatively, you can use ImageFX's MACRO tooltype
to start Commander. This has the disadvantage of ImageFX popping up
it's "ARexx macro in progress" window.

        MACRO=Commander/Startup.ifx

If you prefer, Commander can be started manually by just
double-clicking on it's icon after starting ImageFX. Or, for that
matter, without even starting ImageFX - in which case the interface
won't actually be able to do anything, but you can play with it
nonetheless.

– You may want to set the tooltypes as described here in the  Configuration
   section. Or, if you're using the autostart functionality, you should
   instead edit the "Startup.ifx" file to set the options there. If you're
   using ImageFX on it's own screen (ImageFX.1), Commander is already set
   to go.

– Copy the contents of "Commander/fonts/" to your FONTS: directory. The
   included Topaz2, and Topaz2p fonts are designed for use with the demo
   modules. These modules have been defined to use these fonts, but they
   are not necesarily required for custom modules you create. Topaz2 is a
   nicer looking version of Topaz, Topaz2p is a proportional version of
   Topaz2.

   The demo modules use fonts included with WB: Garnet/16 and Helvetica/11
   and /13. datatypes.library is required for image support.

– CycleToMenu (aminet: util/cdity/CycleToMenu-21.lha) is highly
   recommended for use with the Amiga's Cycle gadgets. With it, when you
   click on a cycle gadget, a list pops up so you can see all the options
   at a glance. This is great for Commander and all other applications.


## 1.6  Configuration

Configuration
-------------

Configuration is supported through use of Workbench ToolTypes or CLI
parameters. If any options are not included, the stated defaults will be
used.

Configuration options are not case sensitive.

Invalid options will be ignored. In this case, the default values will
be used.


The following configuration commands are supported:

  Font
    Description: Sets the font for requesters. This setting has no
                 control over fonts defined within modules. Can be
                 enclosed within quotes, but is not necessary.

```
    Default: Topaz2p.font

    Example: Font=helvetica.font


  FontSize
    Description: Sets the font size for requesters. This setting has no
                 control over fonts defined within modules.

    Default: 8

    Example: FontSize=11


  Last
    Description: Opens the last opened module upon startup. No file
                 requester is presented unless the last used module can
                 not be found.

    Valid values: Last

    Default: NO

    Example: Last


  Module
    Description: Opens the given module upon startup. No file requester
                 is present unless the module can not be found.

                 The module name is referenced as the path and filename
                 within the 'modules' directory.

    Default: NONE : FileRequester

    Example: Module=AllInOne_List.cmndr


  OffsetX
    Description: This sets the X offset of the main window.

    Valid values: >= 0

    Default: 0

    Example: OffsetX=50


  OffsetY
    Description: This sets the Y offset of the main window.

    Valid values: >= 0

    Default: 14

    Example: OffsetY=50
```

```
  PubScreen
    Description: Sets the publicscreen that Commander will open on.

    Default: "Workbench"

    Example: PubScreen="ImageFX.1"


  RexxPort
    Description: Sets the ImageFX ARexx port to send commands to.

    Default: 1   (IMAGEFX.1)

    Example: RexxPort=2     (IMAGEFX.2)


  WindowH
    Description: Sets the height of the Commander main window.

    Valid values: >= 0

    Default: 13

    Example: WindowH=50


  WindowW
    Description: Sets the width of the Commander main window.

    Valid values: >= 0

    Default: 153

    Example: WindowW=200
```

## 1.7 ARexx_Errors

```
ARexx Errors
------------


Result1:

RC_OK     0  /*  success                   */
RC_WARN   5  /*  warning only              */
RC_ERROR  10 /*  something's wrong         */
RC_FATAL  20 /*  complete or severe failure */

Result2:

ERR10_001  /*  program not found          */
ERR10_002  /*  execution halted           */
ERR10_003  /*  no memory available        */
ERR10_004  /*  invalid character in program */
ERR10_005  /*  unmatched quote            */
```

```
ERR10_006   /*   unterminated comment        */
ERR10_007   /*   clause too long             */
ERR10_008   /*   unrecognized token          */
ERR10_009   /*   symbol or string too long   */

ERR10_010   /*   invalid message packet      */
ERR10_011   /*   command string error        */
ERR10_012   /*   error return from function  */
ERR10_013   /*   host environment not found  */
ERR10_014   /*   required library not found  */
ERR10_015   /*   function not found          */
ERR10_016   /*   no return value             */
ERR10_017   /*   wrong number of arguments   */
ERR10_018   /*   invalid argument to function */
ERR10_019   /*   invalid PROCEDURE           */

ERR10_020   /*   unexpected THEN/ELSE        */
ERR10_021   /*   unexpected WHEN/OTHERWISE   */
ERR10_022   /*   unexpected LEAVE or ITERATE */
ERR10_023   /*   invalid statement in SELECT */
ERR10_024   /*   missing THEN clauses        */
ERR10_025   /*   missing OTHERWISE           */
ERR10_026   /*   missing or unexpected END   */
ERR10_027   /*   symbol mismatch on END      */
ERR10_028   /*   invalid DO syntax           */
ERR10_029   /*   incomplete DO/IF/SELECT     */

ERR10_030   /*   label not found             */
ERR10_031   /*   symbol expected             */
ERR10_032   /*   string or symbol expected   */
ERR10_033   /*   invalid sub-keyword         */
ERR10_034   /*   required keyword missing    */
ERR10_035   /*   extraneous characters       */
ERR10_036   /*   sub-keyword conflict        */
ERR10_037   /*   invalid template            */
ERR10_038   /*   invalid TRACE request       */
ERR10_039   /*   uninitialized variable      */

ERR10_040   /*   invalid variable name       */
ERR10_041   /*   invalid expression          */
ERR10_042   /*   unbalanced parentheses      */
ERR10_043   /*   nesting level exceeded      */
ERR10_044   /*   invalid expression result   */
ERR10_045   /*   expression required         */
ERR10_046   /*   boolean value not 0 or 1    */
ERR10_047   /*   arithmetic conversion error */
ERR10_048   /*   invalid operand             */
```

## 1.8  Tutorial

```
Tutorial
--------
```

If you've configured Commander to autostart, just start ImageFX.
Otherwise, double-click on the Commander icon.

If this is the first time you've run Commander, you will be prompted with
a file requester to choose a module to load. If not, select Project->Open
from the menus. Select the "Demo.cmndr" module. This is a somewhat ugly
and useless module, but it does demonstrate a few of the capabilities of
Commander. You can include one or more images as part of your display
(hopefully something a bit prettier than the one used here). Standard
gadgets, Button, ListView, Cycle, and MX can all be used to attach
commands to. In this example, the gadgets won't do anything though.

This module also demonstrates the use of multiple fonts and the ability
to add shadows, styles and color to text. You can also draw 3D borders,
and filled rectangles with simple definition commands.

Now, from the menus, select Project->Open and choose
"AllInOne_OnePanel.cmndr". This module demonstrates the ability to
configure multiple pages on one panel. This module not only includes all
of the functions of the standard ImageFX panels, but also adds quick
access to all of the newer hooks and modules. Another great addition is a
configurable quick access ARexx script panel. Keep in mind that
everything you are seeing is customizable to your needs: you can remove
any of the pages, add more pages, change the functions on the pages, or
just about anything else you can think of. The Tools page is also a nice
addition, allowing you to quickly record macros and access AutoFX and
other ImageFX tools.

Let's see what those other functions are in the menus. Drag the
"Commander" title bar window to somewhere else on the screen. Notice how
the command panel below it follows it wherever you go. Now, select
Prefs->Snapshot. This saves the new position of the current module's
windows. To see if it worked, select Project->Restart. You should see the
module close and then reopen again in the new position.

Uncheck Prefs->Drag All Windows. Now drag the "Commander" title bar
window again. This time, the command panel stays put. Why would you want
to do that? Well, with this module, it's not particularly useful. But,
particularly when using several modules, you might want to have the main
window always appear in the same place. Select Project->Restart again.
Everything moves back to the position you just snapshoted.

Check Prefs->Ignore Snapshot and then Project->Restart again. This time
the command panel reverts back to it's predefined position. The
"Commander" title bar window didn't move though – that's because it's
position is not predefined (it can be set using tooltypes). If you want
to force Commander to forget the snapshot settings, select
Prefs->UnSnapshot. Note that, these snapshot functions are specific to
each module. So, UnSnapshot only affects the current module, other
snapshoted modules remain as they were.

We've also included some other sample modules. "AllInOne_List.cmndr" puts
all the ImageFX commands into one ListView. If you've got a screen size
of 750 or taller, "AllInOne_List-750.cmndr" is a taller version of the
same. "AllInOne_MovePanels.cmndr" is an alternate approach, similar to
ImageFX's built-in GUI, that groups commands into multiple moveable
panels. This isn't a good choice for people with small screens, as with
all the added functions we've included, things will be overly crowded. Of
course, all of the individual panels can be snapshoted in your favourite

positions. "AllInOne_TwoPanels-711.cmndr" is another alternative for
people with a screen taller than 711.

Of course, since Commander can also run external ARexx scripts, you could
easily configure gadgets to control other programs along with ImageFX.

So, are you excited about the possibilities? Go here to find out
 How to But It


## 1.9  Menus

```
Menus
=====

Project ============

  Restart            - Reloads the current module.
  Open               - Opens a new module.
  About...           - Displays information about Commander and the author.
  ---------------
  Quit               - Quits Commander.

Prefs ==============

  Snapshot           - Saves the positions of the main Commander window
                       and all the current module's window positions for
                       recall the next time that module is loaded.
                       Snapshot information is retained for all modules.
  UnSnapshot         - Deletes the saved window positions of the current
                       module.
  ------------------
  Ignore Snapshot    - When enabled, Commander will ignore any previously
                       snapshoted window positions when opening a module.
  Drag All Windows   - When enabled, moving Commander's main window causes
                       all module windows to move similarly. When
                       disabled, the Commander main window will move
                       independently.
```


## 1.10  Legalness

```
Legalness
---------
```

Commander and all of its components are copyright ©2000 by dhomas trenn
and young monkey. The files included with this distribution may not be
altered in any way.

The software concept and design remain the property of the author,
dhomas trenn and young monkey.

As of April 2, 2001, Commander (Amiga) is freely distributable and may
appear in any public domain software archives, or on any commercial disks

or CDROMs.

The author is in no way liable for any damage resulting from the use of
this program.

## 1.11 Known BUGS

Known BUGS
----------

- None known.

## 1.12 Future Enhancements

Future Enhancements
-------------------

- No further development planned.

## 1.13 Getting Updates

Getting Updates
---------------

- No further development planned.

The latest version and other information can be found in the Commander
support area of our website:

    http://www.youngmonkey.ca/hands/files/Commander/index.html

## 1.14 Locating The Author

Locating The Author
-------------------

Through INTERNET EMAIL at:

                        info@youngmonkey.ca

On the Web:

                    http://www.youngmonkey.ca/

## 1.15   Previous versions

```
Previous versions
-----------------
```

– First release.

## 1.16   New in this version

```
00.Mar.25
---------
```

– First release.