VWM

COLLABORATORS

	TITLE : VWM	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 28, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

VWM

1.1 VWM, Virtual Window Manager, version 1.2

VWM, Virtual Window Manager, version 1.2
For Amiga Technologies - AMIGA

VWM, the virtual screen aware window manager, is a small, fast, and system friendly virtual screen aware program that gives you 100% control over windows' positions.

I would like to thank my beta testers Jonas Edlund, Matthew Warren, Nick Curcio and Peter Bergström. Without them VWM would not exist.

Only Amiga makes it possible.

1.2 disclaimer

Disclaimer

The author of this program is not responsible for any loss of data or damages to software or hardware that may result directly or indirectly from the use of this program. Use this software at your own risk.

1.3 copying

Copying

Commercial use of VWM or inclusion of VWM in commercial applications requires the expressed written consent of the author. Please contact the author directly for further information.

Explicit permission to publish VWM is given to the Fred Fish and the Aminet CD series.

1.4 registration

Registration

Although I am releasing VWM as freeware, I would like to know how many users are using it. To achieve this, VWM opens a reminder window (+ 10 seconds of delay) each time that it reads its prefs file IF it can NOT find a valid user key file in S:

But you really don’t need to worry about that. If you decide to continue using VWM, simply email me your name and email address, and I will send you your free key file. I will not pass the incoming information to anyone else. I will handle the registrations once or twice per week. Fair enough ehh?

1.5 requirements

Requirements

- AmigaOS 2.04 or better.

1.6 installation

Installation

If you want VWM to be running every time you boot up, copy the VWM executable and its icon to SYS:WBStartup, or copy it wherever you like and run it from your User-Startup.

Create a prefs file for the program and copy it to ENV: for the default configuration. This path can be changed, but if you choose to use ENV: don't forget to copy the prefs file to ENVARC: for permanent use. Creating the prefs file and how to alter the path will be discussed later.

1.7 introduction

Introduction

The first generation of Amiga computers that were released more than 10 years ago were capable of showing flicker-free NTSC (640x200) and PAL (640x256) screens. Today, AGA Amigas can show DBLNTSC (640x400) and DBLPAL (640x512) screens without additional hardware. Furthermore, Amigas that are equipped with graphics cards can show screens of up to 1600x1200 pixels.

When working in hi-res 640x200 screens, it really doesn't matter where programs' windows are located. The resolution is so low, and screens are so small that most windows cover the entire visible area of the screen. When working in larger screens though, you really want to have exact control over where applications open their windows. Sadly only a small percentage of modern Amiga applications as well as all the MUI programs honor the users and open their windows in user defined positions.

Another problem is that most of the Amiga programs don't understand virtual screens, a feature that allows an Amiga to open screens that are larger than the monitor's display area. A virtual screen may be any size up to the maximum allowed by that screen type and the amount of graphics RAM in the Amiga. The user is only able to view the visible size at any given time.

To understand the core of the problem, you have to consider the following scenarios:

1) You are running your Workbench in a 1024x768 screen. The mouse pointer is located at the position 1000x700 when you launch your favorite calculator which opens its window at location 10x10. To access the calculator, you have to move the mouse pointer over calculator’s window. Furthermore, if you want to have the window located somewhere else in the screen, you have to move it manually.

   Example VWM.prefs code to solve this problem.

2) You run your favorite word processor in its own window in a 1024x768 Workbench screen. Each time you launch the spell checker of the word processor, it opens its window in the center of the screen, thus covering part of the most important area of the display where the editing text is located. To solve this problem you have to manually move the spell checker window to another location in the screen.

   Example VWM.prefs code to solve this problem.

3) Your favorite CD-player opens its window in the center of the Workbench screen. You are not satisfied with the location and want to have this window in the position 0x16 in your Workbench. Unfortunately, the CD-player is not very user-configurable, so you are stuck with a nice program that doesn’t let you configure its position.

   Example VWM.prefs code to solve this problem.

On virtual screens the problem is much more serious. I am not going to mention any program name, but 99% of the Amiga applications that I have tried do not understand virtual screens ;(The only program I use regularly that understands virtual screens is reqtools.library. I tip my hat to its developer ;)

In the following scenarios, the programs are launched in a 1024x1536 virtual Workbench screen that consists of 2 1024x768 visible parts.

To understand the problem, we’ll call the first visible area (0x0 - 1024x768) ’Area1’ and the second visible area (0x769 - 1024x1536) ’Area2’.

1) You are working in Area2. You launch your favorite calculator that opens its window in Area1. To access the calculator, you have to switch to Area1, place the pointer over the drag bar of the calculator, drag its window to Area2, and place it in a desirable location.

   Example VWM.prefs code to solve this problem.

2) You are working in Area1 when you launch your favorite CD-player which opens its window in the center of the virtual screen that, in our case, is NOT the center of the visible area of the screen (Area1). As a result, the

window is located partly in Area1 and partly in Area2. Again, you manually have to relocate the window to a better location.

Example VWM.prefs code to solve this problem.

3) You are editing a text file in your editor that is located in Area2. The editor opens its "Jump to line #" window somewhere in Area1. To enter your data, you have to switch to Area1, type a line number, and then switch back to Area2 to continue working on the text.

   Example VWM.prefs code to solve this problem.

4) You are editing a text file in your editor that is located in Area2. The editor centers its "Find and replace" window under the mouse pointer position. Unfortunately, if the mouse pointer is located very close to the top border of the visible area (Area2) when you invoke the "Find and replace" function, the program opens the window partly in Area1 and partly in Area2. Again, you have to find the drag bar of the window and move it to Area2.

   Example VWM.prefs code to solve this problem.

5) The File-Manager program that you use opens its window in a predefined location in Area1. The result is that when you are working in Area2 and you need to use the File-Manager, you have to switch to Area1 to access it.

   Example VWM.prefs code to solve this problem.

As you can see by now, it is almost impossible to use the computer in an efficient and creative way when you can't control the exact position of your applications' windows and when programs do not support and understand virtual screens, a feature that was first introduced in AmigaOS 2.04 and that allows an Amiga to open screens that are larger than the monitor's display area.

VWM features 2 ways of defining the new location for the target window, setting an absolute or relative position or using a preset positioning technique usually related to the mouse position. These will be discussed later.

1.8 solution1

Example VWM.prefs code

```
; -- Open my favorite calculator under the mouse pointer.
;
; -- Use the SourceTitle= keyword if the calculator has a valid title.
SourceTitle=Calc
```

```
DestinationFlag=FollowMouse
#

OR

; -- Open my favorite calculator under the mouse pointer.
;
; -- Use one or more of the other source definition keywords if the
;    calculator window does not carry a valid title.
SourcePosition=10,10
SourceDimensions=150,300
DestinationFlag=FollowMouse
#
```

1.9 solution2

```
  Example VWM.prefs code



; -- Open my spell checker at absolute position 10,100.
SourceTitle=Speller
DestinationPosition=10,100
#

OR

; -- Open my spell checker at position 10,100.
;
; -- For the DestinationPosition= keyword, the default DestinationFlag=
; -- is "Absolute", so the 2 definitions in this page do exactly the
; -- same thing.
SourceTitle=Speller
DestinationPosition=10,100
DestinationFlag=Absolute
#
```

1.10 solution3

```
  Example VWM.prefs code



; -- Open my favorite CD player at position 0,16.
;
; -- Here we use pattern matching mechanism of AmigaDOS for the window
; -- title. CDPlay#? matches any window title that starts with "CDPlay"
SourceTitle=CDPlay#?
DestinationPosition=0,16
#
```

1.11 vsolution1

Example VWM.prefs code

```
; -- Open my calculator within my reach when I’m anywhere on my
: -- (virtual) WB. The calculator window will be opened under the mouse
; -- pointer but far enough away from the closest edges of the visible
; -- screen.
SourceTitle=Calc
DestinationFlag=RespectView
#
```

1.12 vsolution2

Example VWM.prefs code

```
; -- Always locate CD player’s window in the the center of the area
; -- (the actual part of the screen) that I am working in.
SourceTitle=CDPlayer
DestinationFlag=CenterWindow
#
```

1.13 vsolution3

Example VWM.prefs code

```
; -- Always open my editor’s "Jump to line #" requester under the mouse
; -- pointer in the area (the actual part of the screen) that I am
; -- working in.
;
; -- Pay attention that the pound sign (#) in the title of window has
; -- been changed to a question mark (?) because AmigaDOS uses the
; -- pound sign (#) for its pattern matching mechanism.
;
; -- Please consult your AmigaDOS manual for further information about
; -- pattern matching feature of AmigaDOS.
SourceTitle=Jump to line ?
DestinationFlag=FollowMouse
#
```

1.14 vsolution4

Example VWM.prefs code

```
; -- Open my editor’s "Find and Replace" requester under my mouse
; -- pointer but keep it in my view.
;
; -- Pay attention that as VWM supports and understands virtual
; -- screens, it will always open the "Find and Replace" window in the
; -- area (the actual part of the screen) that I am working in.
SourceTitle Find and Replace
DestinationFlag=RespectView
#
```

1.15 vsolution5

Example VWM.prefs code

```
; -- Always open my File Manager in fixed position 10,20 that is
; -- related to the point (0x0) of the visible part of the (virtual)
; -- screen.
;
; -- The result is that the File Manager’s window becomes open in fixed
; -- positions in both areas of the virtual Workbench screen.
;
; -- Just try it, and you’ll wonder how you could live without it. ;)
SourceTitle=File Manager
DestinationPosition=10,20
DestinationFlag=Relative
#
```

1.16 features

Features

VWM, the Virtual Window Manager, is a small, fast and system friendly virtual screens aware program that gives you 100% control over windows’ positions.

VWM features 2 ways of defining the new location for the target window, setting an absolute or relative position (Coordinate Positioning) or using a preset positioning technique usually related to the mouse position (Preset Positioning Techniques). These will be discussed later.

VWM can force your programs to:

- open their windows in absolute positions related to the start

```
     point (0x0) of the active screen.

   - open their windows in absolute positions but related to the
     point (0x0) of the "active part" of the screen (for virtual
     screens) (more about this later in this document).

   - center their windows under the mouse pointer (FollowMouse).

   - locate their windows under the mouse pointer but respect the
     view, that is to locate the windows far enough away from the
     closest edges of the visible screen but still under the mouse
     pointer (RespectView).

     FollowView(TM), known as RespectView in this program, is
     Copyright ® 1993-1997 Bahman Moallem and was first introduced
     in MagicWords, the multilingual word translator for the Amiga
     back in 1993.

   - open their windows in the center of the visible part of the
     screen (CenterWindow).

   - open their windows in the corner of the visible part of the
     screens which is farthest away from the mouse pointer position
     (UseCorners).
```

1.17 usage

```
Usage


Upon startup, VWM reads a prefs file that contains information
about what windows it should work on.  If reading the prefs file
doesn’t cause any trouble, VWM patches the OpenWindow() and
OpenWindowTagList() functions of the AmigaOS intuition.library and
sits in the background waiting for programs to try to open their
windows.  If a window that is about to be opened matches one of the
object windows in the prefs file, VWM will modify the structure of
it and then open it at the user defined position.

You can start VWM from a Shell or Workbench.  The program accepts
the following 2 arguments or tooltypes:

    About: (Optional) Pops up the program’s/user’s information
           window and quits afterwards.

    Prefs: (Optional) By default VWM looks for the file
           ENV:VWM.prefs upon startup.  You can use the ’prefs’
           argument if you locate VWM’s prefs file somewhere else.

    For example:

        Prefs=VWM.prefs tells VWM to use the prefs file ’VWM.prefs’
        that is located in the same directory as the program
        itself.
        Prefs=S:VWM_Lisa.prefs tells VWM to use the prefs file
        'VWM_Lisa.prefs' located in the directory S:
```

When run from a shell or somewhere in your startup make sure to use 'run >nil: VWM' if you want to be able to close the console window.

VWM doesn't allow you to launch multiple copies of itself.

To quit VWM, you should send a break to VWM's main task with tools like Scout, XOper or ARTM. This is the same as pressing CTRL-C or using "Break `Status COM=<path>VWM` C" in the Shell if you start the program from a Shell window.

1.18 configuration

Configuration

ENV:VWM.prefs is an ordinary ASCII file that you can create with your favorite text editor.

Any line that begins with a semicolon (;) or a space ( ) is considered a comment line and will be ignored.

VWM uses the AmigaDOS notification mechanism and rereads the prefs file each time it is saved. This allows you to edit the prefs file and autoload it to memory without first quitting the program. If VWM finds an error in a reread prefs file, it will keep using the proper prefs file that was loaded last. In case of a problem, VWM gently informs you about what and where in the prefs file the faulty information is located.

The ENV:VWM.prefs file consists of a number of objects. An object is a group of lines that describes what window is affected and how it is affected. Each object must be separated with a pound sign (#).

Use space characters only when they are needed, for example as part of a window or screen title.

The prefs files uses two types of definitions, source definitions and destination definitions. Each object must contain at least one source-type defintion and at least one destination-type definition.

There are two types of source definitions:

Window Definitions
define how a window is grabbed by VWM.

Screen Definitions
define what screen coordinates are used for the destination definition calculations.

There are also two types of destination definitions:

Coordinate Positioning
defines exactly where, in absolute or relative coordinates,
to position the window.

Preset Positioning Techniques
define where to position the window usually relative to the
current mouse position.

1.19 window definitions

Window definitions

There are four ways to define how a window is grabbed by VWM:

SourceTitle=Window’s title (pattern matching is accepted)

Here you can use the pattern matching mechanism of AmigaDOS for window titles.

```
SourceTitle=Amiga#?
```

matches any window title that starts with "Amiga".

```
SourceTitle=  This window has a very long title
```

matches any window that carries the title
" This window has a very long title"
(Note the two leading spaces.)

If the target window doesn’t have a valid title, you can use the SourcePosition= and/or SourceDimensions= keywords for the source object:

SourcePosition=Left Edge,Top Edge (x-y pixel position)

```
SourcePosition=100,200
```

matches any window whose LeftEdge and TopEdge are at x=100 and y=200.

SourceDimensions=Width,Height

```
SourceDimensions=300,400
```

matches any window that has a width of 300 pixels and a height of 400 pixels.

That’s all there is to it...

...but that’s only three definitions. Where’s the fourth one???

Here it is:

For the greatest control you can use the SourcePosition= and SourceDimensions= keywords together thereby selecting a window by its initial position and size.

If you use more than one of the above keywords for an object, only the last one will be considered as the valid one except, of course, if you are using the SourcePosition= and SourceDimensions= keywords together.

1.20 screen definitions

Screen definitions

Every window belongs to a screen, and VWM tries to gather this info from the target object. If, for any reason, you want to change this behavior, you can use one of the following three optional Screen keywords.

Please observe that VWM only uses this info for calculating the new position for the target object. VWM DOES NOT modify anything but start position of a target window.

```
  SourceScreen=Screen’s Title (pattern matching is accepted)

    SourceScreen=My screen#?

        matches any screen title that starts with "My screen".

    SourceScreen=   This screen title is long

        matches any screen whose title is
        "   This screen title is long"
        (Note the three leading spaces.)

  SourceScreen=FrontScreen

      VWM uses the frontmost screen’s dimensions when calculating
      the new position for the target window.

  SourceScreen=DefaultScreen

      VWM uses the default public screen’s (the Workbench screen
      by default) dimensions when calculating the new position
      for the target window.
```

If you use more than one of the above keywords in an object, only the last one will be considered as the valid one.

1.21 coordinate positioning

Coordinate Positioning

There are two ways of defining the coordinates of the new location for the target window, setting an absolute or relative position, using the following three keywords.

```
DestinationPosition=LeftEdge,TopEdge
DestinationFlag=Absolute
DestinationFlag=Relative
```

For the target window, you can define a target location by using the DestinationPosition= keyword. If you do not use one of the additional keywords, DestinationFlag=Absolute or DestinationFlag=Relative, the first one (absolute) will be assumed.

```
DestinationPosition=10,20
```

VWM opens the target window in the position 10 (LeftEdge) and 20 (TopEdge). As none of the additional keywords are given the absolute location is assumed. That is the window is opened at the absolute location 10x20.

So...

```
DestinationPosition=10,20
```

and

```
DestinationPosition=10,20
DestinationFlag=Absolute
```

both function in exactly the same way.

If you run a virtual screen, you can use the DestinationFlag=Relative keyword instead. VWM will then open the target window in absolute positions 10,20 but related to the point (0x0) of the visible part of the screen.

To understand the usage for this powerful keyword, let’s assume that you use a 1024x1536 virtual Workbench screen that consists of two 1024x768 visible parts. Furthermore, let’s call the first area (0x0 - 1024x768) Area1 and the second area (0x769 - 1024x1536) Area2.

```
DestinationPosition=10,20
DestinationFlag=Relative
```

VWM opens the target window in location 10,20 if you are in Area1 and in location 10,788 if you are in Area2. The visible effect of this is that the target window becomes located at the same visual location no matter which part of the virtual screen you are using :)

This is very useful if there are programs that you want to open in fixed positions in both areas. Just try it, and you'll wonder how you could live without it ;)

1.22 preset positioning techniques

Preset Positioning Techniques

VWM can furthermore open windows using one of four preset positioning techniques that are usually related to the mouse pointer's position rather than to the absolute or relative points 0,0 in the screen:

DestinationFlag=CenterWindow

VWM opens the target window in the center of the visible part of the screen.

DestinationFlag=FollowMouse

VWM opens/centers the target window under the mouse pointer.

DestinationFlag=RespectView

VWM opens the target window under the mouse pointer but will respect the view, that is to locate the window far enough away from the closest edges of the visible screen but still under the mouse pointer.

FollowView(TM), known as RespectView in this program, is Copyright ® 1993-1997 Bahman Moallem and was first introduced in MagicWords, the multilingual word translator for the Amiga back in 1993.

DestinationFlag=UseCorners

VWM opens the target window in the corner of the screen that is farthest away from the mouse pointer.

If more than one of the above is used, only the last entry will be considered valid.

1.23 example prefs file

Example prefs file

Here is a sample of part of the file ENV:VWM.prefs in my system:

```
; --- Open my calculator under the mouse pointer but respect my view
SourceTitle=Calc
DestinationFlag=RespectView
#
; --- Always open my file manager in relative position 300,13
SourceTitle=FManager
DestinationPosition=300,13
DestinationFlag=Relative
#
; --- Give my editor's "Find & Replace" window, virtual screen support
SourceTitle=Enter search/replace text.
DestinationFlag=FollowMouse
#
; Center my CD-players's dragbar-less window
SourcePosition=0,10
SourceDimensions=308,240
DestinationFlag=CenterWindow
#
```

Simple, isn't it? In this example,

```
; Center my CD-players's dragbar-less window
SourcePosition=0,10
SourceDimensions=308,240
DestinationFlag=CenterWindow
#
```

is what I've been referring to as an object. Notice the pound sign (#) at the end of each object. Also notice that each object has at least one source definition and at least one destination definition as mentioned in the previous section.

Remember that some of the definitions can be used together, but others cannot. If definitions that cannot be used together are entered, the last entry will be considered the valid one.

1.24 faq

FAQ - Frequently Asked Questions

Q: It's annoying to have to guess at the coordinates when I'm writing my prefs file. Is there any way I can put the window where I want it and then find out those coordinates?
A: Put the window where you want it and run a system monitor like Scout, XOper, or ARTM. These monitors can tell you the titles, positions, and dimensions of all the open windows.

Q: Why doesn't VWM understand the SourceTitle= definition for a window that obviously has a title?
A: The title of the window was not defined in its structure when VWM analyzed it. Obviously the program uses the SetWindowTitles() function of intuition.library to change the title of the window after launching it. Use one or both of the SourcePosition= or SourceDimensions= keywords for such windows.

Q: Isn’t there any other/better solution to solve the above problem?

A: Currently, no. As I don’t have any problem with the above limitation I have decided to not patch yet another function of intuition, but if a few users ask for it, it is not impossible that the next version of VWM will do just that.

Q: Why does my calculator opens its window under the mouse pointer?

The relative part of VWM.prefs in my system looks like this:

```
; --- Open my calculator window at position 10,10
SourceTitle=Calc
DestinationPosition=10,10
DestinationFlag=FollowMouse
#
```

A: If you use multiple keywords of the same type (destination definition in your prefs file above) VWM will pick up the last one of them that, in your case, is obviously the DestinationFlag=FollowMouse. :) The only time this doesn’t apply is when using DestinationFlag=Absolute/Relative and SourcePosition/SourceDimensions combinations.

Q: Ok, I want to start using virtual screens, but all the virtual screen managers that I tried have GUIs that are kinda in the way.

A: I had the same problem, so I wrote VSM, the virtual Screen Manager, which is a GUI-less virtual screen manager with AREXX support. VSM is available for free download at my WEB site.

Q: Why is a configuration program with a GUI missing in this package?

A: I’d rather edit VWM’s prefs file in my text editor. I am not saying that a prefs program with a handy GUI will never show up, but I don’t give that task a high priority. Sorry. If anyone else wants to create such a program, it is ok with me.

1.25 history

History

19950107 - version 0.1
VWM was born.

[pre-release history deleted]

19970315 - version 1.2
First version released to the public 8^).

1.26 copyright notices

Copyright notices

Amiga is a registered trademark of Amiga Technologies.

VWM version 1.2 Copyright © 1995-1997 Bahman Moallem.
VWM.doc version 1.2 Copyright © 1997 Bahman Moallem.
VWM.guide version 1.2 Copyright © 1997 Bahman Moallem.

FollowView(TM), known as RespectView in this program, is Copyright ® 1993-1997 Bahman Moallem and was first introduced in MagicWords, the multilingual word translator for the Amiga back in 1993.

All program names mentioned in this document are either Copyright or Trademark of their respective owner(s).

1.27 special thanks to

Special thanks to

- Jonas Edlund for beta-testing.

- Matthew Warren for beta-testing.

- Nick Curcio for beta-testing and for helping me to make this document as good as it is 8^)

- Peter Bergström for beta-testing and for translating this manual to Swedish.

1.28 contacting the author

Contacting the author

If you find some bugs or have some constructive ideas on how to improve VWM, please contact the author via Internet mail at:

    bamo@ctools.pp.se

The latest version of VWM is always available at the author’s WWW site:

    http://www.ctools.pp.se/