

AmigaMail

COLLABORATORS

	<i>TITLE :</i> AmigaMail		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 19, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AmigaMail	1
1.1	IV-17: Opening Screens and Windows on Any Amiga	1
1.2	Opening Full Size Screens	1
1.3	PAL or NTSC?	2
1.4	Opening Full Sized Windows	3

Chapter 1

AmigaMail

1.1 IV-17: Opening Screens and Windows on Any Amiga

by Ewout Walraven

The release 2.0 intuition.library has a multitude of new display possibilities at its disposal that developers can use to improve their software. Many products can benefit from the flexibility that higher display resolutions and larger display areas offer and from the polished user interface available with the "New Look" in release 2.0. Because backward compatibility with 1.3 is important to many developers, these new features should be used in a way that will work with older systems.

This article shows how to take advantage of the display resources available to Intuition under any Amiga system. This article assumes the reader has some knowledge of release 2.0 (V36) screens and windows. For more information, see the article "An Introduction to V36 Screens and Windows" from the September/October 1990 issue of Amiga Mail.

Opening Full Size Screens
PAL or NTSC?
Opening Full Sized Windows

1.2 Opening Full Size Screens

Under pre-V36 versions of the operating system, a method commonly used to open a screen according to user preference was to "clone" the Workbench screen (to create a screen with the same resolution and view mode as the Workbench screen). Normally, applications called the Intuition function `GetScreenData()` to learn the Workbench screen's resolution and display mode.

Programmers should no longer rely on `GetScreenData()` to return the actual resolution and display mode under release 2.0. When asked about the Workbench screen, the V2.0 `GetScreenData()` returns the video

mode of a Hires screen, or, if the screen is interlaced, a Hires interlaced screen. The dimensions `GetScreenData()` returns will be the lesser of either the `OSCAN_TEXT` dimensions or the actual Workbench screen dimensions. This change to `GetScreenData()` prevents some programs developed for 1.3 from opening full size screens and windows under 2.0. The change had to be made to avoid confusing programs that couldn't handle the higher resolutions and new display modes that are available in release 2.0. For example, when `GetScreenData()` is called on a system using a SuperHires interlaced Workbench screen, it returns the dimensions and view mode of a Hires interlaced screen. If `GetScreenData()` had supplied the actual resolution instead, a program that assumed the resolution could not be greater than 640x400 could be severely crippled. The exception to this rule is a system running in one of the A2024 (or Hedley Hires) modes. To remain compatible with the V35 version of `GetScreenData()`, the release 2.0 version returns the correct dimensions of a screen in A2024 mode.

Another method previously used to get information about the Workbench screen was to look at the `GfxBase->NormalDisplayRows` and `GfxBase->NormalDisplayColumns` fields. As with `GetScreenData()`, these fields contain the dimensions of a text overscan Hires (or Hires interlace) screen. Obviously, these fields should no longer be used to obtain the actual dimensions of the Workbench screen.

To clone the Workbench screen under 2.0, lock the Workbench screen, get the screen's display mode ID, and get the necessary display information. The article "An Introduction to V36 Screens and Windows" from the September/October 1990 issue of Amiga Mail contains the example `CloneWB.c` which illustrates cloning the Workbench screen under release 2.0. Any application that wants to clone the Workbench screen should be prepared to handle any type of screen, because the user can change the Workbench screen to any type of screen they desire.

1.3 PAL or NTSC?

Some programs need to determine if the Workbench screen is in PAL or NTSC mode. Before release 2.0, it was impossible for the system to switch between NTSC and PAL. The PAL/NTSC state in which the machine booted dictated the mode of its displays. Pre-V36 systems could determine the PAL/NTSC state of the machine by examining a bit in the `GfxBase->DisplayFlags` field, which is set at boot time under all versions of the OS.

Thanks to the ECS and the new system software, PAL and NTSC display resolutions can coexist, which obsoletes examining `GfxBase->DisplayFlags` to determine the PAL/NTSC state of a particular system. Release 2.0 ignores the PAL/NTSC flag after it sets the flag at boot time. This means that under 2.0, if the system boots as NTSC, it is not possible to open a PAL screen in a way that will work correctly with release 1.3.

To respect the user setup of the Workbench rather than the default video mode, a program should use the `V36.graphics.library` and `intuition.library` functions to determine the display mode ID of the Workbench screen. Using that ID, an application can open a screen or

find more information about the properties of the Workbench screen. The extscreen.c example at the end of this article shows how to check for a PAL mode screen under any version of the operating system.

1.4 Opening Full Sized Windows

Because there is no guarantee that `GetScreenData()` will return the correct screen resolution, programs can't use it to obtain the dimensions needed to open a full sized window. To get the correct screen resolution of the Workbench screen, programs have to examine the `Height` and `Width` fields of the `Screen` structure returned by `LockPubScreen()`.

To make supporting new and old versions of the operating system easier, Intuition V36 offers a way to open screens and windows that is compatible with previous versions of the Amiga OS. Instead of requiring the use of new functions to open V36 specific screens and windows, the `OpenWindow()` and `OpenScreen()` functions each accept an extended structure, `ExtNewWindow` and `ExtNewScreen`, respectively. These structures allow the programmer to pass tags to the V36 `OpenScreen()` and `OpenWindow()` functions while remaining compatible with older versions of these functions. Older versions of Intuition ignore the excess baggage at the end of the structure where the tags are kept. These tags are partially used in V35 (a 1.3 release) to support the A2024 modes. Note that the `ExtNewScreen` and `ExtNewWindow` structures may be extended in the future, so programs must not assume their size is static.

To use the "New Look", pass the `SA_Pens` tag and a pen array in `ExtNewScreen`. Using the New Look requires a little bit more responsibility from an application. Programs that aren't careful can experience problems with the layout of graphics and gadgets in New Look windows and screens. For example, a lot of programs made assumptions about the height of a window's title bar. Under 1.3, this didn't present much of a problem because the title bar height didn't normally change. Under 2.0, the user can choose system fonts, varying the size of the title bar. Graphics and gadgets rendered into the application's window can write over the title bar if the application isn't careful.

The example at the end of this article, `extscreen.c`, illustrates how to open a screen and window under release 2.0, using 2.0 specific features if they are available, while staying compatible with older versions of the operating system. This example will open a screen in A2024 mode (under V37, V36, or V35) if the resources are available.