

**AmigaMail**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> AmigaMail		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 23, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>AmigaMail</b>	<b>1</b>
1.1	IV-1: Obtaining an Icon Image . . . . .	1

## Chapter 1

# AmigaMail

### 1.1 IV-1: Obtaining an Icon Image

By David N. Junod

One of the many improvements made to Workbench for release 2.0 lies in its ability to supply an application with arguments from Workbench. Prior to release 2.0, the only Workbench supported way an application could receive Workbench arguments was through the Workbench startup message. Now, Workbench arguments can come from many sources including AppWindows, AppIcons, and the ASL file requester. These enhancements have made processing Workbench arguments more beneficial than ever.

When an application is started via Workbench, the application receives a startup message (struct WBStartup) from Workbench. This message contains a pointer to an array of WBArg structures. The array of WBArgs corresponds to all Workbench icons that were selected at the time the application was launched. Each WBArg structure contains a file lock on the file or directory associated with each icon, as well as the name associated with the icon. A WBArg supplies the link to the icon, or the DiskObject, itself, from which necessary information, such as application or project ToolTypes, can be extracted.

Under release 2.0, Workbench supplies WBArgs using several different structures. The structure used depends on the source of the WBArgs. When it launches an application, Workbench sends the application WBArgs through the WBStartup structure. AppWindows and AppIcons use the AppMessage structure to send WBArgs. The ASL file requester provides the arguments in the FileRequester structure.

Also with release 2.0, it is possible that icons passed to an application do not have an .info file associated with them. Workbench can now display all files and directories iconically, not just those with .info files. For icons without .info files, Workbench supplies a default icon based on the protection bits and object type (file, dir, etc.) so the user can manipulate files and directories from Workbench.

The following routine, `wbarg.c`, uses `GetDiskObjectNew()` which should be used only when an application requires an image for every Workbench argument, whether actual or default. To only read actual icons and

tooltypes, modify the routine to use `GetDiskObject()` instead. For more information, see the release 2.0 Autodocs.