



Supplement for Directory Opus 5 Magellan Version

This manual has been designed as a supplement for the Opus 5.5 manual to describe the changes and new features of the Directory Opus 5 Magellan version 5.61.

If you have purchased a new Opus package, you should have obtained BOTH the original Opus 5.5 manual, this supplement and the Opus 5.5 and Magellan disks.

If you have purchased the upgrade only, you should have obtained the Magellan supplementary manual plus one disk.

Installation

If you are installing this version as an upgrade on Opus 5.5 then make a note of your Opus 5.5 registration number before you install this new version. (The number is in the *About* requester if you have forgotten!) . You will need this to complete your Opus Magellan installation.

Installation of this version is straight forward. Open the disk window, double-click the ***Install_Opus_Magellan*** icon and follow the directions. When the installation is complete, reboot your Amiga.

Since all program components of this new Magellan version have been updated, once you have finished the installation and rebooted your Amiga, you will need to re-enter your registration details from your original Opus 5.5.

 ***The Magellan update version is designed to be installed over a previous Opus 5.5 ONLY. You MUST have already installed Opus 5.5 before installing the Magellan upgrade.***

Introduction

Welcome to the Directory Opus Magellan upgrade!

From its first release in early '95, many Amiga users could see the raw power provided by Opus 5 with its unique concept integrating Workbench Replacement Mode (tm), file management and custom functions in the one package. These concepts were further refined in the Opus 5.5 release in '96. During this time of great uncertainty for the Amiga platform as a whole, we have been buoyed by the support of Opus users, many of whom have given us continual encouragement and much valuable feedback and suggestions on future directions for Opus 5. After a long struggle we are also pleased to have been able to convert a number of *Neo- Luddite* journalists from superfluous glitz to dedicated followers of fashion. (The wine's in the mail, guys ;-)

Through your support, Opus 5 has now become firmly established as the premier workbench replacement and file management program for the Amiga.

Thanks!

With this new version we have attempted to return your loyalty with ongoing support for the Amiga platform and a continuation of our development plan for Opus 5. We have also tried to add as many of your user suggestions as possible which are consistent with the metaphor of Opus.

We hope you will enjoy the advanced features in this new version which we call the ***Directory Opus 5 Magellan*** version.

Protect your investment! Software piracy will kill the remaining Amiga market.

Remember. Pass the word, not the disk!

Summary of Enhancements

The Magellan version is not just a simple upgrade but involves extensive redevelopment of many components of the Opus 5 system to give you enhanced usability, especially in the Workbench Replacement Mode. We give you greater control over icons, actions on and with icons, plus many new features such as Start Menus, extra popup *sticky* menus and new background images to name but a few. After extensive testing from our Beta testers we have also improved the compatibility with Workbench and other Amiga programs with in-built support for the NewIcons system, better Cybergraphics support, faster icon layout, dragging and display routines and so on. A rewritten OpusFTP module complements these changes and not only gives you up to five times faster access to the Internet but now has a full GUI for the Address Book and operational parameters.

The new features are discussed in detail on the subsequent pages. A summary of the changes follows:-

- ***Enhanced Lister functionality*** including:- command functions can now act on icon mode listers; new ***popup menus***; optional ***Space Gauge*** showing free space on drives, ***Inline Editing*** for Name mode listers and the ability to ***drag and drop files directly into sub-directories***.
- A new integrated system of ***Start Menus***.
- New ***Desktop Folder*** mechanism: Drag and drop items to the desktop with the support of optional popup menus.
- New ***Icon Positioning*** system. Configure areas where icons will appear and their priority.
- Global control of *icon labels* and *icon borders*.
- Direct *byte-for-byte copy* of icons.

- **Icon label splitting** for long labels.
- New **Icon Command** function for special "command" files.
- New **Icon Information** requester with extensive popup options.
- Faster **icon dragging** routines.
- Use traditional Workbench **icon positions** for icons or exclusive Opus positioning.
- New **Popup menus** with shadow look.
- **Environment GUI Settings:** New features gives greater user control over Opus configuration options.
- **Enhanced Opus FTP** including:- new **GUI Address book**; new GUI for configuration system; up to **5 times faster** access with **less memory usage**; individual **directory cache** exclusive to each FTP lister; optional **idle timer** and **auto index download** for AmiNet index files.
- **Cybergraphics** support for dragging icons and general display speed with ability to now have full 24 bit backdrops using the V43 datatype, rather than a dithered 256 colour image.
- Integrated **NewIcons** support not just for icons but for all system images including graphic button banks.
- Improved **compatibility** with **MUI**, **DataTypes**, **MCP** and a number of other programs.
- More **Environment Variables:** New environment variables to allow you to customise Opus functionality even more.

What's New

- **Text Viewer:** Rewritten for faster access and better scrolling especially on Cybergraphics screens.
- **FileTypes:** New filetype matching commands and parameters including ability to match on disks. **New User actions** to the filetype editor. There are six new User actions (along with User1-4 making a total of ten), and qualified double-click and drag&drop actions.
- **Scripts:** New system of disk inserted/disk notification gives better control over new disks or newly mounted remote file systems.
- New **Command Functions** plus enhanced **Argument Variables**.
- **Buttons/Images:** Full NewIcon image support plus more support for image and animation file formats including DPaintV/PPaint AnimBrushes.
- **Groups:** Can now contain icon-less items plus you may now add left-out commands to groups.
- **New Arexx commands.**
- More than 40 **new callback hooks** for easier user programming of Opus provides in the new Opus Software Developers Kit.

General Changes

Cybergraphics Support and Graphics Cards

We have improved the support for Cybergraphics systems where the bitmap is non-standard. This provides greater display speed on Cybergraphics and other chunky screens.

On a Cybergraphics 16-bit and higher screens, the lister field titles are now visible when they are dragged.

Opus now allocates the bitmaps for backdrop pictures as *friend* bitmaps of the screen. On machines with graphics cards this gives faster refreshing of the screen and the use of fast memory instead of chip memory. Note that this works with both the V43 24-bit datatype and the old picture.datatype. You can now have full 24 bit backdrops using the V43 datatype, rather than a dithered 256 colour image.

The new set of custom icon dragging routines for Cybergraphics display modes give much faster Icon dragging. Although these new routines work well for Cybergraphics software, we have had some reports of problems with the (older) Picasso software so they are bypassed automatically if the display is not a Cybergraphics one. They can be disabled completely if required by setting the **NoCustomDrag** environment variable to 1.

Screen to Front

You can now specify the **NOSCREENTOFRONT** parameter on the command line or in the icon tooltip when Opus is started. This tells Opus to not move its screen to the front of the display when it initialises.

Fixes for OS2.0 systems

Under OS37 Opus 5.5 had problems with backdrop patterns which could result in a crash. This is now fixed.

File Listers

New Popup Menus

Popup Menus now have a new shadow look while the basic functionality has been enhanced by additions to the various menus. Additions to the filetype-specific RMB 'sticky' popup menus include:-



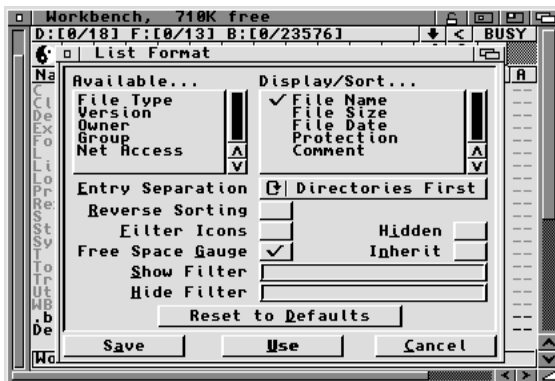
- **Copy to Desktop** in all modes. (See Opus Desktop below.)
- **Open in New Window** in name mode for directories which opens a new Lister for the selected directory.

- **Clean Up** to lister popup in Icon modes
- Project files have a new **Open With..** in the file popup menu, which lets you select an application to open the file with. Opus remembers up to the last 10 programs chosen, and displays them in the menu for you to select from.

You may now control how quickly popup menu sub-menus open by setting the environment variable **PopUpDelay** to a value for the delay in 50ths of a second. The default is 10 or about a 5th of a second.

New Lister Format Editor Features

A new **Free Space Gauge** option adds a fuel-gauge to listers, showing the proportion of space free on the disk. The colour for the 'bar' part of the gauge may be set from the Environment/*Lister Colours* section. The display of the gauge can be controlled on an individual lister or path basis from the **Format Editor**, or may be added as a global from the global format settings in *Environment/Lister Default..*



The advantage of this system is that it truly lets you configure which listers you want the gauge to appear for (eg, if you only want it on the root directory of certain devices, turn it off in the default lister format and then turn it on for those directories).

Gauges are turned off by default for custom handler listers; if you specify the *gauge* keyword for the *lister set <handle> handler rexx* command you can enable them but remember that the user may still have turned them off in the environment.

Also, a new **Inherit** option has been added. This flag has the same effect as the padlock gadget; if turned on, sub-directories will inherit the format of the parent directory (when double-clicking on them in name mode). When turned off, unless a directory has a format defined for it, it will use the default format

Icon & Icon Action Mode Listers

In Opus 5.5, the **Filter Icons** flag from the Lister Format applied when you switched to one of the Icon modes. An unfortunate result was that you did not see any icons! This flag is now ignored in the Icon modes so all icons are now displayed.

Drag and drop in icon mode now ignores filetype settings and always copies (or moves) the dropped items.

Icon layout routines have been improved plus splitting of long icon names has been added. (See Environment)

Listers now automatically resize themselves when switching between icon and icon action mode.

Name Mode Listers

Double-clicking on a program in Name mode now runs it as an AmigaDOS program, with arguments, even if the program has an icon. Opus 5.5 would automatically run the program as a workbench process if there was an associated icon.

However, you may still run a program with an icon as a Workbench process from a name mode lister if you hold down the *shift* key when you *double-click* on it.

You can now drop .info files from a name mode lister onto the IconInfo requester to replace the icon image.

Entering **FTP://<arguments>** in the lister path field will launch Opus FTP in the current Lister. (See the OpusFTP section.)

Drag and drop files directly into a sub-directory

When enabled from *Environment/Listener Options*, you may drag and drop files directly into a sub-directory in Name mode. Moving the mouse when dragging files over a directory in a lister (either the same lister or a different one) will highlight the directory name. If you release the files over that directory, the defined filetype action will use that sub-directory as the destination path rather than the path of the lister itself. If you have this option switched off, you can always *activate it by holding down the shift key while dragging*.

Dropping onto sub-directories is disabled by default for custom handlers. You need to add the '*subdrop*' keyword to the *lister set handler* command to enable dropping into sub-directories for your handler. Note that the user may still have sub-dropping turned off in the environment.

For ARexx users, the "*drop*" message now contains the full destination path in Arg5. You can compare this against the path of the destination lister (handle in Arg1) to see whether the drop was into a sub-directory or not.

Note that this is not yet implemented when dragging icons, only files/directories from a name-mode lister.

OpusFTP now also supports this function from a remote FTP directory to a local dir, but it is NOT supported on or between remote FTP directories.

NewIcons


NewIcons is an advanced GUI enhancement system for the Amiga which allows for a palette independent icon system. By design, icons are mapped to the correct colours on any system. It is widely available from AmiNet and from Internet sites or contact Eric Sauvageau via <http://www.thule.no/~merlin>

Opus Magellan inherently supports NewIcons and it has been extensively tested under NewIcons 3.1 and higher. Opus will use the newicon.library if it is present to remap the icons automatically. NewIcon images for all internal icons and images including AppIcons are also supported. (But see the note below.)

If you turn the **Cache Icon Images** flag on in *Environment/Icon Settings*, Opus will sense if an icon is a NewIcon and will automatically not cache it.

Because of the integrated support, NewIcons images and icons will now have the correct colours when Opus is running on its own custom screen as well as in WBR mode.


The IconInfo module has been enhanced with the ability to remove NewIcons image information from an icon. This is accessed from a RMB popup menu over the *Icon Information* requester.

 ***Unfortunately there is no mechanism to trap the addition of AppIcons with NewIcon images. Opus will automatically map its own AppIcons but if other applications are using AppIcons with NewIcon images, Opus cannot see them. You must ensure that you are running the NewIcons patch program (c:NewIcons) to give correct images. Opus does not need the NewIcons patch program for any other reason.***

DefIcons


The NewIcons system comes with a set of default icons and an associated simple filetype system known as DefIcons. This system was designed to work only with Workbench and so cannot be used directly with Opus. However, you can use the pretty DefIcon icon

images in Opus filetypes. Simply drag the icon image onto the *Default Icon* area in the filetypes editor.

 ***The icons supplied with the DefIcons package are project icons with an associated tool (a program). However, unlike normal project icons, which run the application as a Workbench process, the deficon system is designed to run the associated program as an AmigaDOS process. This can cause a crash is you use an original Deficon as a real icon with either Workbench or Opus. We have attempted to trap such situations but it is not possible to trap them all. Please check if the program defined as the tool for any Deficon is able to be run as a Workbench process. Some programs, such as Multiview, can be run either as a Workbench or AmigaDOS process. Some, such as Execute , Delete and other C: commands, will crash your Amiga if run as a Workbench process.***

Sample NewIcon Images for Opus

On the Opus Magellan installation disk we have included a directory named NewIcons. Here we have collected a range of PD NewIcons for Opus. You should examine these and install them manually if you wish to use them. Instructions are provided in the directory.

 ***Unfortunately, because of space constraints, we have been unable to fit as many of the available NewIcon images on the distribution disk as we would have wished. More images can be found from our web pages or from AmiNet.***

Start Menu

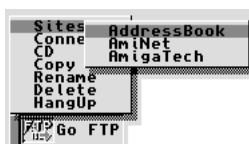
In Opus 5.5, it was actually possible to create a single button with a number of attached popup menus, although it was not easy and the feature was somewhat hidden.

We call these **Start Menus**. They provide a quick and convenient method of having a custom set of popup menus which may be loaded when required and which occupy only a minimal area of your main window or desktop. The visual footprint can be a simple text string or you may add your own custom image.

Opus Magellan provides direct access to these Start Menus. You may create a new Start Menu or load an existing Start Menu directly from the main *Buttons* menu. Once positioned on the screen and saved, these Start Menus become part of the environment as with all other Opus elements and are reloaded automatically with that environment.



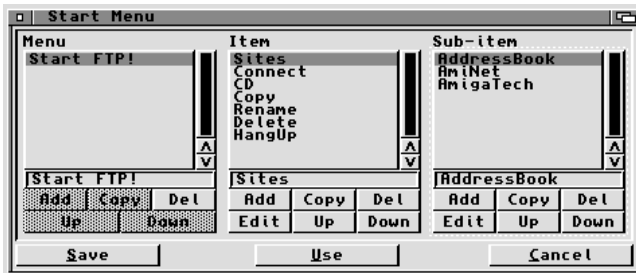
The following is a sample Start Menu which could be used to access the Opus FTP commands.



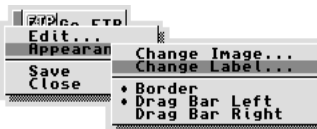
Start Menus are another example of an Opus button or menu object and are created and edited in a manner similar to user menus. Just as with other Opus button/menu objects, you can now

Start Menus

drop files and directories (!) onto a Start Menu to have these items added to the menu automatically. Dropping a directory onto a Start Menu is a quick way of creating a set of menus to access favourite directories! Try it! Note that you must save the Start Menu to make these changes permanent.



Once a Start Menu has been opened, you may edit the menu commands from a special popup menu (*the control menu*) accessed by clicking the right mouse button over the small drag bar which by default is the small area on the left of the Start Menu. From the *control menu* you can also edit the appearance of the Start Menu itself to change the name, image or drag bar orientation. For example, you can put the drag bar on the right-hand side of the Start Menu, which lets you position the Start Menu hard up against the right side of the screen.



A **Save** option in the *control menu* appears whenever anything has changed. This replaces the more usual Snapshot/Unsnapshot options for other Opus objects. If the Start Menu is repositioned, the **Save** option will be visible in the popup menu letting you save the new position. Unfortunately, your old snapshotted position will be lost.

Start Menus

Start Menus are saved in the Environment file (if you have *Save Layout* turned on), and open automatically when the Environment is loaded.

Shortcuts in Start Menus

You can toggle the drag bar on and off by holding either *shift key* and *clicking* on the Start Menu with the left button. If the drag bar is turned off you can access the control menu by pressing the *Ctrl key* and *clicking* on the Start Menu.

Desktop Folder

Opus Magellan introduces a new concept to the Amiga - that of a **Desktop Folder**. The Desktop Folder provides a useful way to access temporary files quickly, and a place to store files that you refer to often.

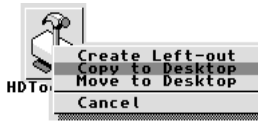
The Desktop Folder is a special directory, usually defined as **DOpus5:Desktop**. Until now, the main window of Opus (the Desktop) has been the area where drive icons, groups and left-out icons are displayed. With the addition of the Desktop Folder concept, any files that are located in the Desktop Folder are also displayed on the Opus main window or Desktop.

Desktop Folder icons displayed on the main window are different from traditional Amiga left-outs, because they are the actual file or directory itself, rather than just a reference to the file. Therefore, if you delete a file from the Desktop Folder, the actual file itself will be deleted. *This requires some caution.*

There are two main ways of copying files to the Desktop Folder.

- Use the **Copy To** function on the popup menu for files. Right-click a file, and move the mouse to the **Copy To** item. You will see a new option in the sub-menu - **to Desktop**. Selecting this will cause the file to be copied to the Desktop Folder. The new file will appear on the Opus main window as an icon.
- Drag a file (or directory) and drop it onto the main window. In the previous version of Opus, doing this would cause a temporary left-out to be created, and this is still the default action in Opus Magellan. However, in the *Environment / Desktop* section, there is a new flag called **PopUp Enabled**. If you turn this flag on, when you drop a file onto the desktop, a popup menu will appear giving you a choice of several actions. From this menu you can choose to create a left-out (like the original Amiga behaviour), or you can choose to copy or move the file to the desktop.

Desktop Folder



The *Environment/Desktop* section also contains a field which lets you modify the *location* of the Desktop Folder. We do not recommend that you change this location, but if you want the desktop folder located on a different drive (say, one with more space), you can modify it here.

There is also a **Default Action** setting. When the Desktop *popup* is *enabled*, the default action option allows you to choose the action to be performed by default when you drag a file to the desktop. If **None** is set (the default), then the popup menu appears as normal. However, if you set **Create Left-out**, **Move to Desktop** or **Copy to Desktop**, then that action will be performed without a popup menu appearing. With a default action set in this way, access to the popup menu is still possible by holding down either *shift* key when you *drop* the file onto the desktop.

Remember, any files or directories in the Desktop Folder are displayed on the Opus main window as icons. They are in all respects normal files, and can be dragged to other windows to copy/move them, or double-clicked on to run them. You can also press the right button over them to bring up the file popup menu, where you can rename or delete them, etc.

Icons

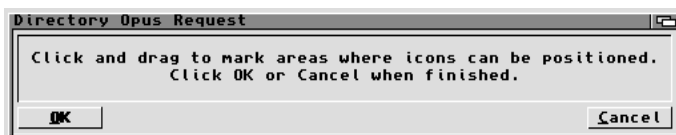
General

- It is now possible to turn off all icon borders globally from the *Environment/Icon Settings* requester. If set, all icons will be rendered without borders by default but you may still change these on an individual basis from the Icon Information requester with the RMB popup menu.
- Setting the environment variable **NoNoLabel** (yes, it is not a misprint it is NoNo..), will disable label-less icons. Some users have reported than in rare cases that their icons are showing up with no labels; obviously the special bit in the icon structure which Opus uses is accidentally set.
- The icon positioning algorithm has been extensively rewritten and positions new icons horizontally rather than vertically. As a bonus the routine is significantly quicker and smarter.
- Some users complained about being able to move icons accidentally when double clicking on them. This behaviour was traced to small movements in the mouse position between the double-click. We have now implemented an effective solution to the problem and Opus now handles this similarly to Workbench.
- Opus 5.5 copied icons using library calls from the Amiga icon.library of GetDiskObject() / PutDiskObject() to provide compatibility with some PD programs. Unfortunately this method had some unwanted artefacts. Opus Magellan now copies icons as normal using a straight byte copy. However, if you set the **SmartIconCopy** environment variable, it will copy icons with the older Get/PutDiskObject() method as before.

Icon Positioning

Several users asked for a method of deciding where new icons would appear on the main window. The new *Icon Positioning system* allows you to configure areas on the main window in which certain types of icons will appear, and the priority they will appear in.

From the *Settings* menu, choose **Icon Positioning** and Opus will enter a mode where you may create special **icon positioning areas** by clicking and dragging on the main window. These areas are represented by windows which can be resized, re-positioned and closed as normal. Each window has a RMB sticky popup menu to configure which types of icons will appear in that area. You can also select one of five priorities for the area, which determines in what order the positioning areas will be used. These areas are saved in the Environment file. Defined areas may optionally include Appicons, Disks, iconified Listers and Buttons, Groups and any Left-outs icons including items in the Desktop Folder.

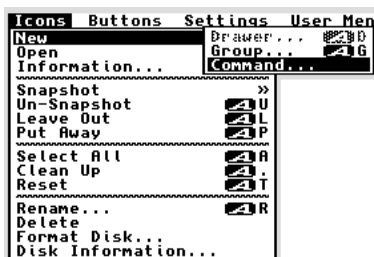


Splitting Long Icon Labels

If an icon's label is more than approximately 1.5 times the width of the icon itself, it can be now split onto multiple lines. (See Environment / Icon Settings.)

New Icon Command Function

A new menu item **Icons/New/Command** allows the creation of a *command* file, essentially a single Opus function in a file. The command is saved to the new *DOPus5:Commands* directory (which is created automatically the first time you run the Magellan version), and a left-out for it is automatically created on the main window.



Double-clicking a left-out command icon will run the command, just like clicking a button in a button bank. Edit the command file by right-clicking on it and selecting **Edit** from the popup menu.

To remove the left-out from the main window, choose **Put Away** from the RMB popup menu. This will not delete the command itself; you must delete it manually from the *DOPus5:Commands* directory if you want to get rid of it permanently.

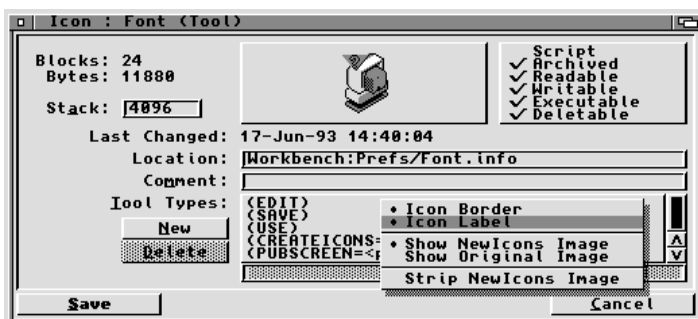
Left-out commands use the default icon *command.info* from *DOPus5:Icons* but you can give individual commands their own icons by just adding an icon (.info) to the file in the *DOPus5:Commands* directory.

Command files may also be added to a standard Opus Group.

Icon Information Requester

The Icon Information requester has been extended to show the full file path or *Location* of the actual file plus has an enhanced set of RMB popup menus. The menus allow you to adjust the attributes of the icon including the icon borders and NewIcon settings independent of the global Environment/Icon Settings. The **Show NewIcon Image** and **Show Original Image** allow you to toggle the display between the two images. Note that the old **No icon border** and **No icon label** options have been renamed to **Icon Border** and **Icon Label**, and their sense reversed.

Icon Information will show the correct filetype-defined icon for the file without an associated .info file where one has been defined in the filetypes default icon. (*Show All mode*).



☞ *If you use the DefIcons from the NewIcons package as default icons for Opus filetypes, be aware that many of these have an AmigaDOS only program as the associated tool. Double-clicking on such icons can cause a system crash by running this AmigaDOS program as a Workbench process. Opus attempts to trap some of these commands from fake icons such as "C:Execute" but you should check the default tool for safety.*

Other Icon Changes and Notes

- Icon borders are now one pixel smaller vertically, to match those used by Workbench.
- Holding *shift* and *double-clicking* a disk icon now forces a lister to open for that disk, even if one was already open.
- If you left out an icon and then renamed it, you would not be able to access the object until the next reset (and from then on renaming it would work fine).
- If you set the environment variable **QuickIconDrag** to 1, icons will no longer be masked when they are dragged. This results in an opaque background to the icon (like in Workbench) but is much quicker.

Environment

Several new sections have been added to the Environment section with new or enhanced requesters.

Backgrounds

Opus Magellan has the ability to display background pictures directly, without going through the WBPatten Prefs program.



Enable Backgrounds lets you turn on or off all background pictures in Opus. Underneath this flag are three fields which select a picture file for the Desktop (Opus main window), Windows (listers and groups) and Requesters. At the far right of each field is a gadget for a popup menu to control the display parameters including whether each picture is tiled (the default) or centred in the display, and the remapping used for each picture.

The remapping level controls how the picture is remapped to be displayed on your screen, and affects the number of pens used and the quality of the final result. The actual level to use should be determined by how many spare pens are available on your screen and the complexity of the actual image. For example, if using a 24-bit display, start with *Best* and experiment with the settings until you achieve the desired result.

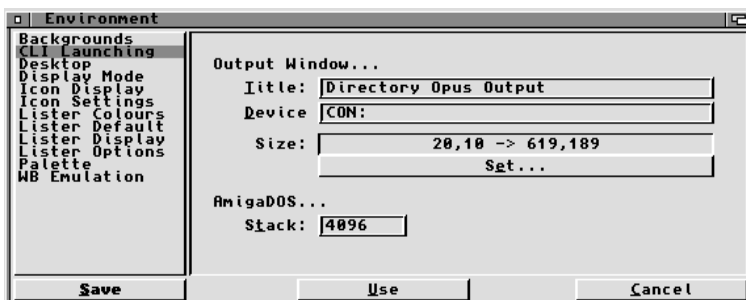
☞ *The Amiga Datatypes system is responsible for remapping the images so the results achieved may be dependent upon your datatypes settings.*

Use **WBPattern Prefs** enables the original behaviour of Opus. When enabled, Opus will use the WBPattern prefs file to set the picture or pattern displayed on the desktop or in listers and groups.

☞ *Under OS2.0 and 2.1, you can only use WBPattern prefs, and only to generate patterns, not pictures. The picture display system relies on the Amiga Datatypes system, which is not available under OS2.0.*

CLI Launching

We have reorganised and renamed the older *Output Window* section to *CLI Launching* and added a **Stack** setting. This allows you to set the default stack for AmigaDOS programs which DOpus executes (the default is 4000 bytes). Although poor programming practice, some Amiga programs will crash if run with a default 4K stack. In such cases you may need to set this to a higher value, say 10K.



Desktop

The Desktop section has been revised with the icon parts moved to their own section.



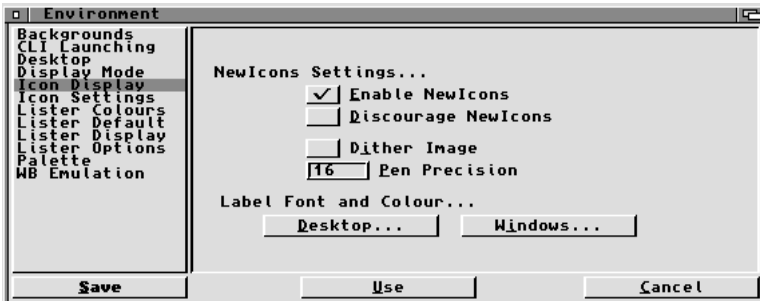
The operation of the **Hidden Drives** list has been changed and now doesn't discard the entries of devices that aren't present. This means that if you have a device which is not always mounted, its entry won't get lost (or reset) if you go into Environment without it mounted.

A new *cycle gadget* allows you to set both hidden disks and hidden bad disks so you may selectively hide devices that have bad disks in them. For example, if you had four filesystems mounted on the one drive, you could hide all of them for bad disks except for one.

Desktop Folder: This is the *physical location* of the drawer used by Opus to hold the desktop files. By default this is the directory *D0pus5:Desktop*. You may change this location but we do not recommend it since it establishes connections outside the parent *D0pus5:* directory tree. If the location is changed, you will have to move any current files in the old desktop drawer to the new location before they will be noticed. See the Opus Desktop section for a description of the other settings.

Icon Display

This new section allows configuration of the NewIcons settings within Opus plus the Font and Colour settings for all icons.



Currently available options include:-

Enable NewIcons- turn NewIcons support on or off.

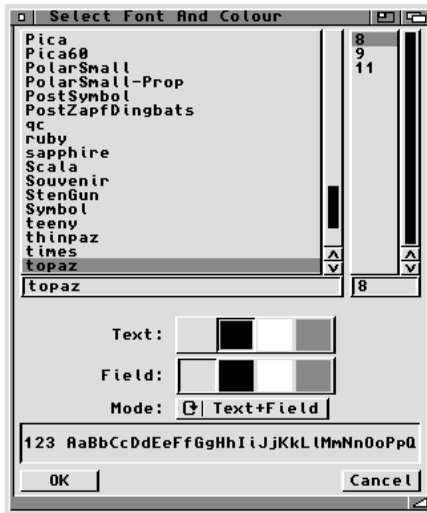
Discourage NewIcons - only shows a NewIcon image if there is no 'standard' image (i.e if the image is less than 5x5 in size)

Dither Image - turns on dithering for NewIcons.

Pen Precision - sets the precision for NewIcons.

The *Dither Image* and *Pen Precision* settings affect the operation of the NewIcons system itself. It was necessary for Opus to have these settings since version 3 of the newicons.library does not load the user-defined NewIcons preferences (unless the c:\NewIcons patch is running.) Version 4 of the newicons.library reads the user preferences automatically, however you can still adjust the settings through this Opus section as well as through the NewIcons Prefs.

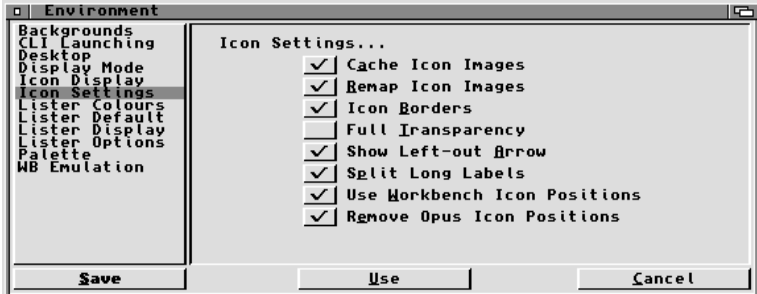
☞ *If a change is made to any of these settings you will need to restart Opus for the change to become effective.*



Desktop allows you to edit the font settings for icons on the desktop and **Windows** allows you to edit the font settings for icons in lists/groups.

Icon Settings

This new section collects and controls aspects of Opus's icon display routines. Some settings are also new.



Cache icon images and **Remap icons** are now a 'positive' flags.

Icon Borders turns off **ALL** icon borders by default but can be overridden on an individual icon basis from the Icon Information requester.

Full Transparency makes Opus render its borderless icons with colour 0 transparent over the whole icon, rather than just colour 0 around the edge. This results in quite a large speed increase when loading borderless icons.

Show Left-out Arrow controls the little arrow shown on the bottom left of left-out icons.

Split Long Labels. If an icon's label is more than 1.5 times the width of the icon itself, it can be split onto multiple lines. The algorithm will only split labels on spaces, punctuation characters or on a capital letter. If there is nowhere to split the text then the label won't be split at all. This behaviour can be modified and Opus can be forced to split the label at the 150% point by setting the **LabelForceSplit** environment variable to 1.

Use Workbench Positions tells Opus to use the same fields in icons as the Workbench does to store and retrieve icon and window positioning information. If you snapshot icons on a disk and give the disk to an unfortunate Opus-less Amiga user, they will still be able to view your icons in the correct positions with Workbench.

Remove Opus Icon Positions. Theoretically, setting this option along with the **Use Workbench Icon Positions** flag enables you to move an older Opus-used system back to a Workbench position system without having to resnapshot all your icons.

With the 'Remove...' flag set, Opus will use the Opus icon position set in the icons if there is one. When the icon is snapshot, the Opus position will be removed, and the Workbench position will be saved. Next time the icon is read, it's the Workbench position that will be used.

Lister Colours

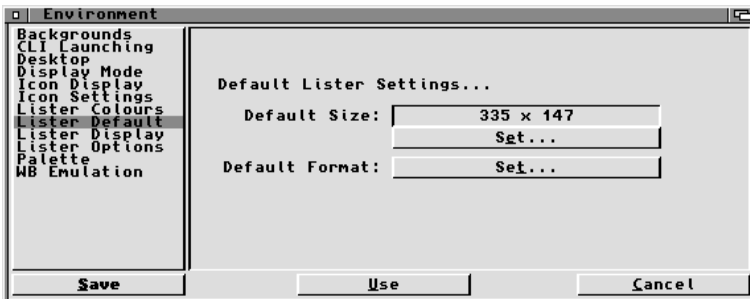
Settings for Lister colour adjustment have been collected in their own section. The colour settings now have an additional custom pen for each of the settings. This pen is completely separate from the standard user/system pens. It can be configured individually for each element, providing there are free pens available. At the moment, the custom pen is only implemented by the free space gauge, but the other elements will be using it in the next version.



☞ When setting the colours for the free space gauge, the foreground colour is used when the bar shows less than 90% full, and the background colour is used when the bar is at 90% or above. This lets you have a warning colour when the disk is getting full!

Lister Default

Lister default allows you to set the **Default size** and **Default format** for Name Mode listers.



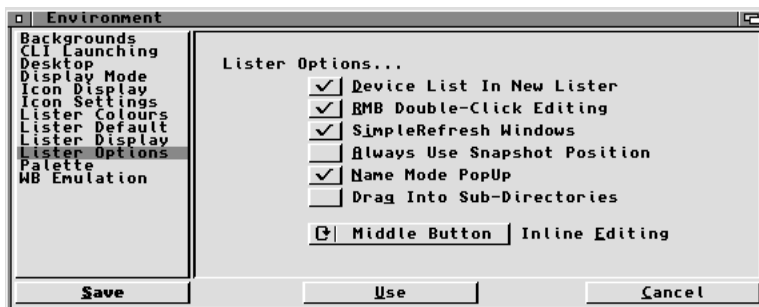
Lister Display

Lister Display now controls the **Default Lister font** and **lister status bar text display**.



Lister Options

These settings collect special options for Lister displays. A new setting has been added for **Inline Editing**. It replaces the old *ListerEditing* environment variable and enables inline editing of the name and details in name mode listers. This lets you perform the equivalent of Rename, Comment, Date and Protect functions directly in the lister without having to bring up a requester.



The setting of the *cycle gadget* controls how the inline editing mode is activated. **Left Button** means that you must hold the left mouse button over the entry for a certain time before editing mode is activated. Likewise, **Middle Button** means that you must hold the middle mouse button down. **Left & Middle** means that both buttons work.

When in *Inline Editing mode*, TAB moves to the next field and SHIFT-TAB moves to the previous one. Press RETURN to save the changed you have made, or ESCape to cancel.

WB Emulation

The old *Display Options* section has been renamed *WB Emulation*, since the backdrop picture settings are now in their own section.



The new setting, **Move AppIcons to Tools Menu** allows you to redirect AppIcons to menu items in the Tools Menu. Note that changing the flag only affects AppIcons added after the change is made.

Changes to Environment Variables

The following additions to the ENV variables (ENV:Dopus / <variable>) allow fine tuning of the Opus interface and command behaviour. Generally these variables are only read when Opus is started. If you wish to change a settings you will need to quit and restart Opus for these variables to be re-read.

NoNoLabel: This will disable label-less icons. Some users have been reporting that their icons are showing up with no labels. Obviously the special bit in the icon structure used by Opus uses is accidentally set.

PopUpDelay: To control the delay for popup menus set this to a value measured in 50ths of a second. The default is 10 or about a 5th of a second.

WorkbenchTitle: When set to 1, Opus will not generate its own clock/memory display in the screen title bar. Instead, it will set the screen title to *Amiga Workbench*. This lets it be intercepted by programs such as MCP, which allows you to configure the title bar display.

QuickIconDrag: If set to 1, icons will no longer be masked when they are dragged. This results in an opaque background to the icon (like in Workbench) but is much quicker.

NoCustomDrag: Set to 1 to disable the new RTG custom drag routines completely.

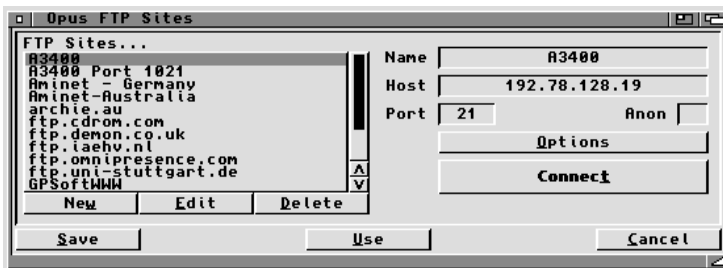
HidePadlock: If set to 1, the padlock titlebar gadget will not be added to listers.

OpusFTP

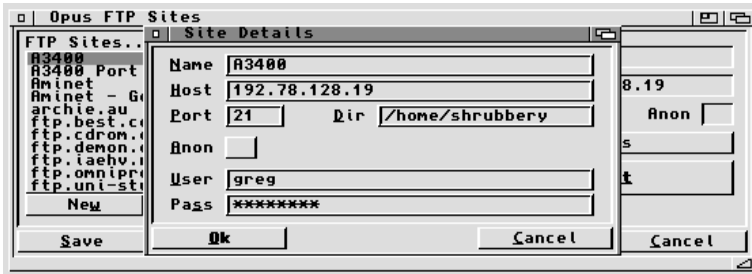
The FTP module has been extensively rewritten. Improved buffering to the TCP sockets plus more efficient memory usage provides faster listings of remote directories and up to 5 times faster file transfers. A **PORT** command has been added so you may connect to remote sites at other than the standard FTP port 21. Transfers directly between remote sites are now also possible. Plus, the module has a new GUI interface for the ftp configuration file (from *DOpus5:system/ftp.config*) giving the ability to edit your site list and the global FTP operational parameters directly.

FTP directories now have an cache exclusive to each FTP Lister available via the *CacheList* command as with normal Opus listers. (In fact, all custom handlers now have this ability.) Because these caches are handled by Opus callbacks directly, they are much faster and effectively displayed at the same speed as local cached directories.

The command **FTPAddressbook** now displays the following requester:-



Double click on an entry to connect or highlight the entry and select **Edit** or **Delete**. Select **New** to add new entries.



Select **Config** to show the configuration sections:-



Some new fields have been added. You may choose a specific password to be used for anonymous connections and easily edit the logfile command. Other options are:-

Log: Turns the log display on or off to display messages from the remote site.

Debug: Enables verbose log display.


Update: Defines the delay between updating display of entries in a directory list. A higher value updates the display less often resulting in faster overall performance.

Timeout: The default timeout for FTP commands. (The default value is 60 secs.)

NOOP: This provides an idle timer mechanism to keep remote sites active. It sends the FTP command NOOP approximately every 30 seconds to prevent the remote site's inactivity timer from being triggered. *By default this should be turned off. Do not activate this unnecessarily. Keeping your connection alive when you are not actually using it often prevents other users from accessing the site. Some ISPs frown on such behaviour.*

Index, Auto & Max Size: Many sites, especially AmiNet, add a special index or comment file to each directory. This provides a short comment with extra information on each file. With **Index** turned on, the FTP module will check for the presence of files called 'INDEX' or 'Files.BBS' when scanning a new remote directory. If found, the index file will be automatically downloaded and the comments added to the file's Comment field. Generally these index files are small, less than 30Kb. If the **Auto** switch is off, files smaller than the **Max Size** value will be automatically downloaded. If the size of the index file exceeds this value, a requester will be presented asking if you wish to download the index.

With **Auto** turned on, only files smaller than the **Max Size** will be downloaded. If the index file is larger than this value it will be ignored.

 ***Most of the index files on AmiNet sites are in a defined standard format. However this cannot be guaranteed and some files may display truncated comments.***

FTP Between Sites

OpusFTP now supports the transfer of files between two remote FTP sites directly *without going through your local machine*. Copy commands and Drag and Drop actions work as per normal listers. An FTP control connection is opened to both sites and the remote machines are told to transfer the files directly between each other. (Some sites may disallow passive transfers such as this.)

Unfortunately because no progress reports are available until the remote sites indicate they have finished the tasks a true file

progress indicator is not available. Instead, the progress bar toggles between one third and two thirds to show activity.

It is possible to abort transfers between sites but this can be problematic. If an FTP-FTP transfer is aborted, it may be necessary to close the connections completely and re-establish them.

Direct Access to FTP from a Lister

If you type "**FTP://**" in a lister path field, Opus will launch the FTP module directly in this lister. The rest of the template for action is the same as FTPConnect.

FTP://<HOST,PORT/N,USER,PASSWORD=PASS,DIR/K>

For example, entering

FTP://livewire.com.au DIR=/pub/aminet

will attempt to connect to the Australian Aminet site in the current lister.

Further Notes of the FTP Module

- The module will quit immediately if AmiTCP sends control-C.
- Aborting a list or transfer is now handled more efficiently so actions are now almost immediate.
- The file lengths accepted by the FTP module can now be a maximum of 108 characters, although anything above about 50 characters may cause some incompatibilities with Opus itself. Remember that AmigaDos files are limited to 30 characters in length so if you transfer a file with a long name the extra characters will be lost when it is saved.

Text Viewer

The Text viewer has had some minor changes plus improved performance on RTG systems. Changes include:-

- The text viewer now uses SetMaxPen() under OS3.0+ instead of setting a pixel mask. This improves the display and scrolling speed if the bitmap is non-standard, for example with Cybergraphics and other chunky screens.
- It is now possible to configure different editors for normal, ansi and hex modes.
- Files are now read in chunks of about 64k, rather than one big allocation. A progress indicator for large files, greater than 64k, allows you to abort the loading process.
- The text viewer can now be iconified. It maintains the position within the file when it is iconified as well as when it is put on a new screen.
- The file requester for **Save As** in the text viewer now has the filename filled in by default.

FileTypes

Changes and Notes

- There is a new filetype matching command **Disk**. This allows you to match a disk (similar to **Directory** for directories). It can optionally take a filesystem ID to match a specific filesystem, or it can match on device name.

For example,

Disk matches any disk

Disk DOS1 matches standard FFS disk

Disk DOS? matches any DOS disk

Disk MSD0 matches a CrossDOS disk

Disk DH3: matches device DH3:

Disk DF0: will match any floppy in drive 0

Currently, none of the events for Disk filetypes are used, but you can use this to add items to the icon popup menus, and also to set the default disk icon for a particular file system or device.

- We have added additional filetype actions to the filetype editor. There are ***six new User actions*** (along with User1-4 making a total of 10), and qualified double-click and drag and drop actions. The filetype editor window has grown to accommodate these. The new *User actions* are accessed via a generic **User** command, which takes a number as a parameter. Note the space between the command User and the number. This was done to avoid having to add 6 new commands. For example,

Command User 5

instead of

Command User5

FileTypes

Note that User1, User2, User3 and User4 still work for compatibility.

You can now also have filetype actions defined for control and alt drag&drop, and control and alt double-click (shift is already used by Opus). The **DoubleClick** and **DragNDrop** internal commands have new templates:-

DoubleClick ALT/S,CTRL=CONTROL/S,NAME/F

DragNDrop ALT/S,CTRL=CONTROL/S,SHIFT/S,NAME/F

- In the Filetype editor, deleting the User1 function will no longer delete the popup menu entries for the filetype.
- **Store** in Filetypes now checks if the filetype already exists in the Storage directory. If so it will just be removed from the current filetype list.
- The **{Q!}** parameter now works when used in a filetype popup menu function. **{Q!}** also now works through the filetype command system (for example, "command source <handle> user4")
- It is now possible to add your own separator bars in filetype popups by specifying a label of three minus signs (---)

Scripts

The dopus5.library now monitors the device list itself, rather than relying on the system to generate disk change events. This means that the disk inserted/disk removed scripts now get notified when any disk is changed (e.g. CDs, removable hard drives, etc), instead of just for floppies.

Compatibility

- Opus now allows the **MUI** title bar popup menu to work correctly if the mouse is over a MUI aware program.
- Earlier problems with the akJFIF, akGIF, etc **datatypes** were traced to the fact that Opus normally turns off DOS requesters by setting pr_WindowPtr=-1. The ak* datatypes weren't checking this properly and were trying to open the progress bar on an invalid screen. This has been also fixed in later versions of the ak* datatypes. (See Aminet for latest releases. Thanks Andrew!) In Opus, the show.module now sets pr_WindowPtr=0 when it tries to load a datatypes picture. An added bonus is that you now get the progress bar from ak* datatypes.

The screenmode that the show.module picks for datatypes pictures is also now more accurate.

- **MCP** and some other utilities look for specific name in the screen title. By setting the ENV variable, **WorkbenchTitle**, you may stop Opus generating its own clock/memory display in the screen title bar. Instead, it will set the title as *Amiga Workbench* to be intercepted by MCP and similar utilities. (See Environment Variables)

Commands and Functions

- **Commands now operate on Icon Mode Listers.**

Many people requested the ability to use their buttons and menus on listers in standard Icon mode. We have enhanced the command subsystem to make this possible, and Icon mode listers can now be normal 'source' listers like those in Name or Icon Action mode. However, because Icon mode listers do not have a 'source/destination' display like the other style of listers, the lister window has to be active when the function is executed for an Icon mode lister to be recognised as a source.

- **Asynchronous Commands**

By default, many Opus commands will detach and operate asynchronously. The main commands that do this are *AnsiRead*, *HexRead*, *Play*, *Print*, *Read*, *Show* and *SmartRead*. This caused a problem for some people who wanted to define a function that called one of these commands for a file, and then deleted the file. For example,

```
Command Play ram:playfile
Command Delete ram:playfile
```

In this example, a 'race condition' would be generated between the *Play* command, which operates asynchronously, and the *Delete* command. Either the *Play* command would get to the file first, in which case the *Delete* command would not be able to delete it, or the *Delete* command would win, in which case the file would not be there for the *Play* command to open.

To solve this problem, we have added a **WAIT** parameter to the above commands. If you specify this argument, the command will not operate asynchronously, but will instead wait until it has finished before returning. Taking the above example,

Commands and Functions

Command Play ram:playfile WAIT

Command Delete ram:playfile

With the **WAIT** parameter specified, the *Play* command will not return until it has finished playing the sound, and the *Delete* command will then be able to delete it.

- **New SetAsBackground command (Only OS3.0 or greater).**

The new *SetAsBackground* command takes the filename of a picture and installs it as the background in either the desktop, windows or requesters. The main use of this command is in a filetype popup menu, whereby you could click the right mouse button on a picture, select the 'Set As Background' option and instantly have the picture as your backdrop image. Another use of this command would be to implement a random background picture switching application for Opus.

The command template for *SetAsBackground* is as follows:

**NAME,DESKTOP/S,LISTER=WINDOW/S,
REQ=REQUESTER/S,TILE/S, CENTER/S,PRECISION/K**

NAME is the name of the picture file.

DESKTOP means install this picture as the desktop backdrop picture. This is the default and does not need to be explicitly specified.

LISTER and **REQ** allow you to set the background picture for listers/groups and requesters.

TILE and **CENTER** will tile or center the picture respectively.

PRECISION lets you specify the remapping precision; valid values are "none" (no remap), "gui" (poor), "icon" (ok), "image" (good), and "exact" (best). Please note that the Amiga datatypes system is responsible for remapping the images, and so the results achieved may be dependent upon your datatypes settings.

- **Changes to the Set command**

You can now use the internal *Set* command to modify icon label colours on the fly. You could use this ability in conjunction with the *SetAsBackground* command to implement a random background switcher script for Opus.

Usage :

Set labelcolour [desktop | windows] <fg> <bg> <drawmode>

For example,

Set labelcolour desktop 3 1 jam1

Set labelcolour desktop 7 0 jam2

- **Changes to Rename command**

The internal *Rename* command now has a command template of **NAME,NEWNAME**. Both **NAME** and **NEWNAME** can accept asterisk * wildcards, like the Rename requester itself.

- **Changes to the Copy and CopyAs commands**

We have added **MOVEWHENSAME** switch to *Copy* and *CopyAs* commands. When specified, this switch *changes the Copy function into the Move function* when the source and destination paths are both on the same disk. This emulates the behaviour when dragging and dropping icons; within the same disk, the files are moved, when dropped on a different disk the files are copied.

Function Editor and Commands

The **No file quote** flag in the Function Editor had the limitation that it was global to the whole function. Many people requested the option of more control over filename quoting. To overcome this limitation we have added several *new argument variables* for use with Opus functions. These variables are:-

Commands and Functions

{o'} Same as **{o}** (first selected file name) except that the file name will always be enclosed within quotes.

{O'} Same as **{O}** (all selected file names) except that the filenames will always be enclosed by quotes.

{f'} Same as **{f}** (first selected pathname) except that the pathname will always be enclosed in quotes.

{F'} Same as **{F}** (all selected pathnames) except that the pathnames will always be enclosed within quotes.

In addition, there are also the variables:

{o~}
{O~}
{f~}
{F~}

These are the same as above except the names will never be enclosed in quotes.

The use of ' or ~ in an argument variable overrides the setting of the **No file quote** flag; without either of these modifiers, the flag setting is used.

Note that if you use the - modifier to strip the file suffix, it must come after the ' or ~ character. For example,

AmigaDOS echo Stripped filename is {o-}

If you wanted the name to be quoted, this would become:

AmigaDOS echo Stripped and quoted filename is {o'-}

Changes to ARexx commands

- ***lister reload*** to reload (or load for the first time) a file in a lister. The template for the command is:-

`lister reload <handle> <name>`

This also has an *update* flag to make it update the lister timestamp (to save unnecessary reloading)

`lister reload <handle> <name> update`

For example,

`lister reload 12381928 'filename.lha'`

If the file previously existed in the lister, the user data, user menus and version information of the file is preserved.

- ***lister <handle> query window*** gets the window pointer of a lister. This is the address (in base 10) of the window structure, or 0 if the lister has no window.
- ***lister findcache*** allows you to find a cached directory and display it in the lister, for example,

`lister findcache <handle> <path>`

When it returns, **RESULT** is set to 0 if the path was not found, or 1 if it was found. If the path is found, it will be automatically displayed in the lister and you don't need to do any more. If it is not found, you'll have to read the directory as normal.

- ***lister clearcaches*** will flush any caches that were created by your lister, using your custom handler. No other caches will be affected.

- For file Listers, the popup menu **CopyTo** has been expanded so you can now *copy* to ARexx scripts. To do this,
 - a) Create an ARexx script in the *DOPus5:system/CopyTo* directory.
 - b) Set the comment of the script file to be the name that you want displayed in the menu
 - c) Set the Script protection bit of the script file

The ARexx script will be passed the name of the file that was selected and the Opus ARexx port name. For example,

```
> list DOpus5:system/copyto
```

```
TestScript.dopus 5135 -s--rw-d Today 00:07:12 : Test ARexx Script
```

```
> type DOpus5:system/copyto/testscript.dopus5
```

```
/* A test of a CopyTo script */  
parse arg filename portname source options results  
str="File : " | filename | "" dopus request str "Ok"
```

See Dopus Tricks and Tips from our web site for more details.

- You can now add a trap for all internal commands using an asterisk *, for example

```
dopus addtrap * myhandler
```

- The **dopus command** now supports wildcards in the *type* field, allowing you to add menus to multiple filetypes at once. For example,

```
dopus command "Unpack" program "Unpacker" 'source'  
ext 'Unpack...' type a*
```

This would add the *Unpack...* menu item to all filetypes with IDs starting with "a". Full wildcards are supported.

- There is now a *nopopups* keyword for the *lister set handler* command to disable all file popups in that lister. If you have added your own popup menu to an entry it will be shown even if the *nopopups* flag is set.
- Free Space Gauges are turned off by default for custom handler listers; if you specify the *gauge* keyword for the *lister set <handle> handler rexx* command you can enable them but remember that the user may still have turned them off in the environment.
- Drag and dropping onto sub-directories is disabled by default for custom handlers. You need to add the '*subdrop*' keyword to the *lister set handler* command to enable dropping into sub-directories for your handler. Note that the user may still have sub-dropping turned off in the environment.

The "*drop*" message now contains the full destination path in Arg5. You can compare this against the path of the destination lister (handle in Arg1) to see whether the drop was into a sub-directory or not.

Other ARexx Notes

- In some circumstances DOpus would set a result string for an ARexx message even if it didn't request one. This is fixed.
- Routines have been added to the *dopus5.library* that let you get and set stem variables for ARexx messages from C (see the Opus SDK).
- Fixed a problem with the *lister query entry* command with filenames longer than 40 characters in length.

Miscellaneous Enhancements

- The **LoadDB** command now supports the **NEWPATH** switch of the LoadWB command. If Opus is already running and you execute **LoadDB NEWPATH** (or LoadWB NEWPATH if running as WBR), Opus will not try to run again but instead will update its copy of the path list with the path in the current shell. LoadDB is smart enough to pass the NEWPATH command through to the original LoadWB if Workbench is also running (or running instead of Opus).
- We now correctly support DPaintV/PPaint AnimBrushes for use as buttons and images.
- Groups can now contain items that don't have their own icons so you can now add left-out commands to groups.
- Filename length limit increased from 30 to 31 characters to support AFS partitions.
- The **Disk Information** item in disk popup menus is now disabled for bad disks.
- The version routine is improved and will find version information stored in data hunk rather than just a code hunk (Andreas Kleinert and AK datatypes).
- Page up/down and home/end keys now operate in all listviews.
- You can now press 'y' or 'n' in most requesters for ok/cancel. Except of course where it would clash with the buttons text.
- ESCape will close the About requester.
- Clipping filenames from a lister to the clipboard now uses the volume name rather than the device name. (For example, Workbench:C/Info rather than DH0:C/Info)

Bugs Squashed and Other Notes

The following is a list of some problems in the earlier version which have now been identified and fixed.

- Opus did not recognise that the DONOTWAIT tooltipe was not set on objects in the WBStartup drawer, and always returned immediately. Opus now behaves properly if DONOTWAIT is not present, by waiting for about 5 seconds for the program to return, and then displaying a requester.
- Bad disk icons now disappear when the disk is ejected
- The text viewer was losing 96 bytes. Also if you tried to *Read* a file that had the 'read' protection bit cleared, the text viewer would crash.
- The IFFOpen() routine in the dopus5.library now fails in safe mode if the existing file is read-only. (SDK issue)
- The command *Devicelist NEW FULL* would incorrectly fall back to a standard devicelist if there were any SRC listers.
- If the 'Prefs' field in *Environment/Display Options* was empty, and you clicked the '!' gadget to edit the pattern, it would pass a bad filename which would result in an error requester.
- If you selected a 24 bit screenmode in *Environment/Display Mode* and dragged the colours slider to the far right, Opus would crash. Fixed.
- In some languages (eg Swedish) name substitution in the date field in listers resulted in incorrect padding between the date and time.
- If you copied a whole function from the function editor, and then tried to paste it into a string gadget as a text string, it would cause the whole system to lock up.

- If you tried to drag an entry out of the cachelist by pressing the right button it would cause enforcer hits.
- Entering the date in the Datestamp requester, lister editing mode, or in a range for the complex Select requester now respects the date mode set in Options/Locale. Also removed the extra space in the date column.
- Fixed the problem where the first iconified lister would go to the extreme top left corner of the screen.
- Fixed problem where the icons would get confused if you had two disks with the same names.
- Fixed some problems with the AppIcon emulation. This also fixes problems that the (external) IconClock module could cause when changing screen modes.
- If you started a second copy of Opus running, and then cancelled the warning requester, it could cause a crash/enforcer hits.
- If an iconified lister in icon mode, show all, was saved in the environment, then when Opus started up and the icon was double-clicked on, the icons within the lister would not appear.
- Fixed the wrap-around problem on the progress bar when copying files > 20mb.
- Deleting a link to a directory now just deletes the link, instead of the contents of the directory.
- Fixed problems where sometimes when deleting a directory tree, Opus would report an 'Object in use' message and prompt with a requester before continuing.

- Fixed problem where NFS-mounted volumes would not show up as icons even though they would show up in the device list.
- If you were in lister edit mode, had made a change and then closed the lister while edit was still active, Opus would make the change in its buffers but not on disk. The change is now discarded completely.
- Added check for possible divide-by-zero in progress bar calculation to fix the reported crashes when copying large files.
- Fixed bug in CleanUp module that was losing AppIcon positions.
- If you set the Output Window title to a string greater than about 28 characters you would get enforcer hits/crashes when a program opens an output window.
- Fixed a problem with AppIcon positioning where internal settings were not respected.

SDK and Associated Programming

The Opus programming interface has been greatly expanded in the Magellan version. Over 40 callback hooks are provided which let your programs access Opus directly, making custom modules easier to write and more powerful than ever before. For information about the new features, refer to the Opus SDK.