DirectoryOpus 5.82

Copyright Notice

Disclaimer

Program Licence

Acknowledgments

Directory Opus 5, Opus Magellan and Opus Magellan II programmed by Jonathan Potter for GPSoftware, Brisbane, 1995-1998. Opus was written by Jonathan Potter. OpusFTP and AFC modules were programmed by Andrew Dunbar and Greg Perry. Graphic design by Leigh Malpas. Manual written by Greg Perry and Jon Potter.

Jon and I express our appreciation to those who took part in the extensive beta testing of this Opus Magellan-II version. Special thanks to Martin Bernd, Neil Bothwick, Dave Clarke, Leo Davidson, Eric Giroux, Helmet Hummel, John Lowery, Trevor Morris, Dieter Muller, Bill Near, Neil Sedger, Martin Steigerwald, Frederic Steinfels, Thomas Svenson, Ben Vost and others.

Updates and Changes

Please read the ***ReadMe_Magellan*** file included in the distribution archive for any changes to the software or additions to the manual since the time of publication.

Table of Contents

Supplement for Directory Opus 5 Magellan-II 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 and Magellan-II versions.

You will need BOTH the original Opus 5.5 manual and this supplement for complete documentation of Directory Opus 5 Magellan-II.

Installation

Installation of this version of Directory Opus 5 is straight forward. It is not intended to be installed over a previous version of the program. Open the disk window, double-click the Install_Opus_Magellan-II icon and follow the directions. When the installation is complete, reboot your Amiga.

All program components of this new Magellan-II version have been updated, so a complete installation is required.

Introduction

Welcome to the Directory Opus Magellan-II upgrade!

From its first release in early '95, many Amiga users could see the raw power provided by Directory Opus 5 with its unique concept integrating Workbench Replacement Mode (tm), file management and custom functions in the one package. These concepts have been further refined with each new version. 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 for future directions.

Through your support, Directory Opus has become established as the premier Workbench replacement and file management program for the Amiga. This new version attempts to return your loyalty with ongoing support for the Amiga platform and a continuation of our development plan for Opus.

Thanks!

Opus Magellan-II brings you another evolutionary step. With over 500 changes in the internal code, we are proud to give you a significantly improved Workbench Replacement (tm) system which provides the Opus user with a modern front-end and Graphical User Interface to our aging Operating System. We have also tried to add as many of your user suggestions as possible which are consistent with the metaphor of Opus.

Remember. Pass the word, not the buck!

Summary of Enhancements

The Magellan-II version not only upgrades many of the earlier functions but also involves an extensive re-development of many components of the Opus 5 system to give you enhanced usability, especially in the Workbench Replacement Mode. Amongst other features, in this version we have concentrated on providing greater user control over your custom Environment with a more powerful but easier to use configuration section, plus many new features such as custom Themes, custom sounds and scripts, improved User and Start Menus, greater Lister functionality, and a dramatically enhanced OpusFTP system where you can now tailor custom setting for individual remote sites and even leave out the site or remote files on your Opus desktop for instant access.

Amongst many other features, the Magellan version gave you:-

- ***Enhanced Lister functionality*** including:- command functions which 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***.

- A new ***Desktop Folder*** mechanism: Drag and drop items to the desktop with the support of optional popup menus.

- A new ***Icon Positioning*** system. Configure areas where icons will appear and their priority.

- Improved ***Icon dragging*** routines plus global control of ***icon labels and borders***; ***icon label splitting*** for long labels; a new ***Icon Command functio***n for special "command" files; and the ability to use traditional Workbench icon positions for icons or exclusive Opus positioning.

- New ***Popup menus*** with shadow look.

- ***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 plus support for image and animation file formats including DPaintV/PPaint AnimBrushes.

The **new Magellan-II version** refines and extends these features further. These changes are discussed in detail on the subsequent pages, but in summary, the new features include:-

- ***Custom Theme***s - customise your system with your favourite images, sounds, fonts and colours.

- ***Improved Lister layouts*** including proportional fonts, resizable fields, sort indicator, better in-line editing.

- An ***amalgamated Options and Environment*** editor to give you a more intuitive layout for better management of the display and program operations.

- ***Background pictures*** in button banks and start menus including ***Random images***.

- New ***Outline and Shadow font*** options for Desktop display.

- Fully user ***configurable screen title*** for display of used and available memory, OS versions, processes etc.

- New ***Sound events*** for Opus and system events - attach a sound to inserting and removing a disk!

- New Opus ***Path List*** settings allowing you to easily set the system paths for commands etc.

- Support for Long Filenames up to 107 characters.

- ***New graphical layout*** and better configurability of Button banks and start menus including an '***auto-close***' feature. Buttons now support '***Active Popups'*** - an extended popup menu feature where each button act as a start menu.

- ***Enhanced Scripts*** system with more events and linking to internal commands and OpusFTP. ***Internal commands*** now ***replaceable*** by user-defined scripts and functions.

- ***Improved Start Menus***. You can now have multilevel menus and start menus are now always *sticky*.

- ***Improved filetype*** identification system plus new filetype matching functions for foreign disk types.

- New and extended icon and Lister and Group ***popup menus*** plus ***Improved Snapshot*** ability.

- New ***Icon Information*** requester with more detailed display plus new functionality such as ability to change icon type and edit the icon images.

- Improved support for ***Drag and Drop*** functions.

- Better ***Icon support*** with faster displays.

- Many ***new ARexx*** commands.

This functionality is complemented by a ***dramatically enhanced OpusFTP module*** with many new features including:-

- Completely ***revised GUI*** with new ***FTPAddressbook***, ***FTPOptions*** and ***FTPConnect*** requesters.

- ***Individual configuration options*** for each site including custom format and custom tool bars on a site-by-site basis.

- ***Enhanced Drag and Drop*** allows you to create shortcuts to your favourite FTP sites, directories and files by leaving them out on the Opus Desktop for instant access.

- Full ***recursive directory copying*** can copy whole directories in a single operation!

- Support for more Opus commands including ***Getsizes***, ***FindFiles*** and the ***Copy*** command now supports the UPDATE and NEWER flags. Great for web site maintenance!

- ***Protect*** is fully supported with a new GUI allowing you to set remote protection bits in either Amiga or Unix fashion.

- Seven ***new Scripts*** exclusively for FTP on a site by site basis.

- Automatic ***reconnection*** on failure or lost connection.

- Better ***site-to-site transfe***r capability - transfer files between two remote sites without going via your Amiga.

- Optional ***transfer and progress displays*** with more detailed information.

- Support for ***firewalls*** and IP Masking with ***PASSIVE*** transfer mode.

- Lister path gadgets accept ***industry standard URL syntax***.

- ***Inline editing*** of Name and Protection fields now supported.

- ***Synchronous command structure*** so ***interactive ARexx scripts*** are now possible! Write ***BATCH files*** to transfer selected files.

- ***Speed and efficiency*** improvements with better management of socket options and improved support for all AmigaTCP stacks.

General Changes

Long File Names

Because Directory Opus was designed to complement and enhance the Amiga OS, we have traditionally supported the normal Amiga filesystems (OFS and FFS). These filesystem have an in-built limit of 30 characters for length of the filename as defined from AmigaDOS programming guidelines. However, there are now some new, third party filesystems for both disk and CD which have added support for longer names. To support these we have rewritten the Opus file management routines to support filenames of up to 107 characters for all internal operations.

Since it is not possible for Opus to detect what the maximum file length supported by any given filesystem may be, you must define the maximum length in the Environment / Directories section. (See Environment.) By default the limit will be set at 30 characters but you may set this to any length up to the 107 limit.

We recommended that you do not set this limit higher than supported by your filesystem. If set too high you may encounter problems when copying files and it also will result in a considerable waste of memory and may slow operations of the program in general.

Setting the limit lower than used by the filesystem will result in Opus truncating the names to this length. (Note: you cannot set the length to less than 30 characters.)

Because of internal AmigaDOS requirements, there is a set limit of 256 characters for the directory plus filename combination, unless relative directories are used.

WARNING: We provide no guarantees for long filename support. Both we and the developers of such filesystems have tested Opus extensively with such long names and while there seems to be few problems, some artefacts have been observed depending on the specific filesystem and action being performed. Remember, since the Amiga was developed with a 30 character limit as the standard, some Amiga functions and some third party software can potentially crash your system when used with long filenames.

For example, the icon.library calls for GetDiskObject and PutDiskObject calls will fail if the filename (plus .info) is longer than 30 characters. There are others!

Opus will internally process files with long names but since it cannot detect the absolute limit imposed by a given filesystem, Opus does no filename translation when performing file operations. It is the responsibility of the user and the filesystem to truncate the file length as required. In some cases you may get errors when attempting to access files. In other cases involving a recursive copy of a complete directory structure, you may end up with files in the wrong place because the parent directory could not be created or the copy operation has failed because the name cannot be found. *See also Environment*

Menus

The program menu structure has been revised to improve readability and ease of use. The Settings menu has been revised as shown below.

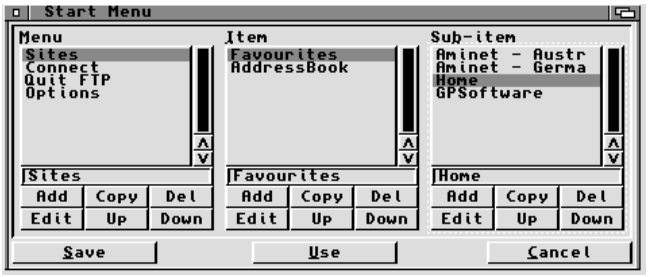
The older *Options* and *Environment* menus have been combined into one *Settings/Environment* menu and new menus have been added to support *Themes* and the loading and saving of environments, which may be done from the separate menus or from the Environment editor itself. *Save Layout* is now a function in the *Settings* menu that lets you save the current layout on demand.

Some users suggested that our use of the generic name *Snapshot*' could be confusing depending on the specific context in which it was used. So, we have changed terminology of the word

throughout the whole program including various menus. We now explicitly tell you which objects you can snapshot.

Menu Editor

In previous versions, Start Menus and Lister Menus could only have single items like traditional Amiga Menus. In Opus Magellan-II you can now have an extra level of sub-items. We have expanded the menu editor to support this. What were items and sub-items have been shifted one place to the left to allow an increased menu depth. *See also Listers, Start Menus*

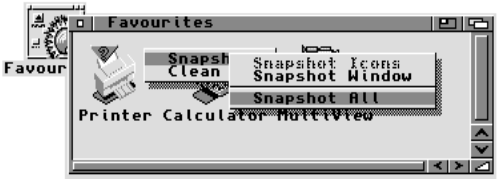


Popup Menus

All popup menus, including start menus, are now sticky even when the underlying window is not active. *See also Start Menus*

Groups

Group drawers have a new popup menu giving you the direct support for Cleanup and Snapshot of the group icons, the window or the complete group directly.



Cybergraphics Support and Graphics Cards

Cybergraphics RTG systems have improved support where the bitmap is non-standard. This provides greater display speed on Cybergraphics and other chunky screens and the new set of custom icon dragging routines gives much faster Icon dragging.

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.

Screen to Front

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

Themes - To go boldly....

In many modern operating systems you can adjust the visual display characteristics of your system to reflect a particular style or *'theme*'. Examples of such themes might be Animals, Adventure, Space, Science, favourite TV shows, Cartoons and so on. Many users have asked for better support for configuring the overall visual look of their Amiga display, so to support this, we have added this concept of *Themes* for all display features controlled within Opus itself. In Opus, a *Theme* consist of :-

- background pictures for desktop, Lister and buttons,
- sounds for various events,
- user pens and palette colours,
- font settings.

Technically, an Opus '*theme*' is a small ARexx script file, called the '*xxxx.theme*', which references the image and sound files contained in your theme. The theme files are designed to be stored in the DOpus5:Themes directory and depending on whether you have 'built' or imported a theme or just saved the theme, there may also be an associated directory under the same name.

When you load the theme, Opus runs the ARexx script and makes the internal changes to the Opus environment settings for your theme design.

We have provided a few sample themes on the distribution disk but more themes plus other material will be available separately on a *DOpus Plus CD*, via our web site, AmiNet and through popular Amiga magazines. See the readme on the distribution disks for details of how to install and test these themes.

If you have access to the internet (or CD) you will find that there are literally thousands of themes available for Windows95/98. For copyright reasons we cannot include these directly with Opus but we do provide a special command which allows you to convert these files to a format suitable for Opus.

Themes are accessed from a new Themes sub-menu in the Settings menu or via four new Opus commands for ***LoadTheme***, ***SaveTheme***, ***BuildTheme*** and ***ConvertTheme***.

☞ ***Remember that a 'theme' is simply a set of Environment settings. Once you have set the theme you want to use, select Save Environment from the settings menu to have these settings preserved for the next time you run Opus.***

Creating Themes

To create a new theme for your own use, configure your Opus Environment and set the desired images, sounds, colours and fonts, then 'save' the theme under a defined name. You may then load this theme to reconfigure your Opus whenever you wish.

To create a theme to give to others, you *build* the theme. By default, when you *save* the theme Opus creates a special theme file which *references* the images and sounds in their original place on your hard disk. This saves disk space on your system and does not require multiple copies of the same image or sound file. However, if you wish to give a theme to others, all the relevant files must be collected into one directory. This is known as *building* the theme. Once built you can archive the theme file plus its associated directory and give it to others.

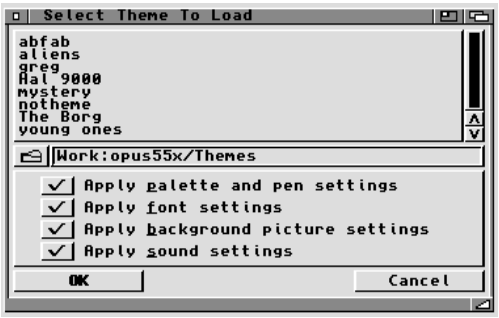
We encourage you to build your favourite new themes and place them on AmiNet or our web site for others to access.

Theme Commands

Themes are supported by four new Opus commands:-

LoadTheme

The **LoadTheme** menu or command is used to load and display a new theme. From the displayed requester, you can choose which



elements of the theme to apply to your system.
Opus needs to store your favourite theme files in a defined place. By default this is set to *DOpus5:Themes* and is assigned to the name *D5THEMES:*, but this may be changed in the *Environment / Backgrounds* section.

Opus also supports loading themes from places other than *D5THEMES:* from the **LoadTheme** requester. Therefore you can load a new theme from a CD or other directory on your hard drive if you do not have space on the partition containing DOpus. However, we recommend that you keep all themes under the *DOpus5:Themes* directory to avoid confusion.

SaveTheme

The **SaveTheme** menu or command allows you to store a theme while you are working on it, or to store multiple themes for your own use.

SaveTheme does not copy the actual sound and picture files to your themes directory. Instead, it generates the special Opus '.theme' file in the themes directory. This file references the real locations of the component files on your system and sets the Opus Environment to use these internal settings.

BuildTheme

If you wish to collect all your theme related files into the one directory, install themes from an outside source such as a CD, or prepare a theme for distribution, you should use the **BuildTheme** menu or command. This creates a sub-directory in *DOpus5:Themes* and copies all sound and picture files used in the theme to that directory automatically. It then creates a new theme file which references the component files via *D5THEMES:* not their absolute pathnames. Once you have used the **BuildTheme** command, you can archive both the theme file and the associated directory and distribute the theme archive to other users.

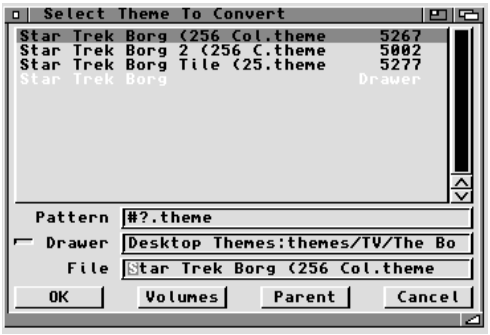
ConvertTheme

or "How to convert a Windows 95 Theme for use with Opus".

Unfortunately, the theme files from Windows 95 are not directly compatible with the Amiga and Opus. But, we provide a fairly simple process which allows you to take a Windows 95 theme file and convert it for use with Opus.

ConvertTheme is designed to read an original Windows 95 theme file and produce a DOpus theme file containing the correct internal references to the images and sounds. You can then use the **LoadTheme** command to attempt to view the theme. If you like the theme, you can use the **BuildTheme** command to copy it and related files to your Amiga.

Perhaps the easiest way to run this command is from the Execute Command requester. Select ***Execute Command*** (RAmiga-E) from the main Opus menu, and enter *"+ConvertTheme"* into the command field. (Note the plus sign, this tells Opus to execute an internal Opus command, not an external AmigaDOS program.) If you are going to be converting a lot of themes, you might want to add the **ConvertTheme** command to a button or menu.



From the file requester, navigate to the Windows 95 '.theme' file that describes the theme you wish to convert, select it and click OK.

From the next file requester, enter the output name for the converted theme and click OK (the .theme suffix will be added automatically for you). The requester will default to showing the *D5THEMES:* directory and we suggest you save the converted theme there.

The ConvertTheme process will then convert the theme file - it's that simple. You can then use the **LoadTheme** command or select 'Load Theme' from the *Settings* menu to load the new theme and apply its settings to Opus.

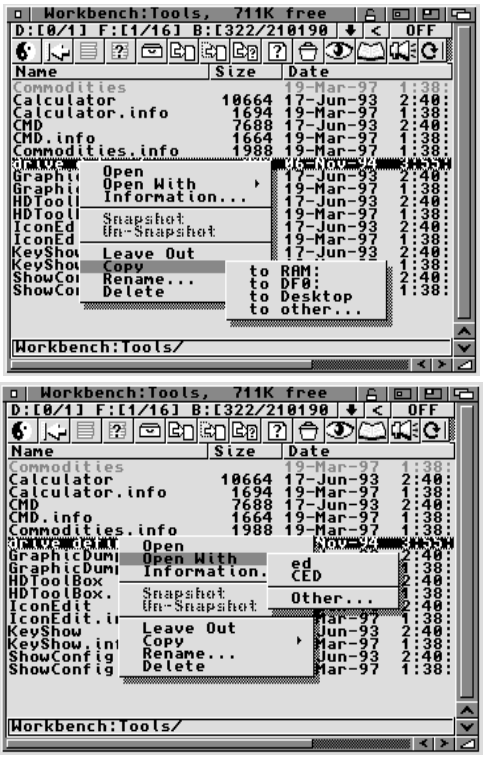
☞ ***ConvertTheme simply builds a new Opus theme file containing references to the original files. If you convert a theme from a CD, the new theme file would still point to that CD for its graphics and sound files. If you want to copy the complete theme to your system so you can use it permanently, you can use the BuildTheme command (once you have converted the theme).***

☞ ***ConvertTheme does not convert the formats of the image and sound files themselves, it only converts the '.theme' description file. Windows theme files usually contain images in JPG, GIF or BMP format with sound files in WAV format and these can often be used on the Amiga with the appropriate datatypes installed. However, you may find that your system performs better if you convert these to Amiga specific formats before use. For example, many WAV files do not play correctly from the WAV datatype and are best when converted to 8SVX. (There are a number of third-party programs which will do these conversions and many are available from Aminet and the DOpus Plus CD.)***

File Listers

Lister Popup Menus

Popup menus 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

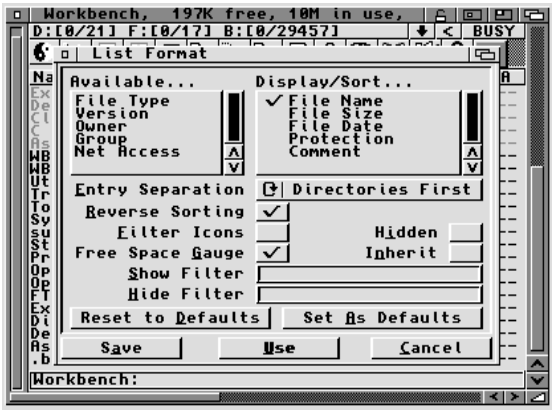
- Project files have a new **Open With..** in the file popup menu, which lets you select an application to open the file with. By default, Opus remembers up to the last 10 programs chosen but this may be adjusted from the Environment / Miscellaneous settings.

- The **delay** before opening the popup sub-menus may be adjusted from the *PopUpDelay* value in the *Environment / Miscellaneous* settings.

- **Sub-Menus -** Lister menus can now have sub-items. You can effectively have three levels of menu commands. In the menu editor, what were items and sub-items have been shifted one place to the left to allow an increased menu depth.

New Lister Format Editor Features

The **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*.

This lets you set which Listers you want the gauge to appear. For example, if you only want a gauge on the root directory of certain devices, turn it off in the default Lister format and then turn it on for those directories.

The **Inherit** option 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

The **Filter Icons** flag from the Lister Format is now ignored in the Icon modes so all icons are now displayed correctly.

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

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

Holding down the Ctrl key while double-clicking on a directory in an Icon Action mode Lister will cause the new directory to be read into the existing Lister instead of opening a new one.

Name Mode Listers

Proportional Fonts for Listers

Proportional fonts are now fully supported with name mode Listers. The selected font may be set in the *Environment/Lister Display*

Adjustable Field Sizes

The size of the Lister fields is set dynamically by Opus according to the width of the field's contents. However, you may now adjust the size of any field in real time by selecting and *dragging the separators* between Lister field titles. This feature allows you to resize a wide field such as 'Name' when using long filenames so you may see the other fields more easily.

To restore a field that you have resized to dynamic sizing, *double-click on the separator*.

Field sizes are reset dynamically when you change directories. If you change field sizes in a given directory, the sizes in that specific directory will remain set until the directory goes out of the cache or until memory is flushed by Avail Flush or a low memory condition.

Remember, you can also re-order the Lister field titles and format of the Lister fields by dragging the field title themselves to the required position. For example, you could pick up and drag the 'Date' field to the left hand edge of the field titles and drop it on the Name field to rearrange the display.

Visible Sort Order

You may change the sort method for entries in a Lister by clicking on a field title. Opus now complements this ability with a visual indicator to instantly show you the order in which your files are sorted. A small triangle is displayed in field title bar to indicate the current sort field and sort order.

Inline Editing

You can edit an entry in the Lister by using the rename command (from a command, toolbar button or RMB popup menu). Or, you can activate inline editing mode in a name mode Lister by clicking on an item and holding the mouse button down for a few seconds. (The activation method of inline editing, if any, using left or middle mouse buttons can be set in the *Environment / Lister Options* settings.)

Edit the details as required and press **TAB** or **SHIFT-TAB** to move between fields. **SHIFT-DEL** and **SHIFT-BACKSPACE** are supported and delete from the cursor to the end of field or start of the field respectively.

Double-Click

Double-clicking on a program in Name mode 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.

FTP URL in Path Field

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

Drag and Drop into a Sub-Directory

When enabled from *Environment/Lister 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.

Note that this does not apply when dragging icons, only for files/directories from a name mode Lister.

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

Drag and Drop to Parent Directory

The Magellan II version complements drag and drop into a sub-directory by adding the ability to drag and drop to the parent directory of a Lister (in name mode only). This is done by dropping files on the *left edge of the window* (the hidden parent gadget). Note: This can be slightly difficult to perform with a normal Lister and is much easier if the fuel gauge is enabled.

Special Keys

The '0' key was actually predefined by Listers, and so could not be used to scroll to a file beginning with a 0. This is now changed (the '=' key now does what the '0' used to do)

Lister Snapshot

We have changed the '*Snapshot*' terminology to avoid potential confusion. The Lister popup menu now specifies whether you wish to snapshot the Lister (its display mode or status and the position) or the icons.

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.

Opus Magellan-II 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.***

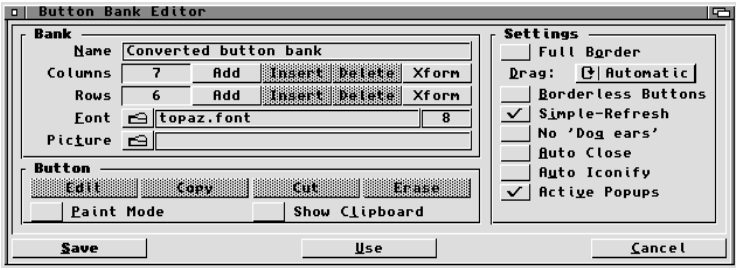
Sample NewIcon Images for Opus

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 on the DOpus Plus CD and from our web pages at http://www.gpsoft.com.au or from AmiNet.***

Button Banks

Button banks have been improved with a number of new features and additions. These include:-



Background Pictures including Random Images

Button banks can now have background pictures (only under OS39+.) From the Picture gadget, select the required image from any format supported by the datatypes you have installed. For **random pictures**, enter either the directory name only or use a wildcard pattern ('*' or '#?' patterns) for specific pictures. Some examples might be:-

Work:Picture
or Work:Pictures/Shrubbery#?
or Work:Pictures/#?.gif

See also Environment, Start Menus

New Settings Options

Auto Close and Auto Iconify

Buttons now have options for **Auto Close** and **Auto Iconify** options. When selected, these cause the button bank to be automatically closed or iconified whenever a button in the bank is chosen. This provides a mechanism which saves you having to add a CloseButtons commands to every button in the bank.

Drag

New options have been added to the dragbar orientation settings (now just called 'Drag'). You can choose to let Opus pick the most appropriate orientation by setting **Automatic** or you may choose to set the drag bar to be horizontal on the top or bottom, vertical on the left or right, or to turn the drag bar off completely.

Active popups

If this is enabled, buttons with multiple functions behave more like start menus when you hold the mouse button down. Instead of allowing you to select the "default" function for the button, it activates the function you have selected immediately.

Button bank Drag Bar

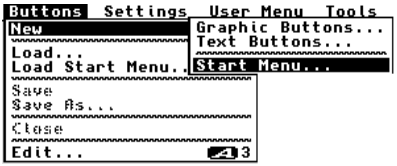
Many users like to have button banks with a hidden drag bar. Even when hidden you can still access the drag bar quickly.

Control-left click on a button bank toggles the drag bar on or off.

Control-right click on a button bank brings up the bank popup menu (previously you could only access this by right-click on the drag bar).

Start Menus

A 'Start Menu' is a single button with a number of attached popup 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.



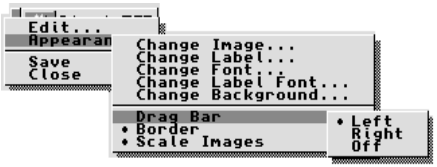
You may create a new Start Menu or load an existing Start Menu directly from the main Buttons menu.

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 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, called 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. **Edit** allows you to edit the commands in the menu; **Appearance** lets you edit the display of the Start Menu itself to change the name, image, fonts, borders and drag bar orientation. For example, you could put the drag bar on the right-hand side of the Start Menu, and then position the Start Menu hard up against the right side of the screen or you could turn off the drag bar altogether.



As with all popup menus, Start Menus are now *sticky* even when the underlying window is not active.

Sub Menus

As with other Opus menus, Start Menus can have sub-items giving you three levels of menus. In the menu editor, what were items and sub-items have been shifted one place to the left to allow an increased menu depth.

Background Images

Similar to button banks, start menus can also have background pictures (only under OS39+). The **Change Background** menu allows you to select the desired image. Random background images can be chosen as with button banks by selecting a directory only and/or using a wildcard pattern.

Fonts and Font Colours

The font requesters for the **Change Font** and **Change Label Font** functions in Start Menus now let you select the pen colour used to render the text.

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 are saved with the Environment when you select **Save Layout** from the settings menu and open automatically when the Environment is loaded.

Start Menu Drag Bar and Edit Menu

You can toggle the drag bar on and off by holding the Ctrl key and clicking on the Start Menu with the left mouse button. If the drag bar is turned off you can access the edit menu directly by pressing the Ctrl key and clicking the right mouse button over the Start Menu. This is consistent with the behaviour of button banks.

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.

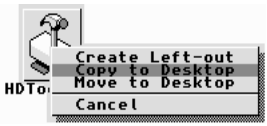
The Desktop Folder is a special directory, usually defined as *DOpus5:Desktop*. Until now, the main window of Opus (the Desktop) has been the area where drive icons, groups and left-out icons are displayed. With the addition of the Desktop Folder concept, any files that are located in the Desktop Folder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



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 Comments

- It is now possible to select borderless icons by clicking anywhere within the icon rectangle (including transparent areas). Similarly, it is now also possible to bring up the popup menu for icons by clicking anywhere within the icon rectangle (including transparent areas for borderless icons) with the right mouse button.

- When an icon is double-clicked, all other icons in that window which were selected are now automatically de-selected, unless the shift key is held down.

- You may 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.

- 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.

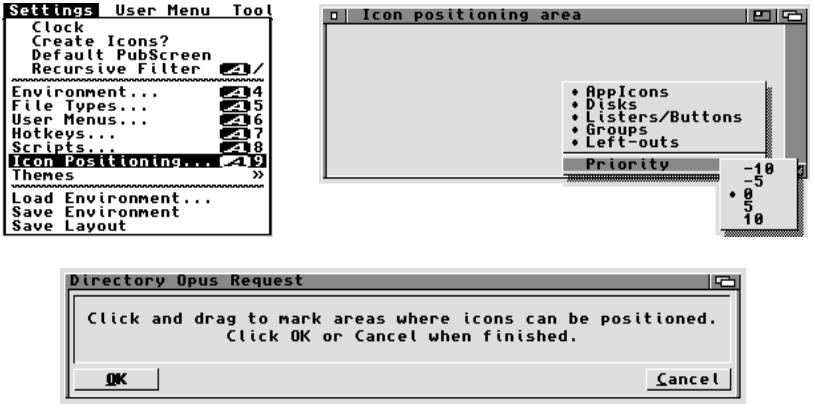
- Icon borders are now one pixel smaller vertically, to match those used by Workbench.

- Holding shift and double-clicking a disk icon now forces a Lister to open for that disk, even if one was already open.

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.

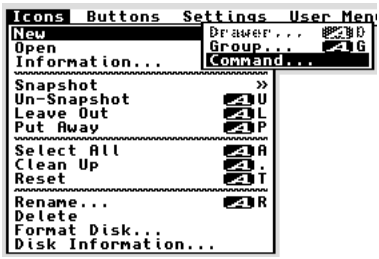


Splitting Long Icon Labels

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*.)

Icon Command Function

The **Icons/New/Command** menu allows the creation of a command file, essentially a single Opus function in a file. The command is saved to the *DOpus5:Commands* directory 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.

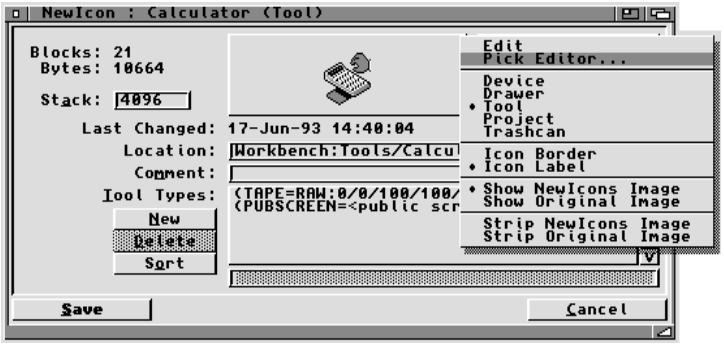
Icon Information Requester

The Icon Information (**IconInfo** command) has been expanded to provide more functionality plus better support for drag and drop of both conventional and NewIcon images and icons.

The re-sizeable requester shows 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 various attributes of the icon or toggle the display between the two NewIcon and conventional images.

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*).

The module also provides features found in some third-party icon information replacement programs. These features are available from the new popup menu or by drag and drop of icons onto the information requester.



From the menus or buttons you may access the following:-

- **Icon Type:** Change the type of a icon. The window will be updated dynamically to accommodate the extra info.

- **Default Tool:** You may select the default tool for the icon.

- **Icon Author:** This is shown as a regular tooltype so it can be edited or deleted.

- **Sort Tooltypes:** If you hold the shift key when clicking this button all disabled tooltypes will be moved to the bottom of the listview.

- **Edit:** Edit the icon using IconEdit or any other icon editor.

  ☞ ***IconEdit takes neither command line nor Workbench style arguments. To compensate for this, we use a method for sending IconEdit an AppMessage to simulate an icon being dropped onto it. IconEdit has a further limitation. When the editor starts, the mouse pointer must be within the large editing window or it will not load your icon. To compensate for this, Opus opens IconEdit under the mouse hopefully to ensure the message is recognised. It is possible that IconEdit will sometimes start but not with your icon. The only guaranteed remedy for this situation is to use a third-party icon editor instead. 'Iconian' is highly recommended.***

- **Pick Editor:** To support external third-party icon editors you can specify a command which will be used instead of IconEdit from the menu. The '*Send AppMessage*' option ***must*** be ticked for *IconEdit* to work. For other editors this is not necessary but may still work.

Drag and Drop: You may drag and drop another icon onto the different parts of the requester - onto the tooltypes area will copy the tooltypes ONLY, onto any other area will display a popup menu allowing you to copy the original image, NewIcon image or both as desired.

☞ ***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.***

Environment

With the increase in functionality of the Opus configuration system, the distinction between Environment settings which just affect the visual display and Options which just affect the operation of the program functions has become blurred. To allow you to find settings more easily, the Options editor has been removed and the settings amalgamated into a new expanded Environment section. The older Environment sub-menu has been removed from the *Settings* menu. Now there is just one Environment menu to access the editor.

Loading and saving of Environments is done from the new *Load/Save Environment* menus directly. Alternatively you may load it directly from Environment editor's menu just like all the other preference-style editors.

If you load an Environment from the *Load Environment* menu, this will load the full Environment settings including all aspect of the visual display including referenced objects such as buttons, start menus, Listers, etc. However, if you load a new Environment from within the editor, only the direct Environment settings will be loaded and reset.

The **Save Layout** option is now a distinct function in the *Settings* menu. This lets you save the current layout immediately on demand.

The following details the changes to these environment sections.

See also Menus, Options

Backgrounds

Opus Magellan can display background images directly using a defined or random picture or you may use your old WBPattern Preferences. (Similarly, you may also have images in Button Banks and Start Menus.)

Enable Backgrounds lets you turn on or off all background pictures in Opus. Underneath 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 how the images is displayed.

Display gives you the ability to centre, tile or stretch the image and set the border colour for centred pictures (normal, black or white).

☞ ***The term 'Stretch' means adjust the size of the image either larger or smaller to fit the current screen display size. Stretch may not function correctly under RTG systems.***

Remapping controls how the picture is colours are remapped to be displayed on your screen, and affects the number of pens used and the quality of the final result. The actual level used 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 as in earlier versions of Opus. When enabled, Opus will use the WBPattern preferences 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.***

Random Pictures

Random background images are now supported. If a directory or wildcard pattern is specified instead of a filename, DOpus will pick a new random picture from the directory every time it is run.

Note on Comment Field of Image files

As a special feature, the file comment of background pictures can be used to specify *individual settings* for that picture. These will *override* the settings configured in the environment. This feature might be useful if you were using random background pictures and had some pictures that needed to be tiled, some that needed to be centred, etc. It also allows you to specific the layout of a designed background image more exactly when you give files to others. For example, your image may look best when tiled or maybe when centred with a black border.

To enable this feature, the file comment of the picture file must begin with the word "**dopus** " followed by a number of keywords:

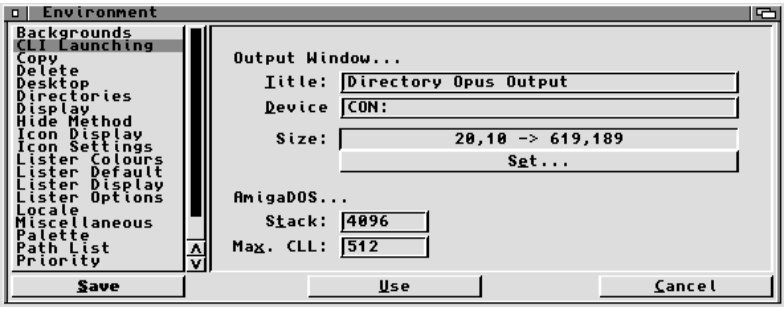
dopus [tile | center | stretch] [precision <precision>], [border off | <colour>]

tile, *center*, *stretch* specify the layout of the picture; *precision* lets you specify the remapping precision: *none*, *gui*, *icon*, *image*, *exact*; *border* lets you specify the colour for the border (when a picture is centred), or turn the border off. The colour value is specified in hex. If any or all of this information is not provided, the current configuration settings will be used instead.

☞ ***This system of file comments is specifically designed to OVERRIDE your Environment settings. ANY image file with the correct comments will override all user layout settings including when used as part of a theme. If you wish to adjust the settings yourself then you MUST REMOVE the comments first!***

CLI Launching

We have reorganised and renamed the older Output Window section to CLI Launching. These settings allows you to set the title, size and device of the Output Window used by Opus 5 to display CLI tasks and associated messages.



Device: The name of the console device used for the output window. Normally this will be set to CON: but there are other console devices which provide extra features that you may have installed. To use KingCon, for instance, enter KCON: in this field.

Stack: This allows you to set the 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 traditional default 4K stack. In such cases you may need to set this to a higher value, say 50K. (Converted UNIX programs run by the Ixemul system are an example where large stacks are required.)

CLI Launching/Max: This controls the maximum line length used for all internal Opus commands.

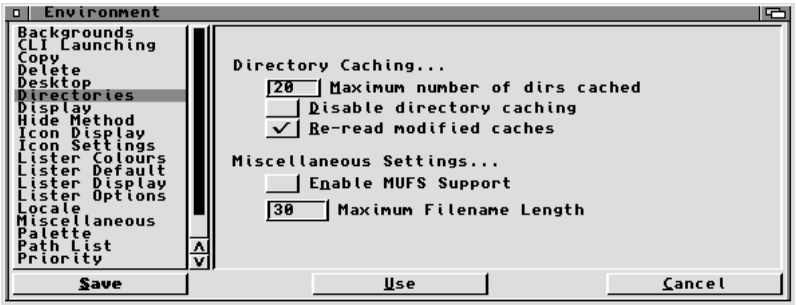
Desktop

The Desktop section controls have 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 *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 DOpus5:Desktop. See the Opus Desktop section for a description of the other settings.



Directories

The older Cache settings from Options has been expanded and split into two new sections named **Directories** and **Miscellaneous**. Some features previously controlled by environment variables have been moved to these sections for ease of access. New are:-

Enable MUFS Support: Provides limited support for the MultiUserFileSystem and will open the library on startup.

Robert Milkowski <milek@disneyland.mif.pg.gda.pl> posted the following on the DOpus5 Mailing list which sums up how to easily get MUFs and Opus running together.

"*I've got better solution that this one on Aminet. Just after LoadWB (using Opus as WB replacement) in startup-sequence type: Login own task "DO_LAUNCHER" global and everything will be ok. You have to have .166 version of multiuser.library (on Aminet). Of course in startup-sequence there is : logout global [gui] somewhere before loadwb. It's better 'coz you don't have to retype password like in this one on Aminet..*"

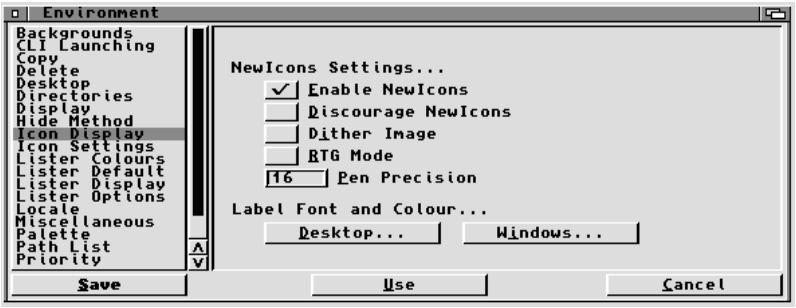
So just use the following lines:-

```
Logout <NIL: >NIL: Global GUI
C:LoadWB
Login <NIL: >NIL: Own Task "DO_LAUNCHER" Global
```

Maximum Filename Length: Opus now internally supports file names up to 107 characters in length. Set the maximum length you wish to use in this variable. The default is 30. *See General.*

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 turns NewIcons support on or off.

Discourage NewIcons tells Opus to 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.***

From **Desktop** you may edit the font settings for icons on the desktop and from **Windows** you may edit the font settings for icons in Listers/groups.

Shadow or *Outline fonts* for Desktop and window icons may be chosen from the font requesters.

Icon Settings

The main options which control icon handling have been collected and incorporated into the new section. In previous versions of Opus, some of these settings were controlled by esoteric ENV: variables. These have been removed and are now displayed as normal configuration items.

Allow icons with no labels: 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. (*See the FixIcons Command*.)

Borderless Icons are fully transparent: Tells Opus to 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.

Force split of long labels: This setting complement the **Split Long Icon Labels**. By default 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 setting forces Opus to split the label at the 150% point.

Icon Borders on by default: Turns on ***all*** icon borders by default but can be overridden on an individual icon basis from the Icon Information requester.

Quick icon dragging: If enabled, icons will no longer be masked when they are dragged. This results in an opaque background to the icon (as in Workbench) but is much quicker.

Remove Opus Icon Positions: Theoretically, setting this option along with the **Use Workbench Icon Position**s enables you to move an older Opus-ised system back to a Workbench position system without having to resnapshot all your icons. With this flag set, Opus will use the Opus icon position set in the icon structure if there is one. When the icon is snapshotted, the Opus position will be removed, and the Workbench position will be saved. Next time the icon is read the Workbench position will be used.

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

Smart icon copying: Opus 5.5 copied icons using library calls from the Amiga icon.library of GetDiskObject() / PutDiskObject() to provide compatibility with some Public Domain programs. Unfortunately this method had some unwanted artefacts. Opus Magellan now copies icons as normal using a straight byte copy. However, if you set this option, it will copy icons with the older Get/PutDiskObject() method as before.

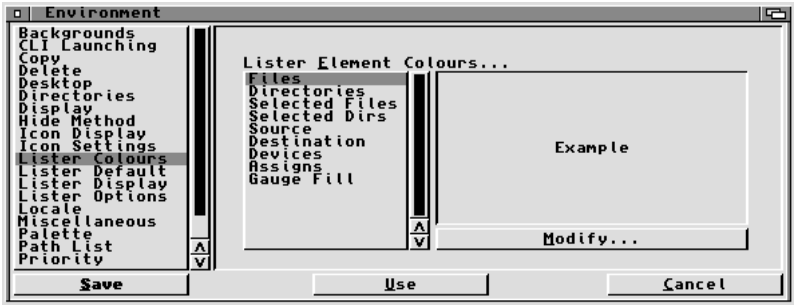
Split Long icon labels: If an icon's label is more than 1.5 times the width of the icon itself, it can be split onto multiple lines. The algorithm will only split labels on spaces, punctuation or on a capital letter. If there is nowhere to split the text then the label won't be split at all unless Force split of long labels is turned on.

Use custom drag routines: This switch should be ON for non-RTG (AGA,ECS) since it will make dragging faster. You may want to turn this switch OFF if you are using an RTG system that has its own dragging routines (For example, CyberGraphics, Picasso).

Use Workbench icon positions: Tells Opus to use the same fields in the icon as used by Workbench to store and retrieve icon and window positioning information. This provides better compatibility when you are swapping disks with other users. If you use Opus's internal icon positioning system then snapshot a disk and give this to another user, the icons will not be arranged correctly for them.

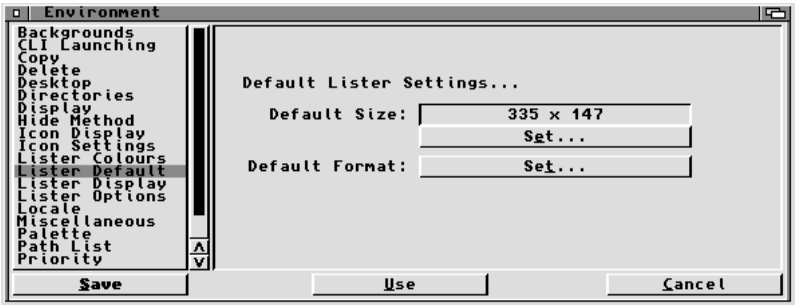
Lister Colours

The 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.



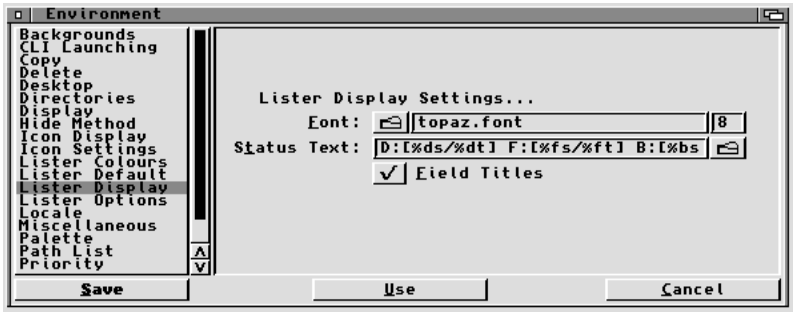
☞ *When setting the colours for the free space gauge, the foreground colour is used when the bar shows less than 90% full, and the background colour is used when the bar is at 90% or above. This lets you have a warning colour when the disk is getting full!*

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. New settings include:-

Allow file select on source activation: Traditionally, when you clicked in a Lister to activate it as a source (SRCE) Lister, if you clicked on a file, that file would also become selected. With this new option turned **on**, Opus will behave as before.

With this option turned **off**, when you click on a file in a Lister that is ***not*** the source Lister, the Lister will be set to source but the file will ***not*** be selected. ***Note that this option only has effect if there is more than one Lister currently open.***

Window zooms to title bar: For many users, the traditional 'zoom' gadget is less than useful so we have provided a mechanism to change its function. With this option set, the zoom gadget causes the Lister to shrink down to just the title bar. This is often much more useful than fully iconifying the Lister since it provides a smaller visual footprint which is easy to move around the desktop.

Inline Editing: Enables inline editing of the name and details in name mode Listers. You can perform the equivalent of Rename, Comment, Date and Protect functions directly in the Lister without having to bring up a requester.

The *cycle gadget* controls how the activation of inline editing. **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 *ESC* to cancel. *SHIFT-Del* and *SHIFT-BACKSPACE* delete from the cursor to the end of field or start of the field respectively.

Miscellaneous

A new section derived from assorted settings from the old Options area. New are:-

Thin gadget borders: Turning this on will make all Opus gadgets (except the ones provided by GadTools) use single-pixel vertical borders, to improve the appearance on 1:1 ratio screens.

Popup Delay: You may now control how quickly popup menu sub-menus open by setting the value in 50ths of a second. The default is 10 or about a fifth of a second.

Maximum 'Open With' Entries: 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 can control the maximum number of items in this menu with this setting.

Custom Screen Title: Many users have traditionally used various third party utilities to customise the text display in the screen (Workbench/Desktop) title bar. Some of these utilities caused problems when used with Opus so we have now incorporated this feature internally in Opus itself. The screen title can now be extensively customised using a simple text string with command arguments. You may choose from the following command sequences:-

%**cp**	CoProcessor type
%**cs**	Graphics chip set
%**dv**	Directory Opus version
%**ev%**	sequence Environment variable (Note)
%**fc**	Free chip memory
%**ff**	Free fast memory
%**fm**	Free memory
%**fp**	Free Pen Count
%**kv**	Kickstart version
%**pm**	Phase of the Moon
%**pr**	Processor type
%**pu**	Processor usage
%**sc**	Screen Count
%**ta**	Task count
%**tc**	Total chip memory
%**tf**	Total fast memory
%**tm**	Total memory
%**uc**	Chip memory in use
%**uf**	Fast memory in use
%**um**	Memory in use
%**wv**	Workbench version
%**%%**	character

Note the special Environment variable. This allows you to expand the Opus display to include practically any string or value. This sequence reads the named variable and displays the result. Usage is %ev<name>%. For example,

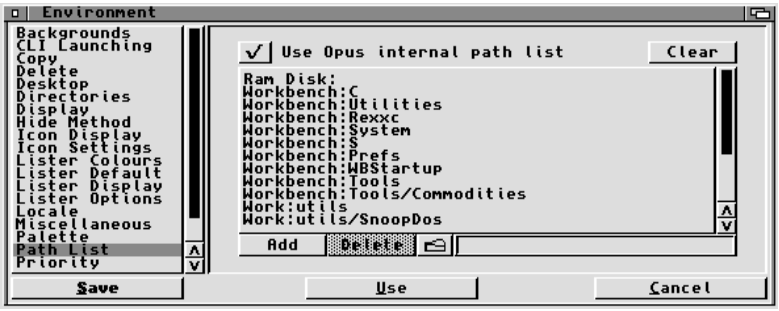
%evWorkbench%.

The memory fields can be further customised by following them with these codes:

K	Kilobytes, to one decimal place
k	Kilobytes, no decimal places
M	Megabytes, to one decimal place
m	Megabytes, no decimal places
S	"Smart" mode (bytes, kb, mb, depending on the actual amount), one decimal place
s	Smart mode, no decimal places
%	Percentage of total memory (free or in use memory only)
%cs	reports P96 for Picasso96 and CGX for CyberGraphics screen modes

For example,

Opus v%dv Memory %fm%%% free, %um%%% in use
CPU %pu%% %ta tasks



Path List

The Amiga has a rudimentary system of setting the system search path which is searched when you attempt to run a program. Normally, paths are set by *S:User-Startup* when you boot your system. Like Workbench, Opus inherits the path list from the initial CLI from whence it is run. However, depending on exactly what boot utilities you are running, paths within Opus can be incompletely defined. Specifically, if you are running one of the dual-boot or multi-user systems on the Amiga, paths can be lost.

Now you may configure the path list used by Opus directly, without having to worry about where the path list has been inherited from.

Sound Events

This new section allows you to configure a sound to be played automatically for certain events. The event list is the same as for scripts, but it saves you having to configure a whole script just to play a sound file.

The **Test** button allows you to quickly test the highlighted sound.

The **Volume** of the sound many be set from 0-64 by entering a number or using the slider.

☞ ***Some datatypes do not support the volume command. At least one of the WAV datatypes does not and you cannot alter the volume - it will play at full volume only.***

Exclusive Startup/Shutdown Sounds. If you have a multitude of sound events set to be called when Opus starts up or shuts down, because the Amiga can only handle a limited number of sounds at once, such multiple sounds can interfere with each other or be lost. With this option set, no sounds will be played until the Startup sound has been played, and no sounds other than the Shutdown sound will be played on shutdown.

Random Sounds

As with background pictures, Sound Events supports random sounds by specifying either a directory or a wildcard pattern as the sound filename. Random sounds are only assigned when the environment is loaded, or when a sound setting is modified - they do not change every time they play. If required, you can specify the volume for a sound by setting the comment of the sound file to the desired volume to a number from 1 to 64. *See also ARexx/Dopus Set Sound, ARexx/Dopus Query Sound, Scripts.*

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 esoteric aspects 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.

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. But why not use Opus's in-built system to customise the screen title? See Environment/Miscellaneous.

HidePadlock: If set to 1, the padlock title bar gadget will not be added to Listers.

WARNING: Opus also contains a number of other environment variables for internal use. DO NOT SET any environment variables unless advised by GPSoftware technical support. We will not guarantee the behaviour of Opus if you set random variables. Just Don't Do It!

FileTypes

The Opus filetype system underpins much of the functionality of Opus. By providing the ability to identify files by a specific type, it allows custom actions when you double-click on a file, press a RMB over a file and so on. For the Magellan-II version we have made a number of refinements and additions designed to provide faster identification of files as well as more control. Many of these changes are internal. Others include:-

- The internal filetypes for Opus configuration files (buttons, environment, etc) have all been moved to priority 125 so they will be matched ahead of user-configured filetypes.

- A new internal filetype for Start Menus has been added. Double-clicking a start menu will now load it as a start menu and not as a normal button bank.

- A new internal filetype has been added for Amiga system font preferences files. If you double-click on a system font preferences file, Opus will set its own internal fonts (icon font and screen font if on its own screen) to the settings in the preferences file. (This is complemented by the new *LoadFonts* command.)

- Additional filetype actions have been added 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&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. This was done to avoid having to add 6 new commands. For example,

  ```
  Command  User 5
  ```

  instead of:

  ```
  Command  User5
  ```

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

- You can now have filetype actions defined for *control* and *alt drag&drop*, and *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,NAME/F

- A new **Search Range** filetype matching command has been added. This is similar to the **Search For** command, except that it limits the range of the search. The first parameter on the line is the maximum number of bytes to search. This must be followed by a space, and then by the text to search for as normal. Searching takes places from the current position within the file. For example,

  Move To 512
  Search Range 1024 hello

  This would move to position 512 and then search the next 1024 bytes for the string 'hello'.

- A new filetype command, **Match Chunk** has been added. This is similar to **Find Chunk** except it only tests the ***next*** chunk in the file for a match. It does not search the whole file. This command can dramatically speed up matching of IFF files if you know the order in which chunks appear.

- The {**Ql**} parameter now works when used in a filetype popup menu function. {**Ql**} also now works through the filetype command system.

- It is now possible to add your own separator bars in filetype popups by specifying a label of three minus signs ("---").

The Disk Filetype

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.

Normally an icon provided in a filetype will only be used if the disk does not have an associated disk.info icon. The **override** switch lets you define an icon for a disk type that will override any real icon on the disk if there is one. For example,

Disk DF0: **override**

You can match disk names as well as device names. This is best used in combination with selective priorities to provide the correct matching order. For example:-

Disk CD0: **override** (set to priority 0)
Disk CD0:CDDA **override** (set to priority 1)

The first filetype would provide an icon for all data CDs, and the second would provide a different icon for audio CDs only (assuming audio CDs are labelled CDDA by the filesystem).

You may specify a device name using the question mark character '**?**' as a wildcard (For example:-

Disk DF?

Scripts

The Opus scripts system provides the ability to monitor certain system events and execute an Opus function or command batch when the event occurs. In Magellan-II, we have expanded the Scripts system with more user features plus we give you the ability to globally replace many internal Opus commands with relative ease. (This could be done in earlier versions but only by using ARexx.)

Command Replacement Scripts

To support this feature we have added several new internal script events for:-

AnsiRead	HexRead
IconInfo	Play
Print	Read
Show	SmartRead

These scripts let you override the internal commands of the same name. For instance, you could define a function for the **Show** event so that whenever the internal command **Show** is called, your function would be run instead. A typical example of this may be on an RTG system where you have a preferred viewer which displays pictures on the default screen in a window.

Custom Scripts

To provide for future expansion, we have added the new ability to define additional scripts for integration with the Opus script interface.

To add a script, create a file in the *DOpus5:System/Scripts* directory. The name of the file does not matter, but it must have a '**.scp**' suffix. The file can contain an unlimited number of script names, one on each line. The format of the file is:

<Script Name>,<Flags>

Flags is optional. At the moment, the only flag defined is 1, which causes Opus not to lock the current source Lister when the script is executed. For example,

```
FTP Finished,1
FTP Connect,1
```

If you have added or edited script files, you will need to restart Opus for the new script definition file to be noticed. When you do, the names of your scripts will appear in the script configuration list.

☞ ***The new OpusFTP scripts you see in the Scripts editor are actually added in this manner.***

There is also a ***new ARexx command*** for triggering both internal and custom scripts.

dopus script <name> <data>

<*name*> is the name of the script (not case sensitive), and <*data*> is an optional string that is passed to the script function in the {*Qa*} parameter. *See ARexx.*

Commands and Functions

A number of new internal commands have been added to Opus and a number of commands have additional features.

☞ ***Remember a summary of the command details is provided for quick access in the updated context sensitive DOpus Help guide. Press Help from within Opus itself or editor.***

Command Replacement Scripts

The following commands are supported by a new internal script event which lets you override the internal command. If you define a function for the any of these commands, whenever the internal command is called, your function would be run instead. The commands and templates are as follows:-

Ansiread	NAME/F, WAIT/S
Hexread	NAME/F, WAIT/S
Iconinfo	NAME/F, WAIT/S
Play	NAME, WAIT=SYNC/S, QUIET/S, ICON/S, VOLUME=VOL/K/N
Print	NAME/F, WAIT/S
Read	NAME/F, WAIT/S
Show	NAME/F, WAIT/S
Smartread	NAME/F, WAIT/S

See also Scripts

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 could be a problem in cases where you wished 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, we have added a **WAIT** parameter to the above commands. If you specify this argument, the command will operate synchronously and will wait until it has finished before returning. Taking the above example,

```
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 so the *Delete* command will function as expected.

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 sub-system 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.

New or Changed Opus Commands

Buildtheme AS=TO

See Themes.

Closebuttons NAME,ALL/S,ICONIFY/S,START/S

The CloseButtons command now looks for the name of the button file as well as the name of the button bank. This means that CloseButtons can now take exactly the same name as LoadButtons.

The CloseButtons command can now close Start Menus with the use of the **START/S** switch. The name you provide is the name of the 'menu' as set in the first column in the menu editor. You can also use the ALL/S switch to close all Start Menus.

Converttheme FILE=FROM, AS=TO

See Themes.

Copy NAME, TO, QUIET/S,UPDATE/S, MOVEWHENSAME/S, NEWER/S
Copyas NAME, NEWNAME, QUIET/S, MOVEWHENSAME/S

Remember, if the **UPDATE** switch is set, only files which do not exist will be copied. The **NEWER** switch is similar to the **UPDATE** except will only copy files that either do not exist or which have a later datestamp than the existing file. This is very useful for recursive copying of directories for backup purposes. You can also use this with ***OpusFTP*** for web maintenance.

☞ ***OpusFTP also provides a direct mechanism for you to override the normal copy command operation to force newer or update. See FTPOptions***

When specified, **MOVEWHENSAME** 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.

Converttheme

See Themes

Fixicons **FILE/M/A, ALLOWNOBORDERS=ANB/S, ALLOWNOLABELS=ANL/S, NOFIXOFFSET=NFO/S, SYNCWBTOOPUS=SWO/S, SYNCOPUSTOWB=SOW/S, REPORT/S**

Sometimes Opus and Workbench will display icons differently where some icons are shown in different positions or either with or without borders or text. This command is designed to adjust the icon structures to fix these occasional differences between the behaviour of Opus and Workbench.

The problems can be caused accidentally if a drawer (or disk) icon has an offset within the drawer data structure. Opus assumes the icons are displayed from 0,0 in the drawer window. If the drawer window (icon) had a non-zero offset then you could see different placement between Opus and Workbench.

This command takes a path or list of files from a SRCE Lister - usually from the root directory of a disk, and recursively adjust the icons for the drawer offsets.

It also cleans up the icon structure of those rogue icons which have been created incorrectly and may have randomly set the bits which Opus uses for Icon Text and Icon Border.

To run the command, either

Use the command **Fixicons** <device> (i.e. FixIcons WORK:)

or

Open a SRCE Lister and highlight the directories or files you wish to fix and use the Opus command*Fixicons*, either from a button or from the OpusCLI. See the note below on root directory icons.

The command will look for a *disk.info* icon in the chosen directory.

By default you should not need to use any of the parameter flags. Just call FixIcons on the root directory of the offending device.

☞ ***FixIcons only looks for '.info' files. In the root or current directory, you must select the individual '.info' files as well as the directories if you wish these to be updated as well.***

Technical details:

If an icon file is selected, the function will operate on that file. If another file or directory is selected, the function will try to obtain an icon for that file or directory. Additionally, if a directory is selected, the function will enter directory recursively and perform actions on all contents.

If the *Disk.info* file is one of the selected files, *FixIcons* reads this first and uses the offsets of the container boundaries to adjust the positions of other selected icons in the root directory.

The function acts on all selected file and directory icons recursively, making the following changes to each icon:-

- Resets the icon's internal border flag to display borders. If **ALLOWNOBORDERS** switch is specified the border flags will be turned off.

- Turns on the icon's label unless the override switch **ALLOWNOLABELS** is used.

- Resets the position of drawer/disk icons to 0,0, and offsets all icons within that drawer by the values of this icon's position fields. For example, if dd_CurrentX = 5 and dd_CurrentY = 3, all icons within the drawer would be shifted -5,-3, and the two dd_Current fields would be set to 0. (Unless **NOFIXOFFSET** switch is set.)

- If the **SYNCWBTOOPUS** switch is set, the Workbench position field within the icon will be set to the same coordinates as the Opus position field.

- If the **SYNCOPUSTOWB** switch is set, the Opus position field within the icon will be set to the same coordinates as the Workbench position field.

- If the **REPORT** switch is set, the full details of all icons that were changed and the reasons they were changed will be displayed in the Opus text viewer.

Getsizes FORCE

GetSizes is now accurate on directories up to 4Gb. Getsizes is now also supported in FTP Listers.

See also OpusFTP

Iconinfo NAME/F,WAIT/S

The Iconinfo command which invokes the iconinfo.module and associated icon information requester has been expanded to provide more functionality plus better support for Drag and Drop of both conventional and NewIcon images and icons.

See Icon Information in Icons

Loadbuttons NAME, START/S, LABEL/K, IMAGE/K, UNDERMOUSE/S, TOGGLE/S

Two new switches have been added to this command:-

- **UNDERMOUSE** causes the button bank to be opened under the mouse pointer

- **TOGGLE** will toggle the state of the button bank - if it is already opened, it will be closed, otherwise it will be opened.

The **START** and **IMAGE** switches are for support for Start Menus from previous version of Opus. These are effectively redundant but we have left them for compatibility.

Loadtheme FILE=FROM,APPLYPALETTE=AP/S, APPLYFONTS=AF/S, APPLYBACKGROUNDS=AB/S, APPLYSOUNDS=AS/S

See Themes

Play **NAME,WAIT=SYNC/S, QUIET/S, ICON/S, VOLUME=VOL/K/N**

A **VOLUME** parameter has been added to the PLAY command. It takes a value from 0 to 64.

See also Scripts/Internal

Rename **NAME, NEWNAME**

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.

Savetheme **AS=TO**

See Themes

Select

There was a problem whereby the internal *Select* command would prevent any other commands that followed it in a function from working (eg *ScanDir/Select/Delete* - the *Delete* would not get executed). This has now been fixed.

SET

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

Setasbackground NAME, DESKTOP/S, LISTER=WINDOW/S, REQ=REQUESTER/S, TILE/S, CENTER/S, PRECISION/K (Only OS3.0 or greater).

The *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 switches are:-

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).

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

Stopsniffer

If you have included the '*File Type*' or '*Version*' parameter as part of your Lister format, Opus will start a background task to identify all the files in the Lister. (Note: This process happens in both Name and Icon modes even thorough you may not

visually see the effect.) Examining every file can cause excessive disk activity, especially on slower CD drives. The normal solution to this problem is to NOT use '*File Type*' or '*Version* ' as a general format specification for all Listers but to use it selectively only when required.

However, for those who do wish to include these format parameters, we have added this special command, *Stopsniffer*. It can be used stop the filetype/version sniffer process for the current Lister. Add this to your toolbar if you want a way to interrupt the sniffer on large directories/files.

Function Editor

The global function editor is used when you wish to enter commands and functions for buttons, start menus, menus, scripts and all Opus objects. We have augmented the editor as follows.

Argument Variables

Two new command sequences have been added to functions:

{**QL**} Process address of the current source Lister (in hex).
{**QD**} Process address of the current destination Lister (in hex).

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:-

{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 **'** (single quote) or **~** (tilde) 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 **-** (minus) 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'-}

Icon Files

Icon '*.info*' files without an associated file are now accepted by the **{f}** and related argument variables in functions.

Workbench commands

Workbench commands in Opus objects such as buttons, menus etc, can now be made asynchronous by simply inserting the word '*run*' before the command. For example,

Workbench sys:tools/calculator

is *synchronous* and subsequent commands will be blocked until this process returns. However,

Workbench run sys:tools/calculator

is *asynchronous* and subsequent commands will continue immediately.

ARexx Commands

The ARexx command functions of Opus allow you to write custom scripts to control Opus behaviour or to create complete programs which complement Opus and use the Opus in-built features to perform a wide variety of external tasks. There are many such scripts and these are usually stored in the *DOpus5:ARexx* directory. The excellent***ArcDir*** program which enables the display of archives is an example of the use of Opus ARexx capabilities. This and many other scripts have been extensively revised and have been supplied in the Magellan-II update installations.

The following is a summary of the changes for the ARexx command interface for Opus Magellan-II. For a full explanation of the Opus ARexx commands and interface, please refer to the ARexx tutorial and guide on the DOpus Plus CD or available from the GPSoftware web site.

New and Enhanced ARexx Commands

Many new ARexx commands have been added and others have been enhanced to support more functionality required by the OpusFTP system plus a number of third-party scripts developed by Opus users. The changes are a follows:-

- **command**

  ARexx commands using the "command wait <cmd>" function now return a result code indicating success or failure.

- **doubleclick**

  The ARexx command *command doubleclick* was not sending doubleclick messages to the custom handler if the source Lister had one. This has been rectified.

- **dopus command**

  This command has been greatly expanded. It provides the ability to add new internal commands to Directory Opus, or

to replace existing commands. It is generally called from within the init function of an Opus ARexx module and the program parameter will be the name of that module, without the "*.dopus5*" extension.

The new template is:-

dopus command <name> program <scriptname> [desc <description>] [template <template>] [source] [dest] [private] [help <help file>] [handler] [temp] [ext <menu name> type <filetype>] [remove]

The *program* field is mandatory, and Opus will run the script name you provide here whenever this function is invoked.

Use the *name* parameter to specify what the command should be called, and, optionally, use the *desc* parameter to define a description which will appear in command lists.

Using the *handler* parameter will allow the item to be displayed only if a custom handler matching the *program* parameter is present for the Lister.

The *remove* flag allows you to remove any command specified with name.

temp will allow you to add a 'temporary' command, which does not have an external command file. This command will then do nothing unless it is trapped with the dopus addtrap command.

Use the *remove* flag of the dopus command to remove it when you are done. Using the *private* flag will stop the temporary command from being seen in the internal command list.

The template is not parsed for you in any way - it is simply shown to the user when they request a template for your command. It is up to your script to handle the argument string sent to it.

If you give the *source* or *dest* keyword (or both), your command will not run unless there is a Lister of the given

type. When run, the Lister will automatically go busy, and the *source* or *dest* argument given to your script will contain the Lister's handle. (Each argument still exists when the relevant keyword is not given, but is always "0").

Note that the standard ARexx module example code parses the source Lister handle into a variable called source and the destination handle into dest. Be careful that when you specify the *source* keyword for dopus command that you put it in quotes (*"source"*), otherwise the contents of the source variable may be used in place of the actual word "source". Take similar precautions with the "*dest*" word.

You can specify a help file for the new command with the *help* parameter.

Wildcard Patterns

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.

PopUpExtensions

You can add a command to the pop-up menu of icons of a given type, or to the Lister itself, by specifying a filetype name or ID for the type parameter and the string to appear as the menu item for the ext parameter.

To match more than one filetype you can specify the type parameter multiple times (but you can only have one ext). You can also specify one of the following keywords for the type parameter: all, disk, drawer, tool, project, trash, baddisk, leftout, lister, lister2.

Specifying *lister* for the type parameter will add items to the Lister pop-up menu rather than the icon pop-up menu.

Specifying *lister2* for the type parameter will add items to the pop-up menu available from the SRCE/DEST display in the Lister status bar.

If you add menus to a filetype at priority -124, they will only be shown if no other filetype matched. This allows you to have a "default filetype" with menus that will only be shown if no other filetype menus are displayed.

Commands which are intended for pop-up menus are unlikely to be of general use. You can use the private keyword to hide the command (it will not appear in any list shown to the user). (This is not restricted to PopUpExtensions.)

See ARexx help file and the example scripts for more information.

- **dopus clear <item>**

  Lets you clear a setting (equivalent to doing *dopus set* with a null string). For item use

  background [desktop | lister | req] sound <event>

- **dopus desktoppopup**

  This function triggers the desktop popup menu at the current mouse position, and return a value indicating the user's selection.

  The function is called with a single value - a flags parameter which indicates which items in the menu are to be disabled (by default, all are available). Add the flag values as appropriate.

Add to disable	Item	Returns
2	Create Left-Out	1
4	Copy To Desktop	2
8	Move To Desktop	3

The function returns 0 if the user cancels the operation.

- **dopus getdesktop**

  The *dopus getdesktop* command now returns the full path of the desktop folder rather than the actual configured path (which could be an assign).

- **dopus matchdesktop**

  This commandwill match a supplied path against the desktop folder for you. For example,

  ```
  dopus matchdesktop 'DOpus5:desktop'
    --> 1
  ```

- **dopus progress**

  Two new parameters, *info2* and *info3* have been added to the *dopus progress* and *lister newprogress* commands. With these Opus now supports three information lines in the progress requester.

- **dopus query <item>**

  Dopus query is the partner of *dopus set* and lets you query the current settings of the various items.

- **dopus query background [desktop |lister |req]**

  This new command lets you query the currently set background picture.

- **dopus query sound <event>**

  This new command lets you query the sound events. For example,

  dopus query sound Startup

- **dopus read <pos>**

  dopus read is mostly previous documented, however there are three new flags - hex, ansi, smart - which control the text viewer mode. For example,

  dopus read hex 's:startup-sequence'

  There are actually three already existing undocumented commands to achieve the same effect:

  dopus hexread
  dopus ansiread
  dopus smartread

  However, the flag for the *dopus read* command is more logical.

  The optional <*pos*> flag allows you to specify the position of the text viewer. For example,

  dopus read pos 10/10/400/200 s:startup-sequence

- **dopus refresh <item>**

  Lets you refresh the display of certain items:

  all - refresh the entire display of everything
  background- already documented
  icons - refresh icons in all Listers and on the desktop
  lister [full] - refresh Lister displays

  The various refresh commands are needed after modifying some settings with the *dopus se*t command.

- **dopus refresh**
  dopus refresh background

  Setting the background picture with the *dopus set background* command does NOT refresh the display. The display will not be updated until you call the "dopus refresh background" command.

  There is also a new keyword *custom* to both this command and *dopus set background*'. Using this keyword, you can change the background pictures internally without modifying the actual environment settings. You must use the *custom* keyword on both the *set* and *refresh* commands for this to work. As soon as you do a *refresh* without setting this keyword, the pictures will revert back to the environment settings.

- **dopus set <item> <parameters>**

  Allows you to configure certain items of the DOpus environment. It is primarily used at this stage for the themes system. Item may be one of:-

  background - This new ARexx command has been added to manipulate background pictures:

  > dopus set background <file> [desktop | lister | req] [center | tile] [precision <precision>]

  This sets a new background picture for the specified place (desktop, lister or requesters). Precision, if supplied, can be: none, gui, icon, image or exact.

  There is also a new keyword *custom* to both this command and *dopus refresh backgroun*d. Using this keyword, you can change the background pictures internally without modifying the actual environment settings. You must use the *custom* keyword on both the *set* and *refresh* commands for this to work. As soon as you do a *refresh* without setting this keyword, the pictures will revert back to the environment settings.

sound - This is a new ARexx command to let you set the sound events. The format is:

dopus set sound <event> <file> <volume>

For example,

dopus set sound "Open Lister"
DOpus5:sounds/open_Lister.snd 64

palette - Configures the DOpus palette. Accepts a string of up to 16 hex values; the last 8 represent the Opus colours, and the first 8 represent the system colours when Opus is on its own screen.

For example,

dopus set palette 0xffaabb 0x000055

pens <type> - Configures the pens numbers used for various things where <type> is a string signifying the pen set to alter, and is followed by a number of values which represent which of the pens to use. <type> can be:

icons	<dfg><dbg><dstyle><wfg><wbg><wstyle>
files	<fg> <bg>
dirs	<fg> <bg>
selfiles	<fg> <bg>
seldirs	<fg> <bg>
devices	<fg> <bg>
assigns	<fg> <bg>
source	<fg> <bg>
dest	<fg> <bg>
gauge	<normal> <full>
user	<count>

The pen number is mapped in the following way:

1-4	bottom four system colours
5-8	top four system colours
9-16	Opus user pens

font <type> - Configure the fonts where <type> is a string representing the font to be configured:-

screen	Screen font when Opus is on it's own screen
listers	Font used for file display in Listers
iconsd	Icons on the desktop
iconsw	Icons in windows

<type> is followed by the font name and size, eg

dopus set font screen courier.font 13

- **dopus script**

To trigger a script (either an internal script or a custom one), use this command:

dopus script <name> <data>

<*name*> is the name of the script (not case sensitive), and <*data*> is an optional string that is passed to the script function in the {**Qa**} parameter.

- **lister new**

The *lister new* ARexx command now takes several new parameters:

inactive	Don't activate the Lister when it opens.
invisible	The Lister is created but not shown. Use *lister set visible* to make it visible.
iconify	The Lister is opened iconified.
fromicon	Will open the new Lister using the size and position information from the specified directory's icon (only if a path is specified and an icon exists). For example, *lister new fromicon sys:tools*
mode	Lets you specify the initial mode of the new Lister. For example, *lister new icon* or *lister new action showall*

- **lister clearcache**

  Will flush any caches that were created by your Lister, using your custom handler. No other caches will be affected.

- **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 query active**

  There commandreturns the handle of the currently active Lister (ie the window is active, the Lister is not necessarily source or destination).

- **lister query proc**

  The *lister query <handle> proc* ARexx command will return the process address of the Lister (returned in RESULT in decimal).

- **lister query window**

  The *lister <handle> query window* command 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 query title**
  lister query header

  We have added these commands to complement the *lister set* commands.

We have also added *lister set value* and *lister query value* ARexx commands. These allow you to associate your own data with a Lister in the form of name/value pairs. The Lister will maintain the values until it is closed. For example,

```
lister set <handle> value MyName "Fred Bloggs"
lister query <handle> value MyName
lister clear <handle> value MyName
```

- **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 datestamp (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 set**
  lister set handler

  A new switch for the *lister set handler* ARexx command called '*leavegauge*' causes the fuel gauge to remain if it was already there, and leave it absent if it was not.In contrast, the old '*gauge*' switch will always add a fuel gauge and not providing either switch will always cause the Lister to have no fuel gauge.

  Another new switch, *'synctrap*s' causes messages trapped by ARexx Lister handlers to be synchronous. This is extremely useful when writing scripts.By default (and for compatibility) handler messages remain asynchronous.

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.

- **lister set newprogress**

  Two new parameters, '*info2*' and '*info3*' have been added to the newprogress command. With these Opus now supports three information lines in the progress requester.

- **lister set commentlength**

  This command complements the *'lister set namelength*' command and allows you to set the maximum comment length allowed in a Lister. Note that this only affects the length displayed and does not affect the maximum comment length that can be saved to disk, which is fixed at 79 characters. Its prime purpose is to support ARexx scripts and custom handlers where you may wish to use the comment field for storage of text longer than 79 characters. The FTPBatch download system being developed by third party sources is an example of this.

ARexx Notes

For file Listers, the popup menu **CopyTo** has been expanded so you can now copy to ARexx scripts. To do this,

Create an ARexx script in the *DOpus5:system/CopyTo* directory. Set the comment of the script file to be the name that you want displayed in the menu. Then 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 ARexx guide from the DOpus Plus CD or from our web site for more details.

ARexx Handlers

If the '*synctraps*' flag is set, messages you get from trapped functions now have a new argument. Arg8 contains the address (in decimal) of the FunctionHandle structure for this function, for use with the Opus callback hooks. This handle is only valid until you have replied to the message.

- The "*drop*" and "*dropfrom*" custom handler messages now contain the word "subdrop" in the qualifier argument field if the drop was into a sub-directory.

- When adding files to a Lister via ARexx/callbacks, you can set the datestamp of the file automatically to the current date/time by passing in 0.

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

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

  ```
  dopus addtrap * myhandler
  ```

- 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.

Opus FTP

Since the Opus Magellan release, the in-built OpusFTP module has proved to be very popular because the integration into the Opus makes FTP very easy to use. This has been further enhanced with the addition of many new features as well as improvements in speed and efficiency. New features include:-

- ***A completely revised GUI*** for **FTPAddressbook**, **FTPOptions** and **FTPConnect** gives control over individual site entries. Tailor custom parameters for individual FTP sites with **FTPOptions** or use **FTPConnect** as a quick alternative to the main address book.

- ***Individual Options for each site*** can be edited and tested while the Lister is connected to the remote site.

- Full support for ***recursive copying*** between remote sites and local directories including:-

  recursive Copy/CopyAs
  recursive Move/MoveAs
  recursive Delete

- The ability to ***store changed entries*** or ***add new entries*** to the main address book directly ***while connected*** with the new **FTPAdd** command.

- ***Enhanced Drag and Drop*** of sites from the address book or files or directories from remote sites onto the Opus Desktop to create shortcuts or direct to Opus objects such as buttons, menus etc to automatically create commands.

- Seven ***new event Scripts*** are available for events specific to FTP such as connect and copy success or failure. They can be set on an individual site-by-site basis.

- ***Automatic reconnect on failure*** or lost connection allows you to control the number of retries and adjust the delay between attempts.

- ***Site-to-Site*** transfers have been improved. Unlike stand-alone FTP programs on the Amiga (and PC!), OpusFTP gives you full recursive site-to-site transfers capability which allows you to seemlessly transfer files directly between two remote sites without downloading them to your Amiga first. ***Just Drag and Drop files between remote Listers!***

- ***A revised FTP command set*** gives significantly better integration of FTP into the normal Opus Lister operation with many more commands now supported. *Getsizes* and *FindFiles* are now fully supported for remote sites. *Protect* is fully supported with a new GUI which allows you to set protection bits in either Amiga fashion or Unix fashion. The copy function now supports *preservation of Datestamp and Protection bits*. *Copy UPDATE* and *NEWER* flags are now supported directly via the site FTPOptions. Great for web maintenance because you can recursively select and copy only those files which are different or do not exist on the remote site.

- ***Inline editing*** of Name and Protection fields now supported.

- A ***new ScanDir FTP*** command allows you to connect to a remote site and directory directly from any Lister.

- Both ***logical and relative directories*** and ***remote links***are now supported.

- The ***AutoIndex displays from AmiNet*** are better supported.

- ***More information*** is provided during the establishment of the connection and we have added the ability to display messages from the remote site for startup and directory information in the Opus text viewer.

- Optional ***transfer and progress displays*** give more detailed information including speed and remaining time for download.

- ***Custom toolbars*** may be used for individual sites.

- *Firewalls* and *IP Masking* techniques are supported with the implementation of ***PASSIVE transfers.***

- Lister path gadgets now accept ***industry standard URL syntax*** such as "FTP://user:password@host:port/path" (all but host optional). These "FTP://" style paths now also work from an FTP Lister so you can easily cut and paste site URLs.

- In a very significant change, all Opus FTP commands are now synchronous so ***interactive ARexx scripts*** are now possible!

- Both ***speed and efficiency*** of the module have been improved through better management of TCP socket options along with improved support for all Amiga TCP stacks.

- Each FTP Lister status gadget ***popup menu*** now contains Options and Add. Respectively, These allow you to customise the FTP options for the current Lister, or add the current site entry details and custom options directly to the address book.

Enhanced Drag and Drop support

We have expanded the drag and drop capability to allow you to do some cool things! These include:-

- ***Drag a SITE*** from the address book ***to the Desktop*** to automatically create an Opus command file (FTPConnect SITE) and icon to quickly connect to that site.

- ***Drag a FILE*** from a Lister ***to the Desktop*** to create a COMMAND which calls the newDOpus5:ARexx / FTP_file.dopus5 ARexx script to access this file (connect to site then download it). Note the action performed is the specific action as defined in your desktop popup setting.

- ***Drag and drop a SITE*** from the address book to a function editor, button editor, menu editor, hotkey editor, filetype editor etc to create an FTPConnect function. This will create the command line automatically containing the correct OpusFTP syntax.

- ***Drag within the address book*** to arrange entries in the list if you desire. Normally an entry is inserted at drop position but you can swap entries by holding SHIFT when dropping entry.

- ***Drag a SITE*** from the address book to an already opened Lister to immediately connect to the site.

- ***Drag and drop*** of files and directories ***into sub-directories*** now works for uploading and downloading (except for site-site transfers where it is not permitted).

Icons for FTP

A set of icons is supplied to support Opus Commands in both standard and NewIcon formats. These icons are used automatically when you drag and drop objects onto the desktop to leave out remote sites, or remote files or directories etc. The icons "*FTPSite.info*", "*FTPFile.info*" and "*FTPDirectory.info*" are used respectively.

By default, the standard set will have been installed into the *DOpus5:Icons* but you may replace these with a NewIcons set if you desire. These may be found in the sub-directory Icons-FTPNI.

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.

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.

☞ ***Some FTP servers do not allow site-to-site transfers for security reasons, or, the ability may just not be enabled. Contact your service provider for details if this ability is not supported.***

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

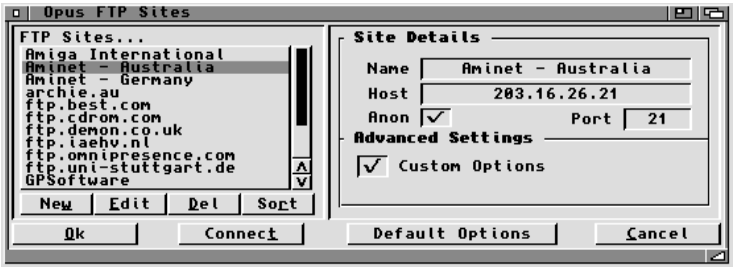
will attempt to connect to the Australian Aminet site in the current Lister.

Long Filenames

The file lengths accepted by the FTP module can now be a maximum of 107 characters. Remember that normal AmigaDos files are limited to 30 characters in length so if you transfer a file with a long name the name will be cropped when it is saved to a local drive unless you are using one of the new Amiga filesystems. *See General / Long File Name Support*

FTPAddressbook

The new address book GUI allows you to maintain your list of favourite FTP sites. Adding or editing of remote entries and associated settings is done through a new FTP Site Editor and independent Options requesters. The address book contains:-



FTP Sites: Your favourite list of FTP sites. These are automatically entered in sorted order but you may re-arrange the order by dragging an item to the new position. You can also drag a site entry out of the Address book. See the Drag and Drop section above for more details.

A double-click on an entry in the site list will either connect to the site or edit the entry depending on the setting in the default options (*See FTPOptions/Global/Sites*)

New and **Edit:** Calls the FTP Site editor, allowing you to add a new entry or edit the highlighted site. This is a multi-threaded process so you can edit multiple sites at the same time.

Delete: Deletes the highlighted entry from the address book.

Sort: Sorts the entries into alphabetical order.

Custom Options: When set, indicates that the selected FTP site has custom settings and does not use the system global values. See the Options editor.

Ok: Closes the address book requester and saves all changes made.

Connect: Connects to the selected FTP site. (Note: a SHIFT click will not close the address book after connect.)

Default Options: Allows you to edit the global option settings which are used by all FTP connections which don't have a custom set.

Cancel: Closes the address book requester and discard any changes made. (The previously saved information will be re-read from disk.)

The Address book Menus

Open: Loads a previously saved FTP address book from disk.

Import: Imports an old address book in ASCII format from an earlier version of OpusFTP and merges the entries with the current address book.

Import AMFTP sites: Converts the site entries from an AmFTP ".AmFTPProfiles" file and merges them with the current entries.

Save: Saves the contents of the FTP address book to disk.

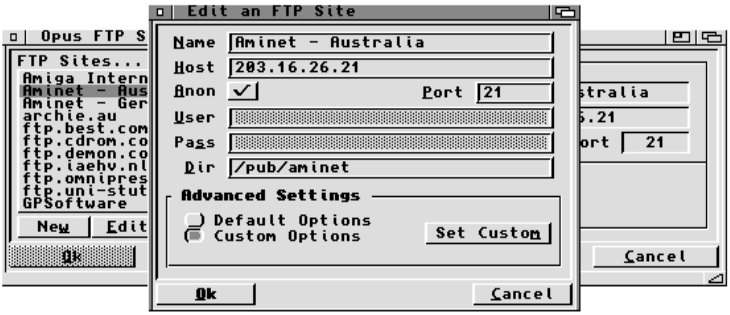
Save As: Saves a copy of the FTP address book to disk with a specified filename.

Export ASCII: Saves a copy of the FTP address book in a simplified ASCII format compatible with earlier versions of OpusFTP. This file may be easily read or edited with any text editor.

☞ ***Since this file contains your remote site details and passwords in readable format, you can readily check a site's details if you have forgotten your password!***

Quit: Closes the FTP address book requester.

FTP Site Editor



Name: A local name for this FTP site. Choose a name which describes the site.

Host: The actual internet address or IP number of the FTP site. For example, for AmiNet you could use 'ftp.livewire.com.au' or '203.16.26.21'

Anon: Allows anonymous connections otherwise you must enter a user name and password.

Port: The TCP port for the FTP service, usually 21 but may be different on some private or restricted sites.

User: For non-anonymous FTP sites, the user name of your account.

Pass: For non-anonymous FTP sites, the password for your account.

☞ ***For security, you may leave this blank. If it is not provided in the address book you will be prompted for it during the login procedure.***

Dir: The initial directory on the remote site. (Can be blank)

Advanced Settings-

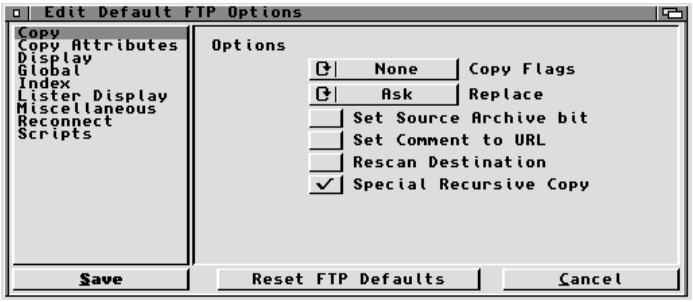
Default Options: The global settings will be used when connecting to this site. (default)

Custom Options: Individual custom options will be used for this site. Selecting this will open the [options editor].

Set Custom: Displays the Options Editor for you to edit the custom options for this site.

FTPOptions

In order to provide more control over individual settings for each remote site, we have separated the older configuration settings from the address book and added a new FTPOptions command with an associated GUI editor.



Save: Saves the current options and closes the requester.

Reset to Defaults: Displayed in the site custom options version of the requester, it resets all options for this site to those as defined by the current global FTP options settings.

Cancel: Cancels any changes made and closes the requester.

Copy Options

Copy Flags: Allows you to modify the behaviour of the copy command under all cases. Choices are:-

None: Copy all files.

Update: Copy only files which do not already exist in the destination Lister.

Newer: Copy files which do not already exist in the destination Lister or which are newer than the destination files. Since these functions also act during recursive directory copying, they help you to easily maintain a remote FTP site.

☞ ***These additional parameters are also available with the normal trapped Opus COPY command but are provided in the GUI as overrides for ease of use. See the COPY command.***

Replace

Ask: Always ask whether to replace existing files.

Always: Always replace existing files without asking.

Never: Never replace existing files and don't ask.

Set Source Archive Bit: Mark the source file as having been archived after copying it. This is only possible when uploading from your computer to an FTP site.

Set Comment to URL: Add a comment field to all files copied indicating where they were obtained. This is only possible when downloading files from an FTP site to your computer. Note that the URL will be trimmed at 79 characters.

Rescan Destination: Causes the FTP Lister to be re-scanned automatically after each upload or site-to-site transfer. This ensures that all protection bits, dates, and sizes are accurately shown but will result in slower performance especially if you are using a slow FTP site.

Special Recursive Copy: Some FTP servers cannot handle a LIST command if the directory name contains spaces. Set this flag to overcome this problem if needed when using recursive copying of directories.

Copy Attributes

Also Copy Source's... Some file attributes may also be copied similar with a normal Opus copy function.

You may 'Use Opus Settings' as currently defined in the main Opus program, or you may chose the individual settings for:-

Datestamp: Copy the last modified time and date of files. This is only possible when downloading files from an FTP site to your computer.

Protection Bits: Copy the file protection bits of files. This is possible when downloading files from an FTP site to your computer and may be possible when uploading or doing site to site transfers. Note that the protection bits will be approximated depending on the operating system of the FTP site computer. See also the Protect command.

Comment: Copy the comment associated with the file if there is one. This is only possible when downloading files from an FTP site to your computer. FTP site comments can come from two sources: If the file was a link on the FTP site it's comment may contain the full path of the file. If you have enabled the 'Index' option and the directory has an index file, there may be a description of the file in the comment field. Note that index file comments are not supported during recursive copying. See also the Set Comment to URL option.

☞ ***For Copy Source's and Set Source Archive Bit options, if the source Lister is a local one the FTP settings are used. For site-site copy, the source is used.***

Display

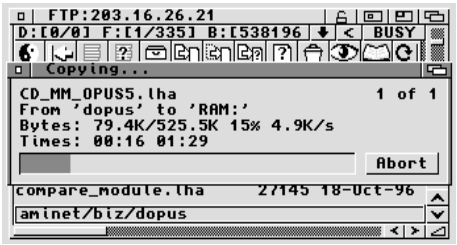
Traditionally with OpusFTP, if you wished to see the messages sent by the remote site during a connection, you activated the Log Option. Now you may selectively view only certain, more important types of messages:-

Show Startup Messages: Many public access sites have a startup message that contains useful information. This option causes such information to be displayed in the Opus text viewer when you connect to a site.

Show Directory Messages: Some sites present a short information message, a description of the directory's contents or some other relevant information each time you change to a new directory. These will be displayed in the Opus text viewer. Subsequent messages will be displayed in the same viewer if it has not been closed.

Progress Windows (CD & List): By default, OpusFTP displays some extra progress requesters not seen in normal Opus. These provide information on the progress of the specific action. Such messages are displayed while changing directories and while scanning the lists of files. On fast sites you may wish to turn these off.

File Transfer Progress Display: Enables a multi-line display when transfering files extra details of file size, bytes counts, speed and elapsed and remaining time for the transfer.



Global FTPOptions DEFAULT ONLY

Enable Log: Opens a log file to record all the output from the FTP site(s).

Enable Debug: Enables verbose log display showing all commands sent to the FTP site as well as numeric responses and error numbers in the log, error requesters, and text viewer.

Log File: Specify the name of a global log file. By default the standard Amiga console is used so the output can be viewed instantly but you may specify a disk file instead.

Sites controls the behaviour of the mouse functions in the main address book display. Choose from:-

Double Click: Allows you to choose the action when double-clicking on a site entry in the address book. Either, Connect to the site or Edit the entry.

Auto Close: Will close the address book automatically when you connect to an FTP site from the connect button or by double-clicking on an entry.

Anon: Allows you to set the username when logging on to an anonymous FTP site. Generally your email address is used. By default, OpusFTP attempts to set this based on the host and user details from your TCP/IP stack.

Index

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 **Download 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.

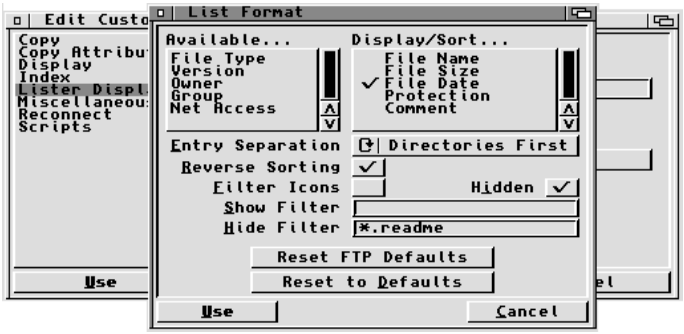
With **Auto** selected, only files smaller than the **Max Size** will be downloaded. If the index file is larger than this value it will be ignored.

☞ ***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.***

Lister Display

Toolbar: Each FTP site in your address book may have its own custom Lister toolbar. Leave blank to use the default, or enter the name of an Opus toolbar. Generally a special toolbar for the FTP is not used but the file 'DOpus5:buttons/toolbar_ftp' will be looked for and used for all Listers if it is present.

FTP Lister Format: As part of the custom settings,each FTP site can have its own special format defined for each site. The tick gadget shows whether one is currently defined. Formats for FTP Listers support ONLY Name, Size, Data, Protection and Comment fields. It is not possible to support the other fields and the *Free Space Gauge* and *Inherit* options are not supported.



☞ ***The HIDDEN setting in an FTP Lister has a special usage. All '.' (DOT) files automatically have the hidden bit set when they are added to the Lister. Since these files are usually system files of little interest, this is a quick way to hide these from the more interesting files in the Lister.***

In the custom Format requester, **Reset to Defaults** reset the format to current Opus default format and **Reset FTP Defaults** resets the format to that set in the global default FTP options format (if any).

Miscellaneous

Keep Last Directory: This will save the current directory name in the address book each time an FTP Lister is closed. Next time you access the site through the address book you will automatically be taken to this directory.

Passive Transfers: Some FTP sites may require you to use passive (PASV) transfer mode. This may also be required for certain types of firewalls and when using IP Masking.

Technical Note: This command requests the server-DTP to "listen" on a data port (which is not its default data port) and to wait for a connection rather than initiate one upon receipt of a transfer command.

Special Directory Names: This option changes the internal parsing of the list format to take into account directory names beginning with one or more spaces. We have not found any problems with this permanently selected, but we have provided the option just in case. Usually, leave it off unless you have a problem.

Technical Note: Traditionally, the listing of a remote directory is provided in a standard format similar to the output from the UNIX LS command, where a space character is used to separate the various fields. One artefact is that if a directory name actually starts with a space, the space character will be stripped. This results in an error if you attempt to access the (now spaceless) directory name.

Network Timeout: The maximum amount of time to wait for a response from an FTP site before giving up and dropping the connection. (Default is 60 seconds.)

List Update: Defines the time between adding and refreshing entries in the Lister. Because scanning directories on FTP sites can be slow, OpusFTP updates the Lister while it is scanning. Set the number of seconds between successive updates. Lower times 'look' better but will reduce performance.

Links:-

Logical Parent Dir: On a remote FTP site, links to directories can cause problems. If you change directory to a link and then change to the parent directory you will not necessarily return to where you started. Normally, OpusFTP assumes that this never happens. When selected, this option ensures that OpusFTP interrogates the site to obtain the actual directory path, despite the link, so that it cannot become confused. There is a small performance penalty associated with this option.

Unknown: It is sometimes not possible to tell if a link on an FTP site is a link to a file or to a directory. Such "unknown" links are normally assumed to be symbolic links to directories. But you can alter this behaviour. 'As Directory' assumes all unknown links are links to directories whereas 'As File' assumes all unknown links are links to files.

Reconnect

These options modify the connection to the remote site.

Enable Retry: Instructs OpusFTP to retry the connection to a site if it failed to connect or login.

Retry Count: Specifies the number of times to attempt a connection login to an FTP site before giving up.

Retry Delay: Specifies the amount of time in seconds to wait between attempts to connect and login.

Auto Reconnect: If a connection is lost due to inactivity, time out, or network error, this option causes the Lister to attempt an automatic reconnection to the site to the directory showing in the Lister.

Because Listers are asynchronous, a lost connection cannot always be detected until you try to use the Lister. If this option is not enabled, a requester will appear allowing you to manually choose to reconnect.

Send NOOPs (Idle): 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 sites inactivity timer from being triggered. By default this should be turned off. Do not use this unnecessarily. Keeping your connection alive when you are not actually using it often prevents other users from accessing the site. It is considered unfriendly by some ISPs. Note that some FTP sites are smart enough to disconnect you even when using this option.

Scripts

OpusFTP now provides a set of scripts for specific events during an FTP session. These can be customised to perform practically any function just as with normal Opus scripts, however the FTP scripts can be enabled and disabled on a site-by-site basis and on a script-by-script basis. All scripts are called with the Lister handle of the Lister which invoked the script. This can be useful with ARexx.

Connect Success: Activated when a connection is established to a remote site.

Connect Fail: Activated when a connection and/or login to an FTP site has been unsuccessful. This script is not triggered until all retries have been exhausted.

Copy Success & Copy Fail: Activated when a Copy / CopyAs / Move / MoveAs operation has been completed successfully or not, respectively. As an added feature, these scripts are only activated if the operation has taken ***more*** time than the minimum specified in the Activation Time field below.

Error: Activated whenever any error occurs.

Close Connection: Activated when FTP connection is closed.

Activation Time: This field applies to the Copy scripts. It species the minimum amount of time that must have elapsed before those scripts are activated. This is useful if you want an alarm bell to sound when a large or recursive copy has completed but not when copying single files.

Options Menus

Project menu:-

Open: Loads an options file and applies the settings to this specific site, or, from the default options editor, loads the a new default set of global options for all FTP sites that don't have their own individual settings.

SaveAs: Saves the options for this FTP site to a file, or, from the default options editor, saves the global options settings.

Quit: Cancels any changes made and closes the options editor.

Edit menu:-

Reset to Defaults: In the site options version, resets all options for this site to those defined by the global options settings.

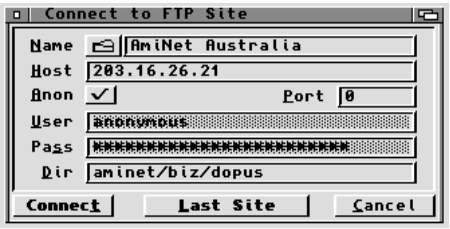
Reset FTP Defaults: In the default options version, resets the global options settings to default factory settings.

Last Saved: Restores the options settings to what was last saved.

Restore: Restores values to when the requester was called.

FTPConnect

A new connect requester has been added to give quick access to a new FTP site or a known site without going through the full address book interface.



Name: If the host field is blank, entering a name will open the site list display at the closest matching entry allowing you to select an item. If the host field is not blank, the name is simply used as a handle for this connection. The file button will also open a list of the FTP site entries from the address book.

Host: Enter the internet address or IP number of the FTP site. ***This field is required.***

Anon: Defines the site as anonymous login.

Port: The number for the FTP service port, usually 21.

User: Your user name if not using an anonymous site.

Pass: Your password if not using an anonymous site.

Dir: The initial directory for the remote site. Usually not required.

Connect: Connects to the site.

Cancel: Closes the requester and does not connect.

Last Site: Recovers the details of the last site to which you connected allowing a fast reconnect.

Opus FTP Commands

- **FTPAddressBook**

  Display the address book GUI to connect to or edit sites.

- **FTPConnect HOST, PORT/N/K, USER, PASSWORD=PASS, DIR/K, LISTER/N/K, SITE/K, GUI/S, NOSCAN/S, RECON/S**

  Useful for connecting to sites without adding them to the address book. This command supports the following options:-

  HOST - Internet address or IP number.

  PORT - Port number of site.

  USER - User name for a non-anonymous FTP.

  PASSWORD - The password for anon-anonymous site.

  DIR - The initial directory to display in the Lister.

  LISTER - The Lister *handle* of an existing Opus Lister. Very useful for ARexx scripts.

  SITE - To connect to a site in the address book using its custom settings use this keyword plus the site name as in the address book, e.g. **FTPConnect SITE Aminet**.

  GUI - Causes the Connect requester to always be displayed even when sufficient details have been given to connect to a site without it.

  NOSCAN - Causes the initial directory to not be scanned into the Lister. Useful for ARexx scripts which log into a site to download known files without wasting the time to scan the directory first.

  RECON - Causes OpusFTP to reconnect to the site if the connection terminates unexpectedly.

The *FTPConnect* command now accepts industry standard URL syntax such as:- FTP://user:password@host:port/path (all but host are optional)

- **FTPCommand QUIET/S, LONG/S, COMMAND/F**

  Some FTP sites recognise special commands that OpusFTP doesn't support directly. These are accessible using FTPCommand. A simple example is '*FTPCommand CWD <dirname>*', which may be useful in an ARexx script to change the current directory instead of using ScanDir which would result in a full rescan and display of the directory.

  The results of the command will be shown in a requester. The following options are also supported:

  QUIET - Do not display results.
  LONG - Display multi line responses.

- **FTPSetVar VAR, VALUE/F**

  This command functions as with earlier versions of Opus but to support the synchronicity of ARexx commands, a new 'Quiet' variable has been added.

  FTPSetVar Quiet <state>

  disables or enables all requesters for the source Lister. To disable requesters, state may be '1' or 'on'. To enable them, state may be '0' or 'off'. This variable is designed for use with automatic or non-interactive scripts which run without user intervention. The command returns 1 if it succeeded, 0 if it failed. Example:- FTPSetVar Quiet 1

- **FTPOptionsDEFAULT/S**

  Displays the options editor for the current source FTP Lister. With the optional switch DEFAULT, this will display the global default settings used by all FTP Listers that don't have their own custom options set. (The command is available from the status popup menu in all FTP listers.)

- **FTPAdd**

  Adds the site in use from the current SRCE FTP Lister to the address book. Use this when you have connected to a new site with the connect requester and wish to remember it for later access. (The command is available from the status popup menu in all FTP listers.)

- **FTPQuit FORCE/S**

  The OpusFTP parent process runs in the background after being invoked and lies dormant when not being used by an active FTP Lister or script. If you wish to quit OpusFTP completely, FTPQuit will instruct all current and pending Lister operations to quit and the main process itself will terminate after they have completed. Although this will free some memory there is usually no need to quit the OpusFTP process

  The **FORCE/S** option forces all Lister operations to be aborted and OpusFTP will quit immediately.

Opus Commands Patched by OpusFTP

The following Opus commands are 'trapped' and directly supported by OpusFTP:-

- **AnsiRead, HexRead, Play, Print, Read, Run, Show and SmartRead** work as in Opus but each file will be acted on as it is downloaded.

- **Delete, Move, MoveAs, Parent, Rename, Root** work as in Opus.

- **Configure** displays the Opus Lister format editor for the FTP Lister showing the current or custom format for the lister.

- **Copy & CopyAs** work as in Opus with the exception of the MOVEWHENSAME switch which has no meaning for FTP.

- **FindFile** works as in Opus. Useful for finding files on FTP sites but can be slow. Many FTP sites have some type of index file which may prove more useful.

- **GetSizes** works as in Opus except for the FORCE flag. Directories will always be scanned in OpusFTP.

- **MakeDir** works as in Opus except for the NOICON switch which is fairly meaningless for FTP. Directories are always created without icons.

- **Protect** works similar to Opus but with extra GUI

- **ScanDir** PATH,NEW/,MODE/K,SHOWALL/S, CONTAINER/S,FTP/S

  Normally, when the ScanDir command affects an FTP Lister the path is treated as a local path and the FTP connection will be terminated. We have maintained this for compatibility with previous versions of OpusFTP. However for specific use with FTP, you may now override this behaviour with the new special FTP switch. ScanDir FTP 'xyz' will cause the Lister to change to directory 'xyz' on the FTP site. ScanDir 'xyz' without the switch will read the local directory 'xyz' into the Lister and disconnect from the FTP site as it used to.

Miscellaneous

Cut and Paste

If you hold the shift key down when pasting a string into a string gadget using **RAmiga+V**, the string will be inserted at the current cursor position instead of replacing the existing string completely.

☞ ***Be careful to distinguish between RAmiga-v and RAmiga-V (with shift held down), because shift-V is the new behaviour.***

Disks

Opus now recognises PSx:(profilesystem on diskspare.device) as another device that can use the floppy drive, when it does its good disk/bad disk magic.

LoadDB (LoadWB replacement)

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).

General Notes and bug fixes

We have fixed all known and reported bugs and the following is a list of some of these problems from earlier version which have been identified and resolved for the Magellan-II version.

- We have made some changes to the way filetypes and versions are displayed in the Lister. Instead of being updated individually as each file is scanned, the changes are cached and updated in batches every few seconds. Also, the Lister does not try to update the display if layers are locked, say,due to a drag and drop problem. These changes will minimise the

rare but real potential for drag and drop lockups.

- Fixed a problem if the length of the volume name of a disk plus the length of the directory on that disk that you were currently in was longer than about 45-50 characters (would cause hits and/or a crash).

- If a Lister showing the root of a volume is iconified, the label of the icon will now leave the colon after the volume name.

- Fixed a potential problem with the use of ARexx scripts in the CopyTo directory (see TestScript.dopus5 in the System/_CopyTo directory on the installation disk/cd). If the comment you had given the ARexx script was shorter than the filename (or you had not given it a comment), it could cause a crash.

- We tweaked the icon positioning routines a bit; viewing a directory in 'show all' mode (or with lots of un-snapshotted icons) is now faster and it also does a better job of positioning the icons.

- If you used an Opus internal command that needs a destination and supplied the destination as a parameter to the command, Opus would erroneously display the 'Select Destination' requester if the command was combined with any other instruction.

- If you used the LoadButtons command with the START parameter and specified the name of a start menu that didn't exist, it could cause enforcer hits and/or crashing.

- Icons used as arguments in a shift-doubleclick operation are now supplied in the order they were selected, instead of reverse order (to be compatible with Workbench).

- If the Comment command is called from a filetype menu, it now correctly displays the original file comment in the requester.

- The DeviceList now shows free and used space to one decimal place.

- Copy to Clipboard now works correctly in the text viewer when there is a tab on the first or last line being copied.

- The left and right edges of name mode Listers are now filled with the appropriate background colour instead of being left clear (this removes the ugly 'borders' in name mode Listers when the files have a background colour other than colour 0)

- Iconified Listers and some other types of icons were offset vertically by one pixel; if you repeatedly iconified and de-iconified a Lister you would see the icon move slowly down the screen.

- Custom script entries that have been removed from the DOpus5:system/scripts directory are now really deleted when you click delete in the scripts editor.

- Fixed a problem with the cache list/device list and custom handlers - if you clicked the cachelist button in a Lister with a custom handler, and then clicked it again, the custom handler would get "lost"

- Eliminated annoying flash of icons after reading icon mode Lister.

- If you double-click on an icon, all other icons in the window that were selected are now automatically de-selected, unless the shift key is held down.

- Fixed a problem that could cause programs doing a FindTask("Workbench") to fail when Opus was running.

- If the mouse is over the screen title bar, and you press the right mouse button, the button is never trapped by any objects (Listers, start menus, etc) underneath it - this is so you can always access the pull-down menus. However, this prevented you from accessing the edit menu in start menus. This has now been changed so that if you hold down control while pushing the right button, the event is trapped by start menus even if the mouse is in the screen title bar.
- Dragging icons from the button bank clipboard was broken - icon would be positioned off the mouse pointer.

- Icons without a 'selected' image, with borders turned off, would leave behind the bottom line of the image when they were moved or removed.

- Fixed problem which could cause a Lister opened by double-clicking on an icon, etc, to open empty.

- Fixed occasional print.module crash when printing to a file.

- Made changes to icon sizes to make them compatible with Workbench.

- Groups can now contain items that don't have their own icons so you can now add left-out commands to groups.

- 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)

- Opus did not recognise that the DONOTWAIT tooltype as 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.

- In some languages (eg Swedish) name substitution in the date

field in Listers resulted in incorrect padding between the date and time.

- 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.

- 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.

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 on the program disk, the DOPus Plus CD or from our web site.

Index