



AMIGA[®]
OS 4.0

Feature List

Revision 1.0

January 7, 2003

© 2003 Hyperion Entertainment

Introduction

The following pages provide a more detailed description of the “new” functionality introduced by Amiga OS 4.0.

Functionality is considered “new” when it was not present in Amiga OS 3.9 (Boing Bag 2).

OS modules which are simply recompiled for PPC (such as Workbench for instance) are not mentioned here.

Note that this document should be considered preliminary and the information it contains is still subject to change and refinement.

Nothing in this document shall be construed as limiting Hyperion’s right to modify the functionality of AmigaOS 4.0 without prior notice.

Further revisions of this document will be made available.

This document is © (2003) Hyperion Entertainment.

Permission is hereby granted to reproduce the content of this document in whole or in part solely in unmodified form.

Exec SG (second generation)

Exec SG is the OS 4.x kernel and as such the heart of the system.

Inherited functionality from older Exec versions:

ExecSG is fully backward compatible to older Exec versions and preserves all of its features including, but not limited to:

- Fully pre-emptive multitasking (round-robin)
- Support for shared libraries
- Very fast inter-process communication (IPC - message passing by reference)
- Fast interrupt handling (low latency)
- Low-overhead task switching

New functionality introduced by ExecSG:

- Fully PPC native: not a single 68k instruction left in ExecSG.
- 68k emulation: both interpreting and JIT emulation available (see below), JIT for speed, interpreting for compatibility.
- Extremely fast context switching times: 1 microsecond on a 600 MHz G3.
- Extremely low latency for IPC: Signal "roundtrip" times below 4 microseconds.

Clarification: Roundtrip time means the time required for task A to send a signal to task B and task B to send a signal back to task A. Roundtrip time is the time needed for task A to send the signal and receive the answering signal.

- Interrupt handler can be in PPC code but can also be emulated, which is useful for old hardware drivers.
- New API for DMA drivers: DMA capable drivers can now query the "scattering" of memory with just one call instead of repeated CachePreDMA/CachePostDMA calls. This makes DMA transfers much more efficient for controllers that support "scatter/gather" interfaces like most busmastering IDE controllers. The new API is also much easier to use.
- Hardware Abstraction Layer (HAL): ExecSG itself is machine independent, it sits on an abstract "hardware". Only the abstraction layer itself has to be ported to a new platform thus making the transition to other hardware relatively easy. The abstraction layer itself is also divided up into different parts: CPU, machine and common. For a transition to another machine with an already supported CPU, only the "machine" part has to be ported.
- Resource tracking: Applications can create "trackable" resources, for example memory, message ports, messages, semaphores, and have them tracked by ExecSG. When the program exits (either normally or abnormally due to a crash) the resources are freed automatically.
- Improved trap handling: Previous incarnations of Exec only allowed one trap handler per task. ExecSG allows trap handlers to be installed for different "events", like illegal instructions, address errors, privilege violations, etc..

- Unified MMU interface: ExecSG now offers a special API to access the Memory management unit. No custom solutions needed anymore.
- Integration of utility.library: ExecSG integrates the functionality of utility.library to present the application programmer with a unified interface throughout the system. This allows Exec functions to use hooks and tag lists instead of simple callbacks or variable argument lists. This functionality is included in its own interface (see below, "new library model", for a discussion about interfaces). All of these functions can of course still be accessed through utility.library.
- Integration of amiga.lib: ExecSG integrates the "Exec support functions" of amiga.lib. This also includes CreateTask, which now has an additional parameter that can take up a taglist, which in turn can be used to pass parameters to a task (these parameters are passed like function parameters).
- Limited memory protection: Critical memory areas can be protected: Kernel memory areas, all code areas, unused memory (also unused free memory). Applications can use the MMU interface to protect their memory from others.
- AltiVec.resource: Used to arbitrate access to AltiVec streams.
- WarpUp emulation.

Planned and already partially implemented functionality:

- Symmetric Multi-Processing (SMP): support for multiple CPU's.
- Multithreading: Tasks can consist of multiple threads of execution which are scheduled individually on a higher frequency.
- Isolated address spaces: This feature will give each task its own address space so that it appears to be the only running task. Communication will still work without copy through shared memory areas. This feature is already available in the current ExecSG but not enabled yet due to incompatibilities with OS3.x. It is also only really useful with multithreading.
- Pluggable schedulers

An in-depth look at some of the more advanced ExecSG features:

New library model

ExecSG sports a new model for shared libraries. The old model will remain to support 68k libraries. PPC native libraries will be organized slightly different: The library base is still there (and looking the same as before) but the jump table only contains 68k functionality (for compatibility).

PPC native functionality is provided by **Interfaces**:

- An Interface can be thought of as a separate jumtable, containing pointers to the actual library functions.
- An Interface can have its own data area in addition to that of the library base.
- An Interface can be shared among tasks (like a library) but may also be created as a private instance, allowing instance data to be stored in it (similar to the old SAS/C method of duplicating

the library base). A task can also create multiple instances of the same interface with each interface having its own data (essentially like real objects).

- Reference counting is used to work around stale pointer problems.
- A library can export any number of interfaces. For example, ExecSG exports the interfaces "main", "utility" and "mmu", which represent the main API, the integrated utility.library and the special MMU interface.
- A library can export the same interface in multiple versions thus allowing for full backward compatibility.
- Interfaces can be modified with SetMethod (like SetFunction), allowing "morphing" interfaces that can efficiently adapt to internal state changes. For example, expansion.library uses this in its PCI interface to adapt a PCI interface to the actual hardware.
- Interfaces can also be created in such a way that they are protected against patching. This is essential for services like SSL (or any security-sensitive service).
- Interfaces can be used to write Object-oriented libraries.
- Interfaces are conceptually similar to component programming models.

New memory system:

- ExecSG uses a fully virtualised address space. As a result addressable memory can be larger than the available physical memory.
- Memory allocation can specifically request physically mapped memory or virtualised memory.

Clarification: virtualised memory is NOT virtual memory in the sense of disk-based memory.

- Memory allocation can be deferred, i.e. virtual addresses are allocated but no physical memory is assigned until it is needed (which is then handled automatically and fully transparently to the application). If an address range is not referenced, it will never be backed up with physical memory so no memory is wasted.
- Memory can be paged-out to disk (virtual memory). This feature can be disabled if required (under certain circumstances, this can be switched on or off on-the-fly, no reboot required).
- Memory can be reallocated: This essentially resizes a block of memory without moving it.
- Memory reallocation in conjunction with deferred mapping can be used to load disk-based data structures in an extremely memory-efficient manner: Allocate a large block of memory deferred, fill it up with data, and reallocate it to the appropriate size. This method will never use more than the memory that is actually being accessed and can be used to load e.g. compressed data when the uncompressed size is not known but a rough estimate is nonetheless possible.
- The memory system is now "object-oriented": The virtual address space is divided into areas (similar to the old MemHeader system) but each memory area can be attached to an allocator and a fault handler.
 - An **allocator** organizes the address space it has been assigned to;
 - it handles allocation/deallocation, DMA preparation etc.

- it handles allocation of physical memory and binding physical memory to virtual addresses
- **A fault handler** is responsible for handling memory faults in its assigned area:
 - it handles deferred memory requests by providing physical memory as soon as virtual areas are accessed.
 - it handles interaction with the pager to swap memory out to/in from disk.
 - it handles illegal access (privilege violation, illegal addresses etc).
- Possibilities for allocator/fault handlers:
 - "Normal" virtual addressed memory: The default case.
 - Stack areas: A stack might underflow (out of stack space), in which case the fault handler can use the allocator to provide a new chunk of physical memory, thus automatically enlarging the stack. Stack underflows are one of the major problems on the Amiga today and account for a significant percentage of crashes.
 - Memory mapped files: The fault handler swaps in portions of a file when they are accessed in virtual address space. This works exactly like virtual memory but allows for normal disk-based files to be accessed as if they were in memory.
 - Address space emulation: The system can emulate alien address spaces, for example emulate an AGA chipset on a machine that doesn't have AGA.

Just in Time (JIT) 68K emulation

Dynamic recompilation or JIT compilation is a technique whereby (in this case) 68K machine code is translated on the fly to PPC machine code which is stored in a temporary buffer to avoid recompilation each time the recompiled code is executed.

All 68040 instructions including FPU instructions are fully emulated.

Even on a lowly 603e@160 Mhz the recompiled code reaches and in some cases surpasses the speed of a 68060@50 Mhz.

- Directly integrated with ExecSG. No "boxed" emulation approach.
- JIT can be switched on/off on a per-task basis
- Different JIT and/or interpreter can be used on a per-task basis ("pluggable" emulation).

AmigaOS 4 uses a task-based emulation which is directly integrated into ExecSG to emulate legacy 68k code. Each emulated task is scheduled independently and has its own emulation process and interpretation/translation stage independently from other emulated tasks. Hence there is no need for virtualization of resources, no delay to build a sandbox mechanism.

While this is in theory less compatible than a sandbox or "virtual machine" type approach, it offers a few key advantages:

- More responsive; such an approach does not break time-critical/real-time behavior.

Clarification: With a sandbox approach, a separate scheduler runs inside the sandbox to schedule 68k tasks, which increases delay times substantially. A task-based scheme does not separate 68k and PPC tasks. They are treated equally.

- External interrupts (generated by the hardware or the kernel) as well as some exceptions do not have to be emulated, which improves emulation speed.
- Legacy applications may benefit (to some extent) from additional functionality introduced by ExecSG (resource tracking and memory protection, for instance), albeit in a limited way.

Generic PCI support available through expansion.library

Currently supports Prometheus and Mediator PCI busboards as well as the Articia S on the AmigaOne hardware.

Completely re-implemented DOS

- There are no longer any limits to DOS path length strings for the following functions: Open(), Lock(), SetProtection(), SetOwner(), SetFileDate(), DeleteFile(), CreateDir(), SetComment().
- Completely new implementation of all the pattern matching functions, providing far better pattern support than the old routines.
- Includes a suite of Address Tracking Functions to complement DOS to provide a consistent interface for various development tools.
- All of active (TripOS) BCPL DOS components from V40, are now defunct resulting in a more stable and consistent C-based DOS library.
- Many bugs fixed including memory leaks and "NIL:" filehandle issues as well as a more robust RunCommand() function.

New Shell and C: commands

- Improved shell with pipes, history, output recording, command line editing, file name completion and new built-in commands. Detects software implementation problems such as unreleased signal bits to help developers to clean up their code.
- New Queue-Handler, implementing the PIPE: device; faster, more memory efficient. No longer suffers from the implementation problems of the original Queue-Handler.
- New and improved Aux-Handler, allowing you to run a command line shell on the serial terminal instead of a console window.
- New MD5Sum command for calculating and verifying checksums of files; compatible with the Unix command of the same name.
- All shell commands now transparently support soft links.

OS 4.x native SCSI drivers for the Symbios SCRIPTS family of SCSI controllers

Support for the onboard SCSI controllers of the Cyberstorm PPC and BlizzardPPC.

Highly optimised layers.library

- Many source code improvements and bugfixes leading to a cleaner implementation.
- Improved memory management (less fragmentation and better low-memory handling).
- Ability to hide layers (ie. make them invisible).

- Introduction of several speed-ups when moving and resizing of layers.
- Offscreen-layers, allows Intuition windows to be dragged off screen.

Picasso 96 V3 RTG system

P96 is a Retargetable Graphics system which redirects graphic output to the monitor and bypasses the Amiga's built in graphics modes (OCS/ECS/AGA).

More details (especially on the supported graphics cards) to follow soon.

New Intuition and Reaction

- The Intuition API has been noticeably extended with 50+ new functions, several new properties for screens, windows, gadgets and images and the integration of features previously only available through third-party extensions or patches.
- Possibility to drag windows outside of their screen's boundaries. This feature is optional and can also be disabled on-the-fly by way of a qualifier key.
- Intuition offers functions to hide/show windows, useful for instance to easily implement iconification in any program.
- Some amiga.lib BOOPSI support functions have been integrated into intuition.library for greater efficiency.
- Intuition offers a screen notification system, allowing client applications to be informed in real time whenever a specific public screen is opened, closed, depth-arranged, locked or unlocked. It is also possible to receive a notification for any window that is opened or closed on that screen.
- Screens can be locked in an Intuition-friendly way, whenever it is needed to directly access their bitmap, without having to go through the locking functions of layers.library. This can be done both by applications and by BOOPSI classes.
- The updated API offers functions to set and get attributes of already open screens and windows, where applicable.
- A new type of window is available, "toolbox window", which cannot be activated and is therefore very useful to implement floating toolboxes (as commonly used in e.g. drawing programs) which don't take away the focus from the project window when the user clicks on their buttons.
- Intuition can optionally remember the last active window and gadget on every screen and reactivate them as soon as that screen is brought to front again.
- All memory allocations within Intuition make use of memory pools, which significantly reduces system memory fragmentation.
- Intuition windows can be dragged and resized in real time (opaquely) without blocking any output occurring within other windows.
- The DrawInfo structure does contain a dri_Screen back-pointer which makes it much easier for BOOPSI image classes to obtain information on the layout of the screen they will use for display.
- Some API functions are available which allow to share pictures between different screens and/or applications, minimizing the need for reloading and remapping. Usage of these functions greatly reduces memory consumption and increases performance.
- An official method has been introduced to add custom gadgets to a window's titlebar in a legal and effortless way. This increases visual consistency and ensures future compatibility.

- A new ReAction class has been added which implements pop-up menus.
- A number of long-standing bugs and quirks of Intuition have been finally fixed and code optimizations have been applied where possible.
- The look and feel of every element of the Intuition GUI can be configured by way of a preferences editor, and all of the user interface settings can at any time be programmatically queried with appropriate API functions.
- Greatly increased the number of DrawInfo pens, for a much more refined control over the colors of all GUI elements.
- Full-intensity and half-intensity bright/dark pens (used in pseudo-3D effects) can be defined in terms of contrast relative to a particular base pen and therefore automatically computed by Intuition.
- The style of almost any frame used in the GUI can be set as 3D, XEN-like or flat (thick or thin). Each of these styles, in turn, can be fine-tuned by configuring the appropriate DrawInfo pens.
- Virtually every element of the GUI can be textured with a bitmap. The user can select different bitmaps for buttons, scrollers, menus, window backgrounds, requester backgrounds, screen titlebars, window titlebars and window borders. For most of these elements, a separate texture can also be associated to each visual state (selected, inactive, disabled...).
- On hi/truecolor screens, most of the 3D elements having a texture as background can have their bright and dark edges drawn as brighter/darker shades of the texture. This spares the user from the hassle of finding suitable edges colors every time a different texture is chosen and also conveys a more realistic 3D effect.
- The look of scrollers can be configured in every detail, such as the type of background (solid, textured or furrow-like), the thickness of the container` sborder, the knob color, the style of the arrows and much more. Separate configurations are possible for scrollers in window borders and scrollers inside windows.
- Intuition menus can optionally be brought up under the mouse pointer (pop-up) rather than on top of the screen (pull-down).
- Intuition menu panels can have an arbitrary shape, be drawn with partial transparency and cast a drop shadow. It is also possible to choose the style of their frame, which can be fully 3D, XEN-like or simply flat.
- Intuition menus use different symbols for items that can be toggled (checkmark) and mutually exclusive items (radio button). They also display a real arrow symbol to mark sub-menus, rather than using the ">" character (ASCII 187) for this purpose as done previously.
- Disabled gadget labels and menu items are drawn with a pseudo-3D embossed look on any screens with 16 or more colors, rather than being covered with an old-style ghosting pattern.
- GadTools and ReAction interfaces share the exact same look (as configured by the user) in almost every detail, for a consistent GUI appearance.
- Window titles can be placed at the left side, centered in the dragbar or centered in the whole titlebar.
- The height of screen and window titlebars can be increased without changing the screen font.
- The user can set the exact width of screen and window titlebar gadgets or just make them always square, wide (3:2) or narrow (2:3).

- The user can freely choose the width and height of window sizing gadgets and border arrow gadgets, as well as the thickness of all four window borders.
- For ReAction windows, it is possible to choose the preferred window border (right, bottom or both) where the sizing gadget is to be placed.
- The style and geometry of system gadgets in window borders and screen titlebars can be selected among a number of pre-defined choices, which can be extended by way of external style plugins.
- It is possible to have individual GUI configurations for specific screens or to allow all screens to use common GUI settings.
- External pictures can be used, in the form of "bitmap sets", to customize the look of screen/window system gadgets, menu glyphs, and most symbols used in GadTools and ReAction gadgets (arrows, checkmarks, etc.); a small logo can also be displayed at the left side of a screen's titlebar.
- Intuition's `sysclass` (system image class) supports alpha channeling on hi/truecolor screens, thus enabling its glyphs to perform effects like anti-aliasing or realistic drop shadows on any background. This is possible for both vector glyphs and external bitmaps.
- Intuition's `sysclass` offers an enhanced public API which developers can use to easily implement custom vector images taking advantage of all the special rendering features of the class.
- Intuition's `sysclass` features several new image types including those commonly used for titlebar gadgets, such as iconify, pop-up, padlock... It also offers a generic frame suitable to implement titlebar gadgets with a custom graphic or textual content.
- Introduces Meta-Dragging and Meta-Sizing for windows: Appropriate qualifier keys allow the user to drag windows by clicking inside their area and resize them by any border or corner.

Port of Mesa 5.0 (OpenGL 1.4).

Mesa is a high-level 3D API with a command set which is nearly identical to that of OpenGL¹, the cross-platform 3D API originally developed by Silicon Graphics (SGI – <http://www.sgi.com>).

(see: <http://mesa3d.sourceforge.net/>)

Warp3D V4 and Warp3D NOVA

The first installment of OS 4 will include the current V4 of the generic Warp3D low-level driver system. For the time being, both the Permedia 2 and Voodoo 3 graphics cards will be supported, with Radeon support to follow.

Warp3D NOVA, the follow-up to Warp3D V4, will be a complete rewrite and all-new design. Planned features include:

- Shader-Centric design. All fixed-function pipeline functionality will be treated as a special case.
- High-level shading language support similar to Cg™.
- Modular, component-based design
- Support for modern graphics card features like
 - Vertex shaders
 - Pixel shaders
 - AGP texturing

¹ OpenGL is a trademark of SGI

- On-board vertex and data buffers
- Environmental and dot-three bump mapping
- Displacement mapping (where available)

OS 4.x native version of AHI v6

AHI is the standard Retargetable Audio subsystem for AmigaOS which supports both Zorro and PCI based audio hardware.

Currently supported audio hardware:

Aura*	Clarity*	Concierto
Delfina	DraCo Motion	ESS Solo 1
ForteMedia FM801	Maestro Pro	Melody
Paula (standard Amiga)	Prelude	Repulse
SoundBlaster 128	SoundBlaster Live!	Sunrize
Terratec 128i PCI	Terratec 512i digital	Toccatà
VIA AC97	Vivanco	Wavetools

More information can be found here: [http://www.lysator.liu.se/\(all\)/~lcs/ahi.html](http://www.lysator.liu.se/(all)/~lcs/ahi.html)

FT2Engine: new powerful font-engine

FT2Engine is an Amiga font engine that allows you to use "alien" font formats with each program that either uses diskfont.library or the bullet.library API.

Features:

- Supports PostScript Type1 and Type42, TrueType/OpenType (including TrueType collections, embedded bitmaps, and OpenType 1.3 with Unicode characters beyond 0xffff), PFR (Truedoc dynamic webfonts), BDF&PCF (X11 bitmap fonts) and Windows .fnt/.fon fonts.
- Global font, glyph and charmap cache with adjustable size.
- Support for embedded fonts (raw font from memory without .otag file).
- Includes font manager FT2Manager with localized MUI GUI that allows you to install, modify, rename and delete fonts, create bitmap fonts in different charsets, select encoding, adjust metrics, select between three types of hinting, view all Unicode3.2 chars, view any text (encoded in UTF-7|8|16|32 or plain 8bit) with the selected font, kerning and charset. Has bubble help, selectable font smoothing etc.
- diskfont, asl and locale library now support:
 - Font names with more than 30 characters
 - Fonts and catalogs with different charsets
 - Selection of the default system charset (e.g. ISO-8859-15 with Euro)
- Improved "bullet" API (font smoothing, bitmap font support etc)

“Roadshow”: TCP/IP stack & PPP drivers

- Compatible with software written for AmiTCP V4, Miami, Miami Deluxe and Termite TCP.

* Sampling only

- Written from scratch, incorporating the 4.4BSD Lite2 TCP/IP kernel.
- Compact and efficient design; fastest Amiga TCP/IP stack to date.
- Automatic configuration via the Dynamic Host Configuration Protocol (DHCP) supported through built-in DHCP client; activate a networking interface and your Amiga automatically becomes part of your local network. This enables you to use cable and DSL modems with your Amiga.
- Uses a small, but comprehensive set of Amiga specific administration tools and configuration files; easier to learn and use than plain Unix style commands and files.
- Just one command required to start up and make your Amiga part of your network; no need to configure routes, IP addresses and domain name system servers separately.
- Supports Ethernet (broadcast) and PPP, SLIP, CSLIP (point-to-point) networking.
- Completely localized, with error messages and reports printed in your native language.
- Built-in Internet Superserver ("inetd") to start server application software in response to client requests (e.g. Apache web server, Samba file server, etc.).
- Built-in "TCP:" device handler which makes it possible to use the TCP/IP stack functionality from within shell scripts or Arexx programs.
- Integrated IP packet filter (IPF) and network address translation (NAT); build your own firewall or use your Amiga as a network gateway.
- Uses the SANA-II standard for networking drivers; supports the release 2, 3 and 4 extensions such as for direct memory access and dial-up networking.
- Multicasting supported.
- Built-in support for the Berkeley Packet Filter (BPF); monitor the local networking using your Amiga.
- Greatly enhanced application programming interface which gives developers control over the networking interfaces, routing and kernel behaviour, allows for data to be filtered, monitored and reshaped; create your own firewall, check which requests your Amiga receives from the network, deny incoming or outgoing connections. The enhanced application programming interface is still backwards compatible with AmiTCP V4.
- Comes with special networking drivers for dial-up networking (asynchronous PPP; for modems and ISDN adaptors which behave like modems) and DSL access (PPP over Ethernet).
- Driver for PPP over Ethernet supports the SANA-II standard for networking drivers including the release 2, 3 and 4 extensions such for as direct memory access. A special "low overhead" mode increases networking performance by cutting the path short which lies between the Ethernet driver and the TCP/IP stack.
- Both PPP drivers (asynchronous PPP and PPP over Ethernet) support automatic configuration of local and network addresses, routes and domain name system server addresses. Link quality monitoring and testing, authentication with plain text and encrypted passwords (CHAP and MS-CHAP) are supported.
- Driver for PPP over Ethernet includes security features to discourage password sniffing on the network.

AmiSSL V2

- Port of latest version of OpenSSL.

- Implementation of SSLv2, SSLv3, TLSv1 and the required code to support both SSLv2, SSLv3 and TLSv1 in one server and client.
- Ciphers: Blowfish, CAST, DES, RC2, RC4, RC5 etc. (all with several modes of encryption).
- Message digests: MD2, MD4, MD5, MDC2, RIPEMD-160, SHA (SHA-0), SHA-1
- Public key encryption/decryption/generation: RSA, DSA, Diffie-Hellman key-exchange/key generation (no limit on the number of bits)
- X.509v3 certificates: encoding/decoding into/from binary ASN1 and a PEM based ASCII-binary encoding which supports encryption with a private key.
- Important design optimization which makes it faster than AmiSSL v1.
- Used as main SSL library in IBrowse. Programs that use AmiSSL v1 (YAM, AWeb, SimpleMail etc) will need only minimal modifications to work with AmiSSL v2.

Re-implementation of the Amiga file system (FFS2)

- Backwards compatible with the original FastFileSystem, including all options such as the Directory Caching File System (DCFS) mode.
- Hard links, file record locking and file notification supported.
- More robust support for soft links than was available in the original Fast File System implementation.
- Reentrant; code can be shared by several storage devices, reducing memory requirements.
- Enhanced to support file and directory names with up to 107 characters.
- Fast and memory efficient built-in data caching system.
- Multithreaded design: multitasking friendly because directory scanning, reading/writing file data and directory manipulation can run in parallel.
- Built-in file system validator.
- ExAll() style directory scanning supported.
- Owner and group IDs can be attached to file and directories; network enabled file systems (e.g. Envoy) depend upon this feature.
- Storage media larger than 4GBytes in size supported.
- Variable storage block sizes supported; large data block sizes help to improve file system performance.
- More robust storage management which reduces the likelihood of data getting corrupted or having to validate a disk's contents.
- Removable media (floppy, ZIP, JAZ, MOD, etc.) supported.
- New plug-in interface for data caching and on-the-fly encryption.

CD/DVD Filesystem

- Supports ISO 9960 including RockRidge with Amiga extensions, Joliet and multisession CDs, HFS and HFS+, reading Audio CDs (the audio tracks are converted to AIFF files) and reading VideoCDs.
- Mount Rainier support:
With a Mount Rainier (CD-MRW) capable CD-RW drive you can use CD-RWs discs as if they

were large floppies. One can copy files to a CD-RW with the Workbench/Shell, delete them again etc.

(See: <http://www.mt-rainier.org>)

- Packet writing support for those CD-RW`s that don`t support Mount Rainier.

USB stack

OS4 will include a USB stack to support peripherals using the USB standard for connectivity.

As the stack is part of the OS it can be available during boot, allowing the use of USB mice and keyboards in the early startup menu.

The USB stack has multilanguage support allowing the user to use other languages with USB devices which supports this feature for device name, descriptions etc.

Some features of the USB stack can be controlled by the user - e.g. whether an attachment notification window should be opened when a new USB device is attached, and how long the window should stay open. Also the amount of logging the USB stack should perform can be controlled by the user.

Appropriate drivers exist for hubs, mice and keyboards. Drivers for scanning and printing are also available with the stack.

Drivers for USB Mass Storage devices (harddisk etc.) and Human Interface Devices (multimedia keyboards, scroll-mice etc.) are in the works along with a GUI for viewing attached USB devices and their status.

Currently supported hardware includes the Highway USB cards as well as the Thylacine Zorro 2 based card.

“Partition Wizard”: Recovery and Salvage tools for FFS2 and SFS

- Check: Checks FFS and SFS partitions for errors.
- Repair: Fix-in-Place repair of errors on a FFS partition.
- Salvage: Recover-by-Copy, copies files still intact from a damaged FFS or SFS partition to another one.
- Undelete: Undeletes deleted files on FFS partitions.
- Unformat: Unformats a ' quick' formatted FFS partition.
- Optimize: Optimizes a partition like ReOrg, DiskOptimizer (FFS) or starts the filesystem defragmenter (SFS).
- Find Partitions: Scans a HD for lost FFS and SFS partitions, the result can be imported into Media ToolBox.
- Convert: Converts Intl. OFS/FFS and DirCache OFS/FFS partitions to the new FFS2 LongName format.

“Media Toolbox”: HDToolbox replacement

Media Toolbox is an advanced and feature-rich partitioning and preparation tool for a wide variety of media, with a comprehensive SCSI toolkit thrown in for good measure.

Partitioning is made easy by the use of a simple and user-friendly GUI.

Complex operations like setting up partitions for OS installations (OS4, Linux, mixed OS4-Linux) are made easy by the use of "layout templates". Salvage and recovery of disk layouts are handled in the same way.

One can choose between two modes of operations, "Normal" and "Expert" (power users). In "Expert" mode the user can edit almost any value of the RDB.

The inbuilt SCSI toolkit sports features such as a rich lowlevel-format (including progress indication and Iomega support), fast disk/partition erase, CD-R brand identification and SCSI mode pages visual editing.

InstallerNG (replacement of the Commodore Installer)

- Enhanced functions: DATABASE, EXISTS
- New functions: BEEP, COMPARE, DELAY, FINDBOARD, LET, NOP, RANDOM, REBOOT, SETENV, SIMULATE-ERROR, SWING, GET-PROPERTY, PUT-PROPERTY, REMOVE-PROPERTY, SAVE-PROPERTY-OBJECT, READ-PROPERTY-OBJECT, CAST-INT, CAST-STRING
- Nice GUI: The built-in GUI is based on a BOOPSI class-collection; these classes allow easy font-adaption and resizing. Additionally, you may "plug-in" other GUI-systems (like MUI, BGUI, ...) via a shared library called "installergui.library".
- "Help" window can stay open, while you install your packages; this is a built-in feature and should be provided by every GUI.
- Comfortable Workbench Start: if you start InstallerNG from Workbench and provide it with no script via tooltypes a requester pops up asking you whether you want to load a script by a file-requester or if you want to app-iconify the installer. If you drop a script-file on the application icon the InstallerNG is started.
- Flexible interpretation: if an error raises while the interpretation process, InstallerNG allows you to continue at the very next function.
- New Tooltypes/CLI-Arguments: LAZYCOMPILE, CREATEUNINSTALL, ALWAYSCONFIRM, COPYFILECOMMENT, NOSYSDELETE, DEBUGMODE
- SOOP - Simple Object Oriented Programing: with help of the new functions PUT-PROPERTY, GET-PROPERTY and REMOVE-PROPERTY InstallerNG implements LISP-like property-lists for symbols.
- Full installation control: If you want to, the InstallerNG asks for confirmation of every action, no matter what the script-programmer specified in his installer script.

AmiDock

The OS4.0 version of AmiDock is a "from the ground up redesigned" version of OS3.9™'s AmiDock.

The new AmiDock features a much improved icon layout engine together with text- and button mode docks.

AmiDock is now far more interactive than the old version. A lot of things like the arrangement, addition or removal of icons, categories and docks can now be achieved without opening the preferences dialog.

The visual appearance of AmiDock now supports borderless, dragbar-less and even transparent docks to integrate easily into every Workbench setup.

The font- and color-setup is now also fully customizable and each dock can be minimized to a tiny bar to save space on the Workbench screen. Every icon or dock can now have its own hotkey for easy and quick access of often used items.

The most impressive new feature is the possibility to have active content within a dock in the form of "plug-ins". An AmiDock plug-in is called a "docky". Each docky has full access to all of AmiDock's internal functions and can cater for a virtually endless number of features.

The usage of dockies is very simple - adding a docky (for example by using drag&drop) will add and activate the docky and removing it will disable it. A virtually unlimited number of dockies can run simultaneously in every dock you want.

A number of dockies are included with OS4.0; the most-requested feature - subdocks - was made possible using a docky. There is also a docky which minimizes a dock automatically if the mouse pointer is not any longer over the dock. Other included dockies allow you to have a clock, a lens, animations and other active content within a dock.

The most thrilling applications will surely come from third party programmers in the form of third-party dockies (the API will be released with OS4.0) - just imagine having the current network traffic, the printer status, volume settings, the current rendering process of your raytracer and other such things in the form of a small and responsive picture within your dock.

application.library

This new library handles a lot of things which were up to now sorely lacking in AmigaOS.

Third-party applications can now "introduce themselves" to the system as "application" and can benefit by doing so in several ways.

By way of example: AmiDock will add the applications to the main dock to provide a taskbar-like access to the application. Each application can of course specify if it wants to run invisible in the background, with a static icon or even by displaying a docky to provide some advanced control features.

Applications which make use of this new library provide standard interfaces for opening and printing documents, can be quit, iconified, called to front, put to back etc. using a standard interface (i.e. AmiDock shows some popup-menus for applications and calls the application to front when it was clicked in the dock).

There is now also a standard way to access a list of last-used documents and applications to get a faster access to your most recently used items.

Whats more, application.library finally adds much requested possibility for applications to prevent screensavers from starting (for example this is very annoying when watching a movie or working with a video application).

Even OS4-games can take advantage of this new library as they can tell other applications to "shut up" as they need a non-disturbed game-mode where no requester should pop-up, no background sounds should be played etc.

The application.library finally also makes accessing program preferences as easy as pie by providing an advanced XML-based preferences object system which allows storing whatever information is required in a very versatile and flexible way. This puts an end to the different proprietary preferences-system used by most applications which are unreadable for users and require a lot of work on the part of developers.

Reading and printing of PDF files

AmiPDF (port of the latest version of xPDF):

- Supports PDF v1.4 files.
- Uses T1Lib and Freetype2 libraries to support great number of different font types (Type0, Type1/1C, Type3, Type42, TrueType, OpenType) with antialiasing.
- The xPDF language support packages can be used with AmiPDF. Currently available packages are: Chinese/simplified, Chinese/traditional, Cyrillic, Greek, Hebrew, Japanese, Korean, Latin2, Thai and Turkish.
- Fully supports password protected files (owner and user passwords).
- Includes full screen mode (useful for presentations).
- Multiple document windows.
- Integrated direct printing support.

AmiGS:

- Frontend for new port of Ghostscript.
- Displays Postscript (PS), Encapsulated PostScript (EPS) and PDF files.
- Full antialiasing support.
- Direct printing support.

AmigaInput API

AmigaInput unifies the game programmer`s need to support various input devices like joysticks, pads, steering wheels, keyboards and mice using one common API.

AmigaInput is primarily aimed at multimedia and game programmers who want to support a wide range of input devices, making the various API`s like `lowlevel.library`, `gameport.device` and `PSXport.device` a thing of the past. Programmers will not need to care about how a device is connected to the system.

Supported interfaces include:

- USB ports
- Classic joystick ports
- Catweasel MK3 joystick and keyboard ports (planned)
- Soundcard joystick ports

Supported device classes include

- Analog and digital joysticks
- Flight-Controller sticks
- Joypads and PSX-Pads
- Steering wheels

A SDK will also be available to allow hardware vendors to write device drivers for their hardware.

Furthermore a rewrite of `lowlevel.library` in OS 4 will directly map to AmigaInput, making it possible to use modern hardware in old, non-AmigaInput aware applications and games.

Archiving and de-archiving utility

DropArc and XAD: Extracts files from all types of archive formats known to XAD (e.g. lha, dms, lzx, arj, zip, cab, bzip2, tar, gzip, ...) by dropping files onto an icon. Creates lha and zip archives the same way. This also includes archive conversion, image extraction and any combination of the extract and compress methods. No GUI is required except the necessary part to change the available options.

Highly optimised datatypes

JPEG

PNG

TIFF

BMP

ILBM

GIF

(more to follow)

MooVid PPC

AmigaOS 4.x native version of the well-know movie-player supporting the following fileformats and audio and video codecs:

Supported file formats:

- AVI (Video for Windows) - with and without index
- QuickTime up to QuickTime6
- MPEG4 video (ISO standard)

Supported codecs:

Video for Windows (.avi)

<i>Supported Video Codecs</i>		
AccuPak	(PalomaAVI) (CLJR)	24 bit
DivX	(DVI3)	24 bit
DivX4/OpenDivX	(DIVX)	24 bit
DivX5.xx	(DX50)	24-bit
Intel Raw	(YUV9)	16 bit
Intel Raw	(YUV9)	24 bit
MPEG4/MPEG4 V1	(DIV1/MPG4/MP41)	24 bit
MPEGV V2	(MP42)	24-bit
MPEG4 V3	(MP43)	24 bit
Microsoft RGB	(RGB)	8 bit
Microsoft RGB	(RGB)	16 bit
Microsoft RGB	(RGB)	24 bit

Microsoft Video 1	(CRAM/MSVC)	8 bit
Microsoft Video 1	(CRAM/MSVC)	16 bit
Motion JPEG	(MJPEG)	24 bit
Radius Cinepak	(CVID)	16 bit
Radius Cinepak	(CVID)	24 bit
Radius Cinepak	(CVID)	32 bit
Run Length Encoded	(RLE)	8 bit

<i>Supported Audio Codecs</i>		
DVI ADPCM	MONO/STEREO	4 bit (16 bit)
PCM	MONO/STEREO	8/16 bit
MPEG1 Layer 2	MONO/STEREO	
MPEG1 Layer 3	MONO/STEREO	
MS-ADPCM	MONO/STEREO	4 bit (16 bit)

QuickTime (.mov, .qt)

<i>Supported Video Codecs</i>		
3iVX Delta 2	(3IV1)	24 bit (req codec from http://www.3ivx.com)
Apple Animation	(RLE)	8 bit (color/gray)
Apple Graphics	(SMC)	8 bit (color/gray)
Apple Video	(RPZA)	16 bit
Intel Indeo 3.1	(IV31)	24 bit
Intel Indeo 3.2	(IV32)	24 bit
Joint Pictures Expert Group	(JPEG/JFIF)	24 bit
Radius Cinepak	(CVID)	16 bit
Radius Cinepak	(CVID)	24 bit
Radius Cinepak	(CVID)	32 bit
Sorenson Video	(SVQ1)	24 bit
MPEG4 video	(QT6)(MP4V)	24 bit

<i>Supported Audio Codecs</i>		
RAW	MONO/STEREO	8/16 bit
TWOS	MONO/STEREO	8/16 bit
IDA-ADPCM	MONO/STEREO	4/(16) bit

<i>Supported Audio Codecs</i>		
MACE 6:1	MONO/STEREO	8 bit
MACE 3:1	MONO/STEREO	8 bit
MPEG Layer 3	MONO/STEREO	

OS 4.x native MUI

As many applications use MUI, the well-known GUI engine, the latest version of MUI will be included with OS 4.x.

The OS 4.x OEM version of MUI has a predefined look which closely matches the OS 4.x default look and which cannot be customised by the user without registering MUI separately.

Web-browser (Ibrowse 2.3 OEM)

IBrowse 2.3 is a very versatile MUI-based web browser. Features include HTML 4 support plus some Netscape and IE extensions, JavaScript and a customizable GUI layout.

Version 2.3 brings IBrowse to a new level - below is a very brief summary of changes since version 2.2:

- Heavily enhanced JavaScript support, based on JavaScript 1.5 and revision 3 of the ECMAScript specification, allowing access to sites which was not possible before.
- Secure connection support is now available via AmiSSL v2, which can be configured extensively in the IBrowse preferences. Importing and exporting of certificates is now also supported.
- Greatly improved general stability and memory leaks plugged
- Improved GUI, including enhanced browser tab functionality (such as drag' n' drop keyboard support).
- Issues with the ARexx interface have been fixed and new commands added
- Hundreds of other minor and major bug fixes and improvements and more workarounds added to improve compatibility with broken sites

OS 4.x native SFS (SmartFileSystem)

Included as contributed Software.