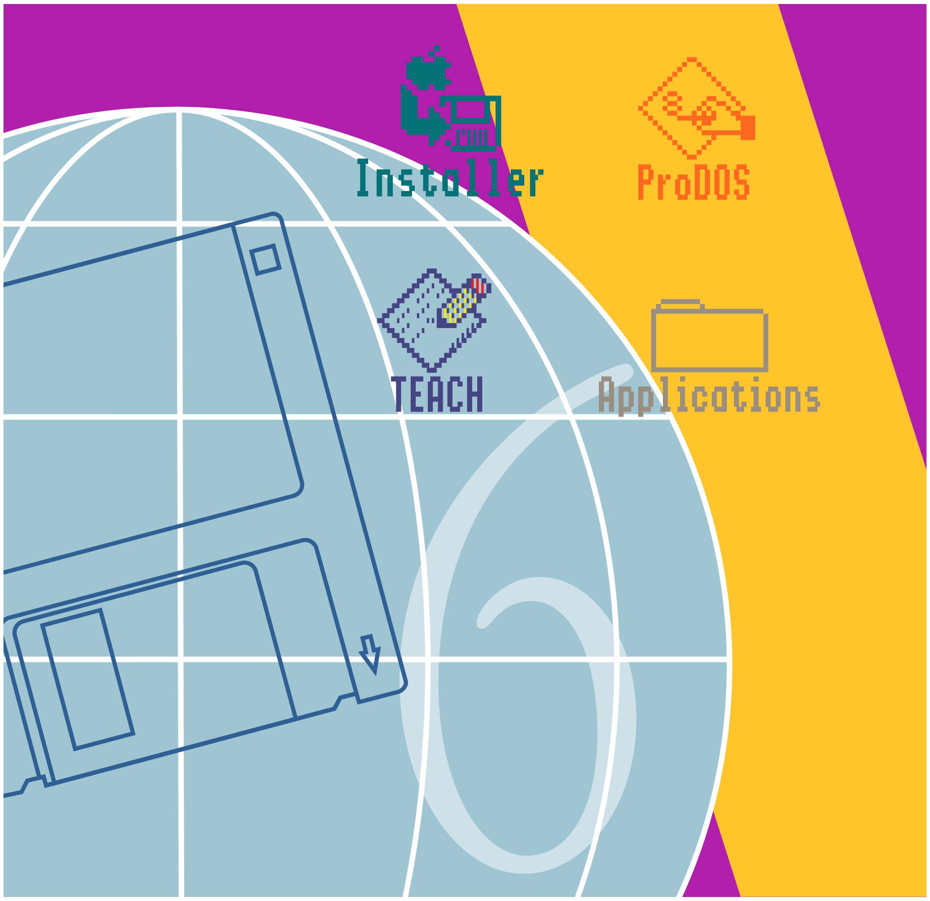
Archiver ERS

Ver. 1.5CD
External



System 6 IIGS

Apple® IIGS® System Software 6.0 — Release Notes
Golden Master 3 Release — March 5, 1992
Archiver ERS v.1.5CD
EXTERNAL

PDF by khaibitgfx@outlook.com
July 8, 2019

20525 Mariani Avenue
Cupertino, CA 95014-6299
(408) 996-1010

khaibitgfx

khaibitgfx@outlook.com

Contents

Introduction and Overview

Archiver is a general-purpose, full-featured data backup utility for the Apple IIGS. It runs under GS/OS™ as a stand-alone Desktop application.

Currently implemented features include:

- **Back up/restore file-by-file:** Allows user to select individual files.
- **Recognize identity and position of backup media during restore:** Robustly prevents user error during restore.
- **Preserve directory structures on restore:** File backup image is “hierarchy aware.”
- **Back up to a file:** Space permitting, any backup can be stored to a GS/OS™ file rather than directly to raw media.
- **Back up/restore entire volume:** Block-by-block volume backup of all blocks or only blocks in use with optional run-length-encoding compression (ProDOS and HFS volumes only).
- **Back up/restore of volumes other than ProDOS and HFS:** For these volumes, all blocks on the volume are backed-up.
- **Automatically find “newer” files** by either comparing mod date to backup date or by checking the “backup needed” bit in the file access word.
- **Provide error summary:** Errors need not interrupt file-by-file backup, but get recorded and reported after completion.
- **Save/restore file lists w/info:** User can keep lists of frequently selected files (with date and type info) and optionally modify the selections before starting a file-by-file backup.
- **Retry bad backup media:** If a bad piece of backup media is encountered during a backup, that piece can be retried without restarting the entire backup operation.
- **Provide pathname substitution between systems:** Automatically or manually corrects differences in file name syntax when moving files between file systems (e.g., from Appleshare to ProDOS).

Backup

The source of backup data is a GS/OS™ volume. The volume may be treated as a collection of files (file-by-file backup) or as a collection of blocks on a device (entire-volume backup). Any GS/OS™ file system can be backed-up with the file-by-file method. Only local volumes can be backed-up as entire volumes.

When file-by-file backup is selected, the user is permitted to choose which files on the selected volume to back up, and all selected files will be read in (using standard GS/OS file calls) and organized into a special format before being sent to the destination routines. This special format, described in the document "Archiver I/O Formats", is designed to support sophisticated file retrieval features.

Restore

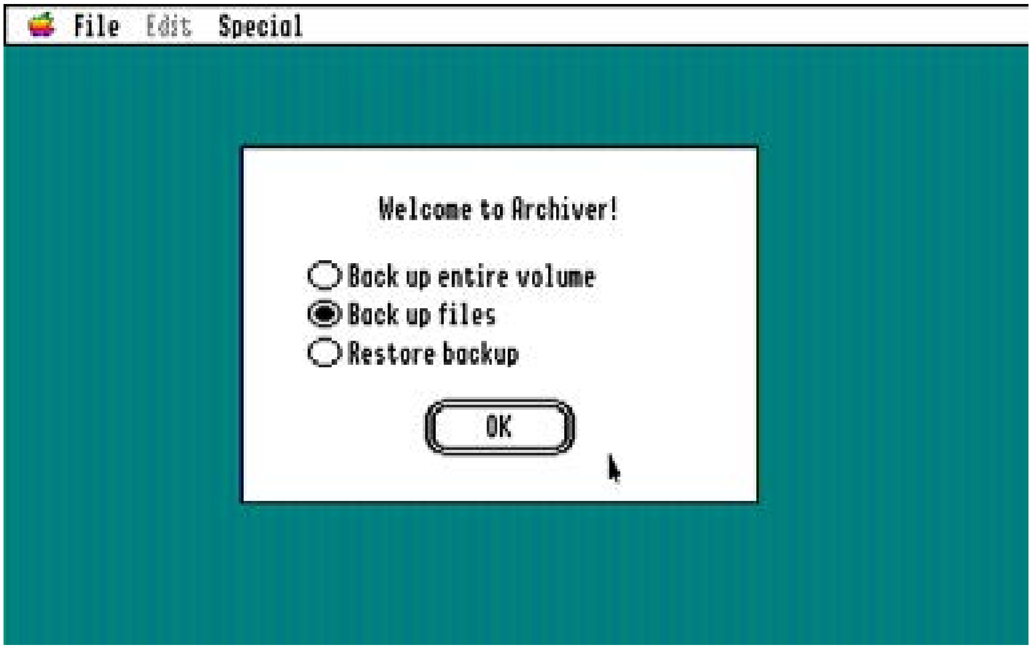
When a file-by-file backup is to be restored, the user must choose a new destination directory which Archiver will create and to which files will be restored. He is then allowed to choose which of the files he wishes to restore from the backup set. If the backup was written to removable media, the user is prompted to insert specific pieces of the backup set as needed.

When a volume backup is to be restored, the user must choose a currently mounted volume to be replaced by the restored image. Archiver will only proceed if the chosen volume is large enough to accept the restored image and is able to be erased by GS/OS, i.e., the volume is writable, and contains no open files.

Design

General Operation

When the Archiver application is launched, it first purges all purgeable Memory Manager handles and then compacts remaining memory. It then checks for GS/OS version 4.0 (corresponding to System 6.0) or later. If that check passes, then Archiver starts up all necessary tools with the requirement that their versions correspond to System 6.0 or later. If either of the preceding operations fails, Archiver displays a text alert window notifying the user that Archiver cannot run unless System 6.0 or later is installed, and then quits.



Welcome to Archiver!

○ Back up entire volume
● Back up files
○ Restore backup

Figure 1. The “pre-greet” window

If the preceding operations succeed, then Archiver attempts to load a user preferences file from the application directory (in the case of a locally-booted system) or the user setup folder (in the case of a network-booted system). If the preferences file cannot be loaded, then Archiver sets the preferences to pre-defined defaults.

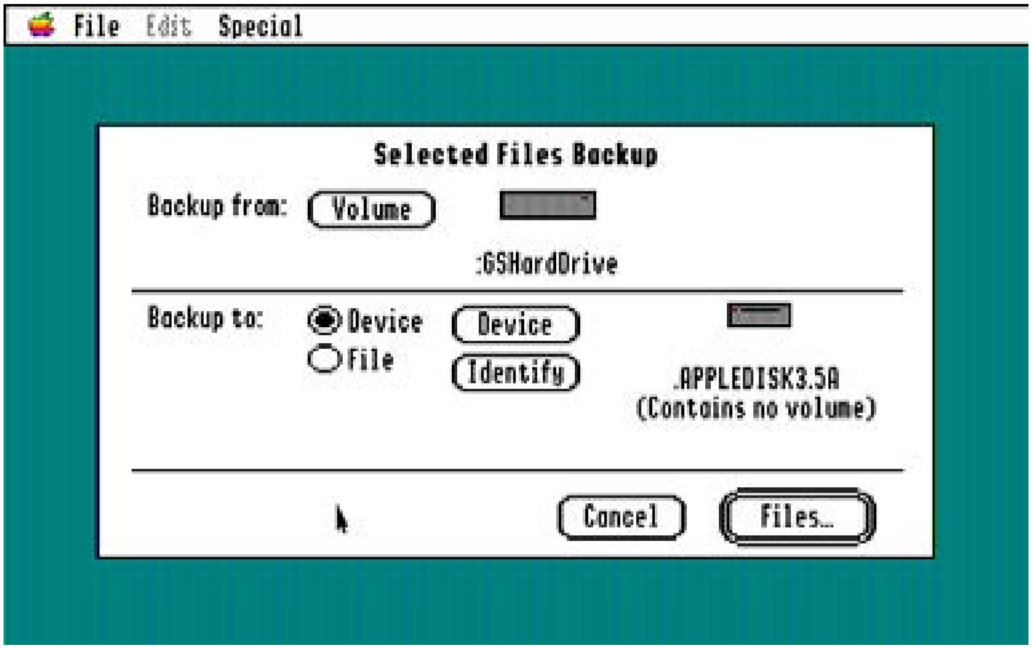
From the user's point of view, Archiver offers three basic functions: back up an entire volume, back up selected files from a volume, and restore from a backup. Internally, the restore function is performed by one of two methods depending on whether the backup image was created as an entire-volume backup or a file-by-file backup. The desired function is selected in the "pre-greet" window, shown in Figure 1, when the application is started.

The "pre-greet" window supports the following button key equivalents:

Key	Action
RETURN	proceed
	activate higher radio button
	activate lower radio button

The associated source and destination are selected in the "greeting" window, shown in Figure 2. If the user clicks the "Back Up"/"Restore"/"Files. . ." button before he has sufficiently specified a source and destination, or if for any reason the source or destination are not suitable for the selected operation, then an alert box is displayed to tell the user the reason he cannot proceed, and he remains at the "greeting" window. Once the source or destination volume is specified, it can be ejected or moved to any device, and Archiver will find it or prompt the user to insert it.

◆ **Note** Once the source or destination device for backup media is selected, that same device must be used throughout the entire backup or restore. ◆



Selected Files Backup

Backup from: :GSHardDrive

Backup to: ● Device ○ File

.APPLEDISK3.5A
(Contains no volume)

Figure 2. The "greeting" window

When the user selects to back up a device, Archiver polls that device for media swap and updates the display in the "greeting" window appropriately. Disks inserted in any other device than the selected one are handled via the standard HandleDiskInsert Tool.

The "greeting" window supports the following button key equivalents:

Key	Button
TAB	Volume
0	Device
?,/	Identify
ESC	Cancel
RETURN	Back Up/Restore/Files

During the normal event loop processing, Archiver polls for disk inserts via the standard HandleDiskInsert Tool and usually allows the standard handlers to prompt the user in the case of unrecognized disk formats or file systems. The exceptions to this rule, as noted in the appropriate sections below, are that the HandleDiskInsert prompts are not used for insertions in the currently selected device in the greeting window during a restore from device, and polling is suspended altogether for all devices except the current backup/restore device during the course of a backup or restore operation. For more information on the functioning of the HandleDiskInsert Tool, consult the latest System 6 Toolbox ERS.

During a backup or restore operation, Archiver robustly detects ejects of the source volume or destination media and asks the user to re-insert them as necessary. Archiver then positively identifies the volume or disk before accepting it, EXCEPT for destination media which have not yet been written to. In general, the user should be able to eject any disk at any time and insert the wrong disk at any time without corrupting a backup or restore. The detection of media swaps is accomplished by checking for disk-switched errors on all disk accesses and through the use of a notification procedure added to the GS/OS notification queue; this notification queue has its event_flags word set to request notifications of disk inserts only, since an eject without an insert will always be detected as a "device offline" error for a device access.

In addition, the GS/OS device dispatcher entry point is patched to allow Archiver to detect accesses of any kind to its destination media when the custom 3.5" floppy routines are in use. This is necessary because those routines write tracks sequentially and assume that the disk head is not moved to a different track between track writes; if an access through the device dispatcher is detected then a flag it set which forces the custom 3.5" floppy routines to recalibrate the head and move it to the correct track before continuing.

Entire Volume Backup

When the user selects the volume to be backed-up and the destination for backup data, he proceeds to the "entire volume backup" window, which is similar to the "file backup" window shown in figure 5. The file system of the source volume is identified in order to determine whether a smart block backup can be performed. Data are then read from the source volume's device using GS/OS™ block device calls. If the preference settings indicate that compression is to be used, then these data are compressed by a run-length-encoding scheme before being sent to the destination routines.

When a ProDOS or HFS volume is selected for entire-volume backup and the user preference to "…back up only blocks in use…" is enabled, Archiver examines the volume allocation bitmap and only backs up blocks marked as in use. When any other type of volume is selected or the described preference is not enabled, Archiver simply reads every block sequentially from the device on which the volume resides, optionally applies run-length-encoding compression, and sends the data to the destination routines. The destination routines handle output to a GS/OS™ file or block device.

Entire-volume backup can only be performed on a local volume, i.e., never on an Appleshare volume. Data on Appleshare volumes may be backed-up by Archiver using the file-by-file method.

As mentioned above, the destination for the backup image can be either a block device or a GS/OS file. Only devices whose ID type is known to Archiver are allowed as a backup destination device. Tape drives are not supported except for the Apple 40MB SCSI Tape Drive.

Note that 5-1/4" floppy drives are not allowed as a backup destination device. When the destination is selected to be a file, the backup image is simply written into a new GS/OS file specified by the user through the Standard File interface. When the destination is selected to be a block device, special formatting is written to each piece of media in addition to a portion of the backup image itself. This special formatting includes headers which contain information to identify the piece of media as part of an Archiver backup set, give information about the type of backup set and where that piece of media belongs in the set, and specify which file was being backed-up when that piece was started (for details on the format of Archiver backup sets, refer to the document "Archiver I/O Formats").

If the backup media can contain a GS/OS volume, then a minimal ProDOS directory image is written to the piece of media to allow it to appear as a ProDOS disk. This volume name is given as "ARCHIVER.XXXXX", where XXXXX is the ordinal number of that piece of media in its backup set, and the bitmap shows that all blocks are in use. In addition, if the destination device is bootable (can contain a local volume) then device-specific boot code is placed on the media which displays a splash screen that provides information about the backup set to which the media belongs, and allows the user to reboot.

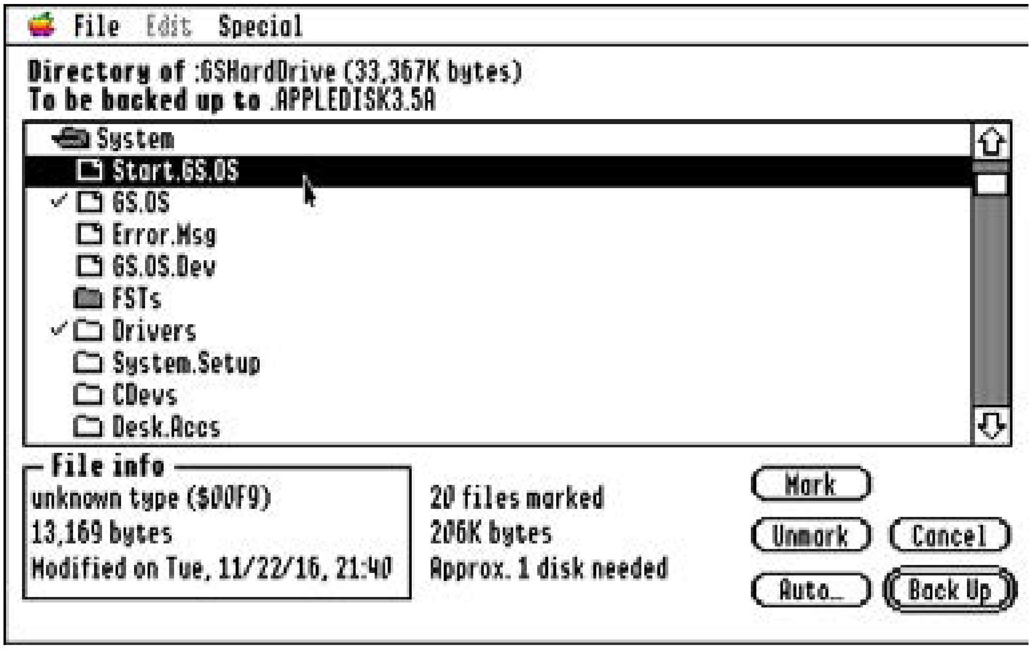
When a piece of media is full on a removable device, a dialog box appears telling the user the piece number just filled and asking him to insert a new blank piece. When the new piece of media comes online, the preparation is performed and the data backup process continues.

If it is determined that the backup medium is bad in the case of removable media, Archiver asks the user to either try again with another piece of media or cancel the backup operation. This feature allows the user to retry a single bad piece without restarting the entire backup operation.

Archiver includes a custom device driver for writing to the Appledisk 3.5 (gray) floppy drive. This driver can write a track's worth of data in a single pass to unformatted media, thus providing a significant performance gain over the standard GS/OS™ driver, especially if the destination media is unformatted at the time of the backup.

Selected Files Backup

When the user selects the volume from which files are to be backed-up and the destination for the backup data, he has the option of either reading in a previously saved file list or reading the directory directly from the source volume; a volume directory is accessed via GS/OS™ calls and is traversed in "depth first" order. The file list is stored in memory in the Table of Contents format which consists of a header followed by variable-size file entries which are laid out sequentially in the order the directory was traversed. Each entry contains the file's filetype, EOF, create date and time, modification date and time, EOF of resource fork, and filename, as well as structural information for internal use by the utility. Entries for directory files have a slightly different format to allow all data about them to reside in their entries so that no information about them need reside in the data portion of the backup stream (for details on the format of Archiver backup sets, refer to the document "Archiver I/O Formats").



Directory of :GSHardDrive (33,367K bytes)
To be backed up to .APPLEDISK3.5A

System
Start.GS.OS
✓ GS.OS
Error.Msg
GS.OS.Dev
FSTs
✓ Drivers
System.Setup
CDevs
Desk.Accs

File info
unknown type ($00F9)
13,169 bytes
Modified on Tue, 11/22/16, 21:40

20 files marked
206K bytes
Approx. 1 disk needed

Figure 3. The "select files" window

The backup stream for a file-by-file backup consists of an image of the Table of Contents, the actual backup data, and another copy of the Table of Contents.

After the volume directory of the selected volume is read in and the Table of Contents is built, the user is presented with the “select files” window as a human interface for marking files desired to be backed-up.

The “select files” window, shown in Figure 3, has the following features:

File List Display

This is implemented as a list control which initially contains the names of all the files at the root level of the source volume’s directory, with Finder-style icons just to the left of the filename. File names may be highlighted by means of the standard interface implemented in the List Manager. Double-clicking on a closed folder causes the folder to open and the names of all the files belonging to that folder to be inserted into the list and displayed at an indented position relative to the folder. Double-clicking on an open folder causes the folder to close and removes the names of all files below that folder in the hierarchy from the list. If the file list was read in from a saved file, folders will be in the open/closed state they were in when the list was saved.

This display also provides information about whether files are marked for backup (or restore): a non-folder file appears with a check symbol () to the left of its icon if it is marked for backup; a folder file appears with a check mark if all of its children are marked because the folder itself was marked explicitly, or filled-in gray with no check mark if any of its children are marked (the pathological case is when all children of a folder have been marked, but not by marking the folder itself explicitly, in which case the folder appears as filled-in gray). If the file list was read in from a saved file, files will be marked as they were in when the list was saved.

By default, whenever a volume directory is read in as part of a file-by-file backup operation, all the files with the “backup needed” bit set in their attributes word are marked for backup.

While in the file list display, the user can save the list and the open/closed state of folders and file markings by selecting the “Save File List…” menu item from the “File” menu. This function presents a Standard File put dialog which allows the user to choose the path of the new or old file which will contain the saved list.

Information Display

In the middle of the space below the list control is the “marked info” section which shows the number of non-folder files marked for backup (or restore), their collective logical size in bytes, and the estimated number of pieces of destination media required to store the backup set.

A second information section, to the left of the “marked info” section in the box labeled “file info,” provides general file information. Highlighting a file’s name causes the file’s size in bytes, filetype, and modification date to be displayed in the section labeled “file info.” The value shown for the size of the file is the sum of the EOF’s of the data fork and resource fork of that file. Note that the size of a closed folder is computed as the sum of the sizes of all the non-directory files beneath that folder in the hierarchy, and the size of an open folder is given as 0, by convention. The text description of the filetype is found by loading up to 6 filetype descriptor files from the “Icons” folder on the system volume and finding the first name that appears in the first non-Apple filetype file if it appears in a non-Apple file, otherwise it is the first name that appears in any of the Apple filetype files. Highlighting more than one file name causes the number of files represented by the selection and their total size in bytes to be displayed.

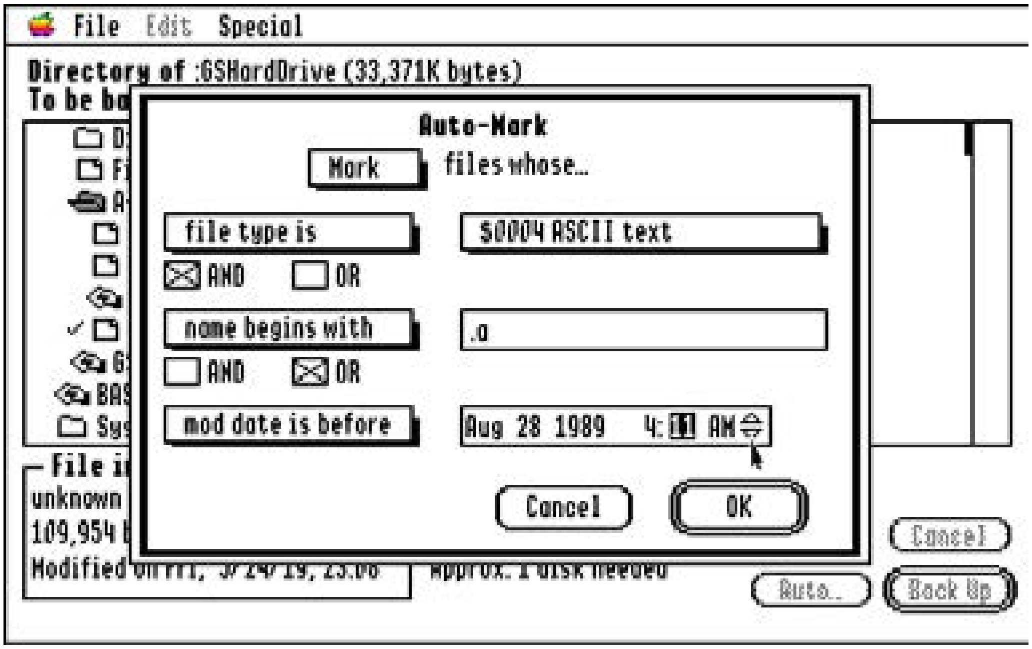
Special Buttons

Clicking the “mark” button causes all files highlighted in the list display to be marked as selected for backup. Clicking the “unmark” button causes all files highlighted in the list display to be unmarked as selected for backup. If a folder file is marked or unmarked, then all files beneath it in the hierarchy are marked or unmarked.

Auto-Mark

The auto-mark window, shown in Figure 4, is invoked by clicking the “Auto. . .” button in the “select files” window. In the auto-mark window the user can mark or unmark files and folders which satisfy a specified logical combination of criteria. Up to three criteria may be specified and they may be combined by logical

AND or OR. Each criterion can be a string which appears at the start, end, or within a filename, a GS/OS™ file type, or a date which is before or after the modification date. Thus, for example, it is possible to unmark all files and folders whose type is $0004 ASCII TEXT AND whose names end in “.C” OR whose modification date is after March 20, 1990, 5:59 PM. The criterion types are selected with popup controls, strings are specified in LineEdit controls, and time and date are specified with a custom control which was implemented specifically for Archiver. Comparisons between strings are case-insensitive.



Directory of :GSHardDrive (33,371K bytes)

Mark files whose...

file type is: $0004 ASCII text

☒ AND ☐ OR

name begins with: .a

☐ AND ☒ OR

mod date is before: Aug 28 1989 4:11 AM

Figure 4. The “auto-mark” window

About the Time/Date Control

The Time/Date Control consists of a number of text fields arranged in a single horizontal line, surrounded by a rectangular border (which is actually the control's rectangle). The Time/Date Control is a targetable control, and may be made the target control by clicking in any of its text fields or by pressing the TAB key in the previous target control in the same window. In addition, the owner of the window in which the control resides can make it the target control at its discretion.

When a Time/Date Control is currently the target, it shows two small arrow icons to the right of the rightmost text field and within the border and shows one of the text fields as highlighted, i.e., text drawn in the background color over a rectangle in the foreground color. If the control was made the target by clicking in a text field, then that text field will be the one highlighted; otherwise, the leftmost field will be highlighted.

The arrows which appear while the control is the target are actually buttons which behave as do the arrow buttons in a standard scroll bar control. Clicking the top arrow increments the value of the parameter represented by the currently highlighted text field and redraws that field; clicking in the bottom arrow decrements the value. If the value represented by the highlighted field is at the limit of its allowable range in the direction of change, then the value will be "wrapped around" to the opposite limit in the allowable range. The arrow buttons will auto-repeat at a pre-defined rate after a pre-defined delay. The currently highlighted text field can be changed by clicking in another text field or by using the applicable key equivalents as described below.

The Time/Date Control supports the following key commands:

Key	Effect
	Increment, decrement value, same as arrow buttons.
	Move hilighting to next text field, if any, in that direction.
TAB	Move hilighting to next text field to the right; if currently highlighted text field is rightmost, then make the next targetable control the target if one exists, otherwise do nothing.
	If currently highlighted field is a numeric field, then "roll" the typed digit into the least-significant position in the value of that field and shift all the digits in that field one position to the left, losing the most significant digit, and redraw the field. If the value resulting from this operation is above the highest allowable value for that field then the most significant digit in that field is internally changed to zero.
0-9	Depending on the setting of the certain bits in the control template, the Time/Date Control can show only date fields, only time fields, or both. All fields always appear on a single line, and the control rectangle is

Depending on the setting of the certain bits in the control template, the Time/Date Control can show only date fields, only time fields, or both. All fields always appear on a single line, and the control rectangle is adjusted accordingly. This option is only respected when the control is created and cannot be changed during the life of the control. For details on how to use these options as a window owner, consult the sourcecode for the Control defproc.

The Control reads the user time and date format preference settings in BRAM when the control is created and changes the format of its display as follows: If the setting for the hour format is set for AM/PM, then the hour field can only be set to or display a value between 1 and 12 and an AM/PM field will appear to the right of the minute's field. If the hour setting is set for 24-hour, then the hour field can be set to or display a value between 0 and 23 and no AM/PM field is shown.

If the date format is set for MM/DD/YY then the date fields appear in the order MONTH-DAY-YEAR. If the date format is set for DD/MM/YY or YY/MM/DD then the date fields appear in the order DAY-MONTH-YEAR (note that in the Time Control Panel implementation, the Time/Date controls displays a second's field for the time and displays date fields in the order YEAR-MONTH-DAY when the BRAM date format is set to YY/MM/DD).

All standard limitations for the values of each field are enforced, including the upper limit for the date for the month of February. In the case where changing the year or month value causes the date value to be above the allowable limit, the date value will be changed to the highest allowable value and the date field will be redrawn.

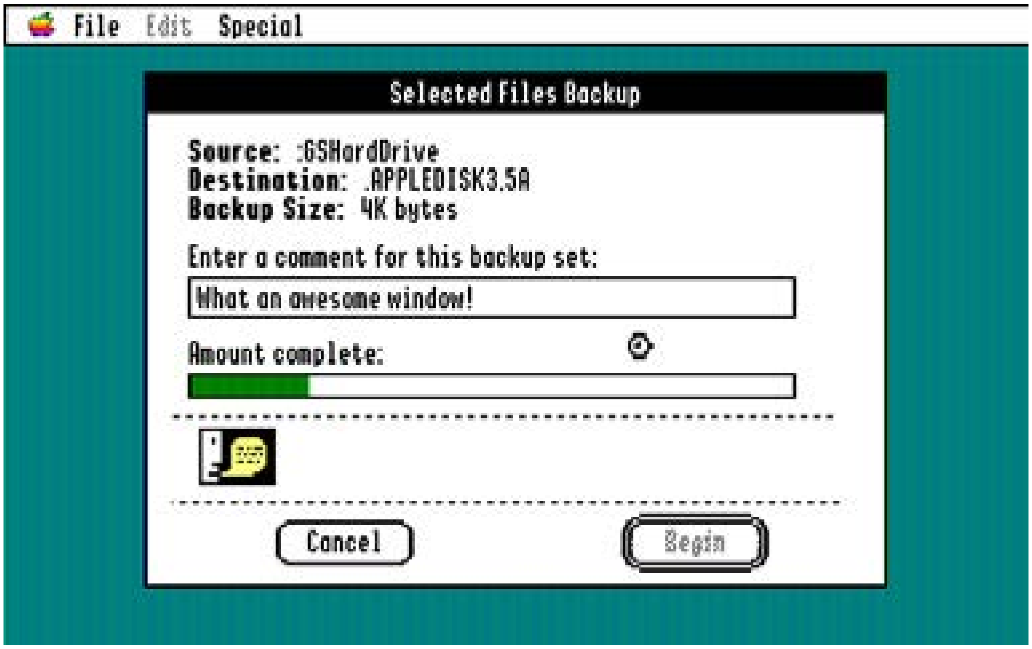
The Time/Date Controls used in the Time Control Panel have slightly different characteristics because the version of the Control defproc used in that module is assembled with different build-time options than the one used in Archiver.

Filetypes are specified in a separate dialog box, invoked by choosing the "choose file type…" item in the filetype popup menu. This box consists of a List control, which contains a list of filetype choices, a LineEdit control which allows the user to enter filetypes in hex, and buttons for "Cancel" and "OK." The list of filetypes is built by merging the filetype descriptor files loaded for the information display in the "select files" window. Only one entry for each major filetype appears in the list; various auxtypes associated with a given major filetype are not listed so that a redundancy by major filetype never occurs. The user can single-click a list entry to choose a filetype, or double-click to choose a filetype and close the dialog box. Upon returning to the "automark" window, the popup control shows the hex value of the specified filetype, and the associated name if the user chose from the list. If the user typed a hex value for the filetype, no name appears.

The "select files" window supports the following button key equivalents:

Key	Button
-M	Mark
-U	Unmark
ESC,-.	Cancel
RETURN	Done

When the user is done marking the files he wishes to back up, he proceeds to the “file backup” window, shown in Figure 5. Here he can enter up to 40 characters as a name for the backup set. This name is of no concern to Archiver, and the internal routines can differentiate between backup sets with the same name, but this name is displayed when the user is asked to decide whether he wishes to restore from that particular set, or when an Archiver backup disk is booted.



Selected Files Backup

Source: :GSHardDrive
Destination: .APPLEDISK3.5A
Backup Size: 4K bytes

Enter a comment for this backup set:

What an awesome window!

Amount complete:

Figure 5. The “file backup - in progress” window

The “file backup” window displays information about the backup operation and contains a thermometer which shows the progress of the backup operation. This window also shows a “talking head” icon in an area where status messages for the user are displayed during the backup operation. Both the thermometer and the status message area remain blank until the user commences the operation by clicking “Begin”.

The backup operation actually begins when the user clicks the “Begin” button. During the course of a backup or restore operation, all standard disk insert handling via the HandleDiskInsert tool is suspended. If the destination for the backup is removable media, then the user is prompted to insert blank media as necessary. Unless user preferences are set to “Destroy media without asking,” the user is warned if the media contains a recognizable volume or a member of another Archiver backup set and is given the chance to swap in a different piece of media. If the user inadvertently inserts an earlier piece in the current backup set, the piece is ejected and an alert box tells the user that he can’t use the disk.

If then custom 3.5” floppy routines are not being used and the destination device is not a tape then Archiver will issue a D_CONTROL call to format the media.

When the destination device is an Apple 40 MB Tape Drive, any tape that the user inserts for backup which passes the checks described above is checked for readability; if a block can be read from the tape then Archiver gives the user the option of re-formatting the tape for maximum reliability, otherwise the user is informed that the tape will require formatting and is given a chance to use another tape. The treatment of the destination device or file is the same as for entire volume backup.

During the backup operation, the “Cancel” button remains active, but only responds intermittently because the utility performs I/O operations in indivisible chunks during which time the user interface is not serviced. If the user clicks in the “Cancel” button, Archiver will eventually respond by asking the user whether he indeed wishes to abort the backup operation. At this point the user can either resume the backup process or abort the backup process and restart Archiver.

If the file list was read in from a saved file rather than by reading the source volume directory, it is possible that certain files on the list no longer actually exist on the volume, or have changed from a folder to a file, or vice-versa. If a file which was marked for backup no longer exists, it will cause a “file not found” error from GS/OS when Archiver attempts to read it. If the user preference to “Interrupt backup for file read errors” is enabled, then an alert box is displayed to tell the user that the file will be skipped. If the backup is completed, then the unfound file will be displayed in the error summary list. If a file has changed to a folder, it is skipped. If a folder has changed to a file, then the file is backed-up anyway.

When the backup image has been completely written out to removable media, the user is prompted to re-insert the first piece in the backup set (if it is not already online) so that the first Table of Contents image may be updated with information calculated while writing the backup image. Archiver then attempts to clear the “backup needed” bit on every file that was backed-up on the source volume. A dialog box then appears to either tell the user that the operation was successfully completed, with buttons to either restart the application or quit, or that some file read errors occurred during the backup, with the option to view the file error summary window.

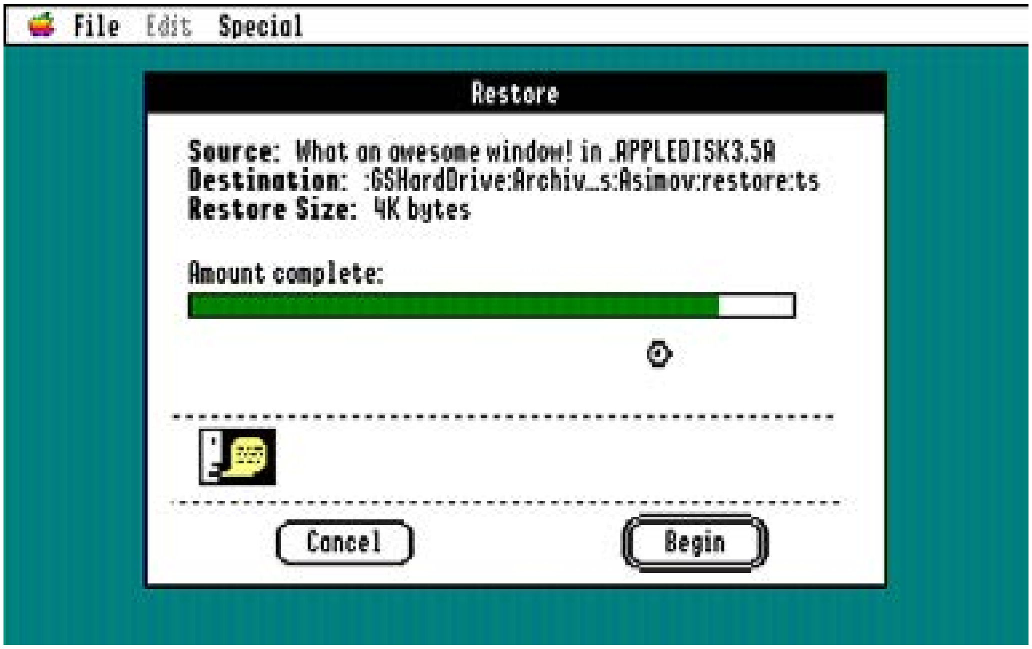
Restore

When the user selects to restore from a device, Archiver polls that device for media swap and updates the display in the “greeting” window appropriately. Disks inserted in any other device than the selected one are handled via the standard HandleDiskInsert Tool. If media is present in the currently selected device, it is checked for Archiver backup data. If it is indeed a member of an Archiver backup set then the user name for the set, the total logical size of the set, and the piece number of that piece of media are displayed. If the media does not contain Archiver backup data, the name of the volume on the media (if any) is displayed below the name of the device.

In checking for a backup set on a piece of media, Archiver first reads the block containing the media header on that piece (i.e., block 1 on a disk, block 0 on a tape) and looks for the Archiver signature in the first 10 bytes of that range. If the signature matches, then Archiver checks that the modification date returned by GET_FILE_INFO for the dummy volume matches the date in the media header; this is necessary to prevent an ERASE_DISK'ed Archiver disk from passing the test because an ERASE_DISK doesn't necessarily destroy the media header in block 1. Note that Archiver never checks the name of the dummy volume ("ARCHIVER. XXXX") for any reason.

If the piece of media contains an entire-volume backup then a "volume" button becomes active in the "Restore to:" area of the window, and the user can use the button to walk through the online volumes until he finds his choice for the destination of the restore. Archiver will display an error message if the user attempts to proceed with a restore to a volume which cannot be erased or which is too small for the restored volume. If the piece of media contains a file-by-file backup then a "folder…" button becomes active; clicking this button produces a Standard File dialog box in which the user can specify a destination folder for restored files. The folder structure of the restored files will be duplicated within this folder.

When a restore operation is to take place from a file-by-file backup, Archiver reads the Table of Contents from the first piece of the backup set and proceeds to the "select files" window as described under "selected files backup." All the functions in this window work as they do for a file-by-file backup operation, except that files are to be marked for restore rather than for backup, and files which were not backed up due to read errors appear with a red "X" to the left of their icons in the list and cannot be marked for restore.



Source: What an awesome window! in .APPLEDISK3.5A
Destination: :GSHardDrive:Archiv…s:Asimov:restore:ts
Restore Size: 4K bytes

Amount complete:

Figure 6. The "restore" window

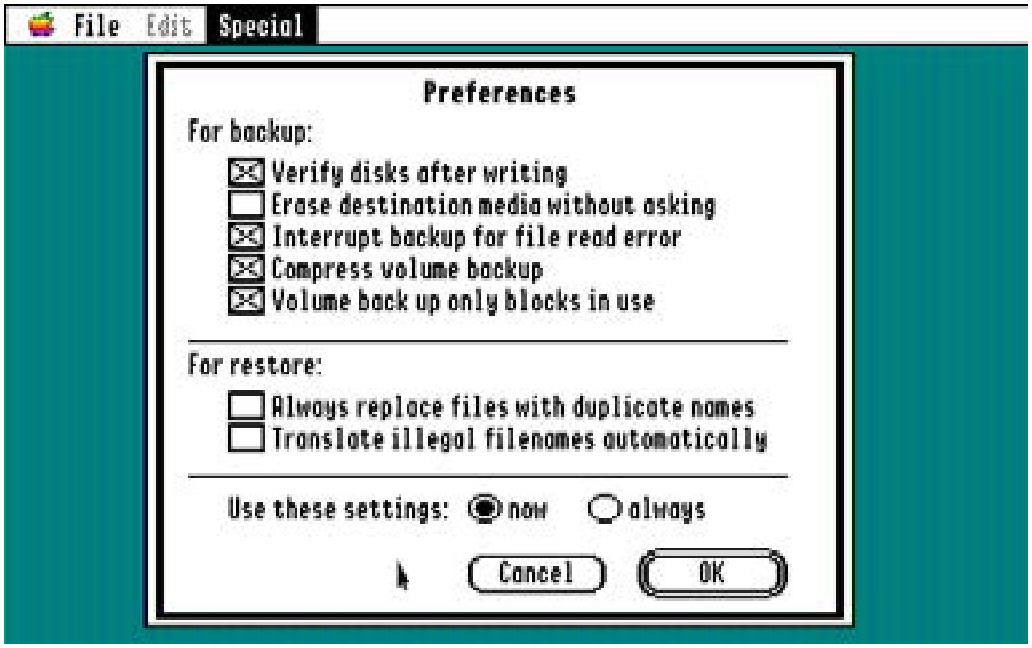
During the course of the restore operation, all standard disk insert handling via the HandleDiskInsert tool is suspended. The user is prompted to insert pieces of media as needed if the source device uses removable media. Pieces are requested by ordinal number within the set, and are identified as having the correct number before they are accepted.

If the user is unable to supply any requested piece of media when attempting to restore from a file-by-file backup, Archiver will display an alert box to notify the user that not all of his files have been restored and some files may be only partially restored. If this is the case during a volume restore, the user is notified that the restored volume will be unusable. In either case the restore is aborted.

The “Cancel” button in the restore window behaves as it does during a backup operation.

Assorted Support Functions

Archiver supports user preferences to control behavior such as error handling and data corruption. These are specified by the user in the preferences window, shown in Figure 7. The effect of each of the preferences are as follows:



Preferences

For backup:

☒ Verify disks after writing
□ Erase destination media without asking
☒ Interrupt backup for file read error
☒ Compress volume backup
☒ Volume back up only blocks in use

For restore:

□ Always replace files with duplicate names
□ Translate illegal filenames automatically

Use these settings: ● now ○ always

Figure 7. The “preferences” window

Verify disks after writing - This option is only meaningful when the special AppleDisk3.5 driver is used during a backup. When the special driver writes out disks, it has no way of knowing whether the data were written correctly, or even if there is real media in the drive during the write operation. Normally, Archiver verifies the entire disk with D_READ calls after using this driver to ensure that the data were really written. Disabling this option skips the verify step, which takes about 30 seconds, but allows a small chance that an error was written to the disk even if it is in good condition. No verify is ever done by Archiver when using the GS/OS drivers because they always verify blocks that they write.

Erase destination media without asking - Normally, when the user backs up to a piece of media, Archiver checks whether a recognizable volume or an Archiver backup exists on that media, and warns the user if one is found. Enabling this option skips this check and the possible warning.

Interrupt backup for file read error - Since one reason a user may want to back up a disk is that he may be experiencing problems using his files due to file structure corruption, Archiver is designed to tolerate file errors during a backup by skipping the bad file a logging the error for reporting later. If a file read error occurs and this option is set, then Archiver will display an alert box informing the user of the error and giving him the option of continuing or aborting. Disabling this option ensures that an unattended backup to a large device will proceed to completion even if file errors are encountered.

Compress volume backup - When this option is set, Archiver applies a simple compression algorithm to volume backup data, thus possibly reducing the amount of storage needed for the backup but requiring more time to complete the backup or to restore the backup.

Volume back up only blocks in use - When this option is set and Archiver is used to back up a ProDOS or HFS volume, Archiver will scan the bitmap to determine which blocks are actually in use on the volume and only back up those blocks. Disabling this option causes all blocks on the volume to be backed up.

Always replace files with duplicate names - Normally, when Archiver encounters a duplicate file name when restoring files, it allows the user to provide a new filename or replace the existing file. When this option is set, duplicate non-folder files are automatically replaced without prompting the user (a duplicate of a folder name always invokes a prompt). This preference is almost useless since the only ways that a duplicate name can be encountered are by it being generated by automatic name translation of illegal syntax (see below) or by being provided by the user as a response to a dialog box presented to handle an illegal or duplicate filename.

Translate illegal filenames automatically - When restoring files to a different file system than the one from which they were backed up, it is possible that a given filename may not be acceptable to the restore file system. Normally, Archiver calls JUDGENAME to correct the name, and that name is presented to the user as the default response in a dialog box requesting a new name for the file. When this option is set, the corrected name given by JUDGENAME is automatically used without prompting the user.

If the user sets the “Use these settings: always” radio button, then the current preferences will be saved in a file in the application directory if the application was locally launched, or in the user’s setup folder in the case of a network launch. Preferences are automatically loaded from this file (if it exists) when Archiver is launched.

Archiver supports saving and loading of file lists to save the user the time required to read the source volume directory and the trouble to select files which are repeatedly backed-up. The user is given an opportunity to load a file list after he clicks “Backup” in the greeting window, but before the directory is read in from the selected volume. If the user decides to use a saved file list, he is presented with a Standard File dialog and prompted for the file he would like to use. If the selected file contains a list which was generated from a volume with a different name than the chosen backup volume, the user is warned and given the option to use the list anyway. If the user still wants to proceed then the saved list is loaded and used in the “select files” window in lieu of a Table of Contents generated from scratch by scanning the entire volume directory; if the user decides not to use a saved list or cancels after attempting to load a saved list, a new one is generated from scratch.

While in the “select files” window, the user can choose “Save File List” from the menu and he will be presented with a Standard File dialog asking him to name the saved file. The saved list file retains the filenames and their hierarchy, the state of opened and closed folders, and the marked/unmarked status of the files.

Implementation

Organization

The basic user-interface shell is implemented in MAX Pascal. This shell starts up the application under GS/OS™ and does most of the work of maintaining the desktop environment for the utility. Almost all Toolbox routines which support the desktop interface are called from the Pascal shell. Most of the windows and controls in Archiver are defined as resources and are invoked through the new Tool calls available with System Disk 5.0 which support resources.

Archiver contains the custom Time/Date control, whose defproc is built as a code resource and loaded from the resource fork by the Control Manager when an instance of the control is created in a window. The Time/Date Control, which was implemented specifically for Archiver by the author, supports a mouse-driven interface for allowing the user to specify a time of day and calendar date for any purpose. A modified version of the Time/Date control is also used in the "Time" CDEV in the Control Panel NDA on System Disk 6.0.

All routines which deal with the large data structures intrinsic to the backup process, such as the Table of Contents, or which perform I/O functions during the backup are implemented in highly-optimized assembly code. These routines are designed to be called from the Pascal shell and are linked together with the object code created by the Pascal compiler.

Memory Management

Responsibility for allocation of system memory is distributed throughout the application according to the scope of the functions for which that memory may require management. For example, the various Tool sets which support the utility may be considered part of the overall environment under which the application runs, so their memory is allocated by the shell. The various assembly language routines require direct-page space, but they are designed to share the same direct page in order to have access to global data. Since none of the assembly routines can be called the "owner" of the shared direct page, that memory is also allocated by the shell. The buffer for the Table of Contents, on the other hand, gets initialized and filled by the routine which reads the directory from the source volume. This routine expands the buffer dynamically as needed since there is no efficient way to know in advance how much room the Table of Contents will require. Therefore, that memory is managed locally by that routine. Similarly, the routines that handle I/O and data compression/

decompression, which are designed to operate independently of the rest of the code, manage their own buffer memory.

◆ **Note** The shell performs deallocation of memory because such an operation is performed in cases that are dictated by the user, such as shutdown or restart of the application. ◆

Archiver performs certain global memory operations at startup in an attempt to secure as much contiguous memory as possible. Presently, these operations are executed by an initialization code segment, and also by the main Pascal shell.

When Archiver allocates buffers for a backup or restore operation, it calls MaxBlock and allocates a block of the returned size minus 32K bytes for reliable operation of the System; the value returned by MaxBlock is stored in a global variable named last_max_block and at location $000002F8 for easy reference when debugging. Since the precise size of various buffers is a factor which can cause a backup or restore bug to manifest itself during a given execution of the application, knowing this value can allow an engineer to reproduce a bug by setting a debugger break on the MaxBlock tool call and substituting the known value for the return value before resuming execution (assuming it is not much larger than the actual returned value).

List of Limitations

- 32 directory levels in file-by-file backup; levels below this are ignored
- Table of Contents must fit on first piece of media
- 32-bit value for byte size of backup stream or any destination device; larger sizes cause OS errors
- 16-bit value for number of entries in TOC
- Largest available block of memory must not exceed 16 MB otherwise Archiver behaves unpredictably
- $10000/5 displayed items in the file list (limited by List Manager Toolset)
- 16-bit value for the number of pieces in a backup set
- 2048 MB HFS volume for entire volume backup; Archiver behaves unpredictably for larger volumes
- 200 characters for filename
- 1024 characters for backup or restore pathname