Apple II
File Type Notes

Developer Technical Support

File Type: $E0 (224)
Auxiliary Type: $8006

Full Name: EZ Backup Saveset document
Short Name: EZ Backup Saveset document

Written by: Peter Easdown and Matt Deatherage September 1990

Files of this type and auxiliary type contain savesets as produced by EZBackup.

EZ Backup is a backup utility that runs under GS/OS. It provides the ability to backup to either a removable block device or to files under any available file system.

For more information on EZ Backup, contact:

EZ-Soft Pty. Ltd.
G.P.O. Box 880
Sydney, N.S.W., 2001
Australia
Telephone: 011.61.2.365.1271
AppleLink: AUST0367

Definitions

The following definition is used in this document in addition to those defined for all Apple II file types:

Date/Time An eight-byte date/time record as used by GS/OS and the Miscellaneous Tools.

The File Format

EZ Backup savesets are divided into three major segments: the header, the file list and the data.

The Header

The header contains information relating to the saveset as a whole. It is a fixed length of 1,024 ($400) bytes and is defined as follows:

backupDateTime	(+000)	**Date/Time**	The date and time on which the backup took place.
fileCount	(+008)	**Word**	The number of files stored in the saveset.
rootDirectory	(+010)	**512 Bytes**	A GS/OS input string containing the full pathname of the top-level directory of the saveset. Note that normally this is the volume name of the volume that was backed up.
fileList	(+522)	**Long**	A pointer used at run-time to point to the first entry of the file list. This field need not be zeroed when writing to disk.
majRelease	(+526)	**Word**	The major release component of the version number of EZ Backup used to create the saveset, as a two-byte integer.
minRelease	(+528)	**Word**	The minor release component of the version number of EZ Backup used to create the saveset, as a two-byte integer.
fileSysID	(+530)	**Word**	The file system ID of the volume that was backed up.
backupType	(+532)	**Boolean Word**	A flag that indicates whether the backup was a full backup or an incremental (changes only) backup. A value of TRUE means the backup is incremental.
selected	(+534)	**Boolean Word**	A run-time flag used to indicate that all files in the directory tree are selected. A value of TRUE indicates that all files are selected.
devIcon	(+536)	**Long**	A number indicating which icon to display for the root level when restoring. This is supplied in case the configuration of the machine doing the restore differs from that of the machine that made the backup. Following are the icon numbers and device types they represent: File Server $FFF5 CD-ROM $FFF8 5.25” Drive $FFF9 RAM Disk $FFFA 3.5” Disk $FFFB 5.25” Disk $FFFC Hard Disk $FFFD
fileListLen	(+540)	**Long**	The length in bytes of the file list (defined later).
totalDisks	(+544)	**Long**	The number of disks required by the saveset (excluding the disk containing the file list). Note that this is not relevant to a file-based backup since a saveset file cannot be larger than the disk on which it resides.
reserved	(+548)	**Word**	Reserved for future use.

backupLen	(+550)	**Long**	The total size in bytes of the backup, including the header, the file list and the data.
reserved	(+554)	**470 Bytes**	Reserved for future use.

The File List

The file list contains a variable number of 128-byte records, the number of which is given by the `fileCount` field in the header. Each file list entry is defined as follows:

nextFile	(+000)	**Long**	A pointer that is used at run-time to point to the next file in the file list. Although only used a run-time, this field is useful in reconstructing the directory hierarchy when restoring savesets.
fileInfo	(+004)	**62 Bytes**	A GS/OS `GetDirEntry` parameter block that describes this file.
dataOffset	(+066)	**Long**	An offset in bytes from the beginning of the file that points to the location within the saveset at which the data fork (if any) is stored. This field is zero if there is no data fork.
resOffset	(+070)	**Long**	An offset in bytes from the beginning of the file that points to the location within the saveset at which the resource fork (if any) is stored. This field is zero if there is no resource fork.
optionListOffset	(+074)	**Long**	An offset in bytes from the beginning of the file that points to the location within the saveset at which the GS/OS option_list (if any) was stored. This field is zero if there is no option_list. Note that EZ Backup does not currently store the option_list for any ProDOS files.
optionListLen	(+078)	**Word**	The length in bytes of the option_list (if any).
parentFile	(+080)	**Long**	A pointer that contains the run-time address of the file list record of the directory that is a parent to the file described by this record. Although only used a run-time, this field is useful in reconstructing the directory hierarchy when restoring savesets.
currentDir	(+084)	**Long**	If the file described by this record is a directory, this field is a pointer to itself. This is supplied primarily for the restore operation so that the directory hierarchy may be rebuilt.
selected	(+088)	**Word**	Used at run-time to indicate a file record selection mode. If this field is zero, then some error occurred during the backup that prevented the file from being backed up. A non-zero value indicates that the file was correctly included in the saveset; only attempt to restore files that have a non-zero selection mode.
created	(+090)	**Boolean Word**	A flag used at restore time to indicate whether the file was created successfully. A value of TRUE means yes while FALSE means no.
fileName	(+092)	**36 Bytes**	A GS/OS output string containing the name of the file.

The Data

The data component of the file contains the contents of all of the files described in the file list in a contiguous stream of bytes. Each file begins at a 512-byte boundary. Each file list record takes 128 bytes, and any remaining bytes in the last 512-byte block of the file list are unused. Similarly, all stored data forks, resource forks, and option_lists start on 512-byte boundaries, and any remaining bytes in their last 512-byte blocks are unused.

Further Reference

- *GS/OS Reference*