Name ar – maintain portable archive or library

Synopsis

```
/usr/bin/ar -d [-SVv] archive file...

/usr/bin/ar -m [-abiSVv] [posname] archive file...

/usr/bin/ar -p [-SsVv] archive [file]...

/usr/bin/ar -q [-cSVv] archive file...

/usr/bin/ar -r [-abciuSVv] [posname] archive file...

/usr/bin/ar -t [-SsVv] archive [file]...

/usr/bin/ar -x [-CSsTVv] archive [file]...

/usr/xpg4/bin/ar -d [-SVv] archive file...

/usr/xpg4/bin/ar -m [-abiSVv] [posname] archive file...

/usr/xpg4/bin/ar -p [-SsVv] archive [file]...

/usr/xpg4/bin/ar -q [-cSVv] archive file...

/usr/xpg4/bin/ar -r [-abciuSVv] [posname] archive file...

/usr/xpg4/bin/ar -t [-SsVv] archive [file]...

/usr/xpg4/bin/ar -x [-CSsTVv] archive [file]...
```

Description The `ar` utility maintains groups of files combined into a single archive file. Its main use is to create and update library files. However, it can be used for any similar purpose. The magic string and the file headers used by `ar` consist of printable ASCII characters. If an archive is composed of printable files, the entire archive is printable.

When `ar` creates an archive, it creates headers in a format that is portable across all machines. The portable archive format and structure are described in detail in `ar.h`(3HEAD). The archive symbol table described there is used by the link editor `ld`(1) to effect multiple passes over libraries of object files in an efficient manner. An archive symbol table is only created and maintained by `ar` when there is at least one object file in the archive. The archive symbol table is in a specially named file that is always the first file in the archive. This file is never mentioned or accessible to the user. Whenever the `ar` command is used to create or update the contents of such an archive, the symbol table is rebuilt. The `-s` option described below forces the symbol table to be rebuilt.

Options The following options are supported:

`-a` Positions new *file*s in *archive* after the file named by the *posname* operand.

`-b` Positions new *file*s in *archive* before the file named by the *posname* operand.

`-c` Suppresses the diagnostic message that is written to standard error by default when *archive* is created.

`-C` Prevents extracted files from replacing like-named files in the file system. This option is useful when `-T` is also used to prevent truncated file names from replacing files with the same prefix.

`-d` Deletes one or more *file*s from *archive*.

`-i` Positions new *file*s in *archive* before the file named by the *posname* operand. This option is equivalent to `-b`.

`-m` Moves *file*s. If `-a`, `-b`, or `-i` with the *posname* operand are specified, the `-m` option moves *file*s to the new position. Otherwise, `-m` moves *file*s to the end of *archive*.

`-p` Prints the contents of *file*s in *archive* to standard output. If no *file*s are specified, the contents of all files in *archive* are written in the order of the archive.

`-q` Quickly appends *file*s to the end of *archive*. Positioning options `-a`, `-b`, and `-i` are invalid. The command does not check whether the added *file*s are already in *archive*. This option is useful to avoid quadratic behavior when creating a large archive piece-by-piece.

`-r` Replaces or adds *file*s in *archive*. If *archive* does not exist, a new archive file is created and a diagnostic message is written to standard error, unless the `-c` option is specified. If no *file*s are specified and the *archive* exists, the results are undefined. Files that replace existing files do not change the order of the archive. If the `-u` option is used with the `-r` option, only those files with dates of modification later than the archive files are replaced. If the `-a`, `-b`, or `-i` option is used, the *posname* argument must be present and specifies that new files are to be placed after (`-a`) or before (`-b` or `-i`) *posname*. Otherwise, the new files are placed at the end.

`-s` Forces the regeneration of the archive symbol table even if `ar` is not invoked with an option that will modify the archive contents. This command is useful to restore the archive symbol table after the `strip`(1) command has been used on the archive.

`-S` When building the archive symbol table, force the use of the 64–bit capable symbol table format. By default, the 32–bit format is used for all archives smaller than 4GB, and the larger format is used for larger archives that exceed the 32–bit limit.

`-t` Prints a table of contents of *archive*. The files specified by the *file* operands are included in the written list. If no *file* operands are specified, all files in *archive* are included in the order of the archive.

`-T` Allows file name truncation of extracted files whose archive names are longer than the file system can support. By default, extracting a file with a name that is too long is an error. In that case, a diagnostic message is written and the file is not extracted.

`-u` Updates older files. When used with the `-r` option, files within *archive* are replaced only if the corresponding *file* has a modification time that is at least as new as the modification time of the file within *archive*.

`-v` Gives verbose output. When used with options `-d`, `-r`, or `-x`, the `-v` option writes a detailed file-by-file description of the archive creation and the constituent *file*s, and maintenance activity. When used with `-p`, `-v` writes the name of the file to the standard output before writing the file itself to the standard output. When used with `-t`, `-v` includes a long listing of information about the files within the archive. When used with `-x`, `-v` prints the filename preceding each extraction. When writing to an archive, `-v` writes a message to the standard error.

`-V` Prints its version number on standard error.

`/usr/xpg4/bin/ar` The following options are supported for `/usr/xpg4/bin/ar`:

`-v` Same as the `/usr/bin/ar` version, except when writing to an archive, no message is written to the standard error.

`-x` Extracts the files named by the *file* operands from *archive*. The contents of *archive* are not changed. If no *file* operands are given, all files in *archive* are extracted. If the file name of a file extracted from *archive* is longer than that supported in the directory to which it is being extracted, the results are undefined. The modification time of each *file* extracted is set to the time *file* is extracted from *archive*.

Operands The following operands are supported:

archive A path name of the archive file.

file A path name. Only the last component is used when comparing against the names of files in the archive. If two or more *file* operands have the same last path name component (see `basename`(1)), the results are unspecified. The implementation's archive format will not truncate valid file names of files added to or replaced in the archive.

posname The name of a file in the archive file, used for relative positioning. See options `-m` and `-r`.

Environment Variables See `environ`(5) for descriptions of the following environment variables that affect the execution of `ar`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, and `NLSPATH`.

`TMPDIR` Determine the pathname that overrides the default directory for temporary files, if any.

`TZ` Determine the timezone used to calculate date and time strings written by `ar -tv`. If `TZ` is unset or null, an unspecified default timezone is used.

Exit Status The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

Attributes See `attributes`(5) for descriptions of the following attributes:

`/usr/bin/ar`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/object-file
Interface Stability	Committed

`/usr/xpg4/bin/ar`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Committed
Standard	See `standards`(5).

See Also `basename`(1), `cpio`(1), `ld`(1), `lorder`(1), `strip`(1), `tar`(1), `ar.h`(3HEAD), `a.out`(4), `attributes`(5), `environ`(5), `standards`(5)

Notes If the same file is mentioned twice in an argument list, it may be put in the archive twice.

By convention, archives are suffixed with "`.a`".

When inserting ELF objects into an archive file, `ar` might add "`\n`" characters to pad these objects to an 8–byte boundary. Such padding improves the efficiency with which `ld`(1) can access the archive. Only ELF object files are padded in this way. Other archive members are not altered. When an object with such padding is extracted from an archive, the padding is not included in the resulting output.

It is faster to create a new archive from scratch than to insert individual files into an existing archive via separate calls to `ar`. When possible, the recommended strategy is to remove the existing archive, and recreate it with a single `ar` invocation.

The overall size of an archive is allowed to exceed 4GB. However, the size of any individual file within an archive is limited to 4GB by the archive file format. See `ar.h`(3HEAD).

The maximum user ID and group ID for an individual file within an archive are limited to 6 decimal digits by the archive file format. Any file with a user or group ID greater than 999999 is quietly set to user ID "nobody" (60001) or group ID "nobody" (6001). See `ar.h`(3HEAD).