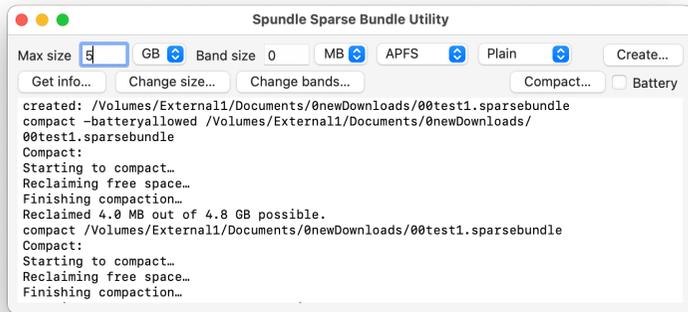


# Start



Spundle is a utility for creating and maintaining sparse bundle disk images. It does this by providing a convenient wrapper to the `hdiutil` command, one of the most complex in macOS. If you are already fluent in use of `hdiutil`, Spundle doesn't offer anything you can't already do in Terminal. It only works with **sparse bundles**; if you want to work with other types of disk image, buy C-Command's excellent [DropDMG](#).

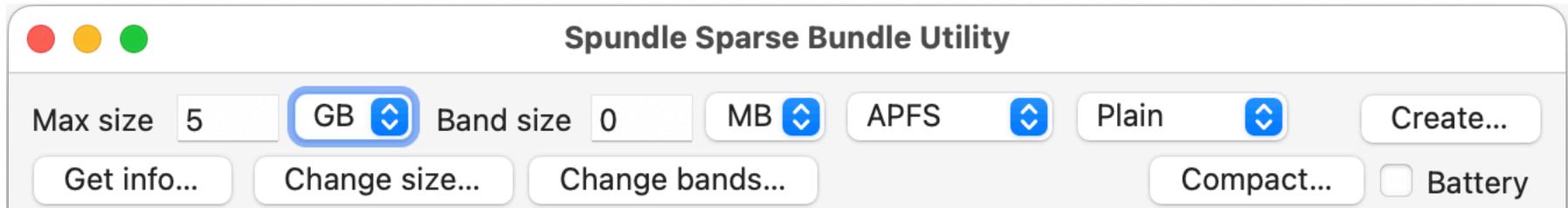
Spundle has five functions:

- to create a new empty sparse bundle, enter settings such as its maximum size and file system in the top row of controls and click on the **Create...** button → [Create](#)
- to view details of an existing sparse bundle, click on the **Get info...** button and select the item to inspect → [Get info](#)
- to resize an existing sparse bundle, ensure that it's unmounted, set the new maximum size in the top row and click on the **Change size...** button below → [Change size](#)
- to change the size of the bands in an existing sparse bundle, ensure that it's unmounted, set the new band size in the top row and click on the **Change bands...** button below → [Change bands](#)
- to compact an existing sparse bundle, ensure that it's unmounted then click on the **Compact...** button → [Compact](#).

→ [Updates](#)

→ [Technical Information](#)

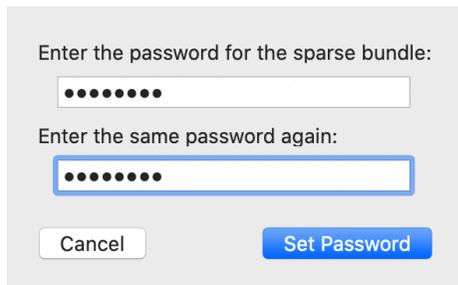
# Create



To create a new empty sparse bundle, specify:

- its **maximum size** by entering the number and units (MB, GB or TB) at the left of the upper row of controls;
- if you wish to set a **custom band size**, enter the number and select its units (MB or GB) in the centre. Leave this number set to 0 (zero) if you want to use the default band size for that format;
- select the **file system** to use on the sparse bundle from HFS+ (Mac Extended), HFS+J (journaled Mac Extended), APFS (Apple File System), Case-sensitive APFS, exFAT (exFAT, NTFS), MS-DOS (FAT32), or UDF;
- whether you wish the sparse bundle to be encrypted. If not, leave the popup set to Plain. For encryption, select either the AES-128 or AES-256 options in that menu.

When you're happy with those settings, click on the **Create...** button.



If you've selected either encryption option, you'll be prompted to enter the password to be used. You must then enter exactly the same password in both the sheet's text boxes and click on the **Set Password** button, or **Cancel**.

→ [Create](#) (concluded)

## Create *(concluded)*

You'll then be prompted to name and locate the sparse bundle to be saved, using a standard Save File dialog. The name you give will also be used to name the new volume.

 Spundle doesn't check whether any given options set here are wise or even valid for that file system. If the combination is not permitted, an error will be reported in the lower view in the window.

Sizes given for the **maximum size** are megabytes, in which 1 MB = 1,024 KB = 1,048,576 bytes, gigabytes or terabytes. The range permitted is 1–999 MB/GB/TB, enabling you to specify sizes between 1 MB and 999 TB. You can use integer or decimal numbers.

Those given in the **band size** are converted to 512 byte sectors as used by `hdiutil`. The conversion process multiplies by 2,000 for MB, so 1 MB is actually 1,024,000 bytes rather than a round million. The range permitted is 1–1,000 MB/GB, enabling you to specify sizes between 1 MB (1.024 MB) and 1 TB (1.024 TB). You can use integer or decimal numbers.

Change the size of the output text using ⌘+ to enlarge and ⌘– to reduce the size.

Each button, including **Create...**, also has a menu command in the **File** menu, and a command key shortcut shown there.

→ [Get info](#)

→ [Change size](#)

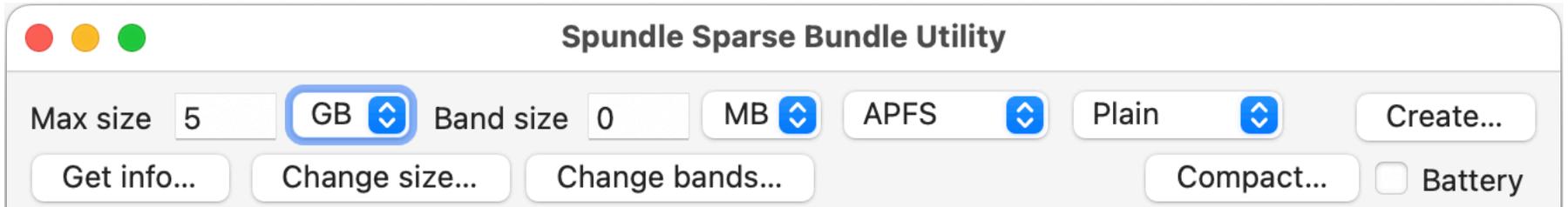
→ [Change bands](#)

→ [Compact](#)

→ [Technical information](#)

→ [Start](#)

# Get info



To see detailed information about any sparse bundle, click on the **Get info...** button. You'll then be prompted to select the sparse bundle to be examined in a standard Open File dialog, and those details will be shown in the view below, replacing all existing text in that view. You can copy text from that to any text editor.

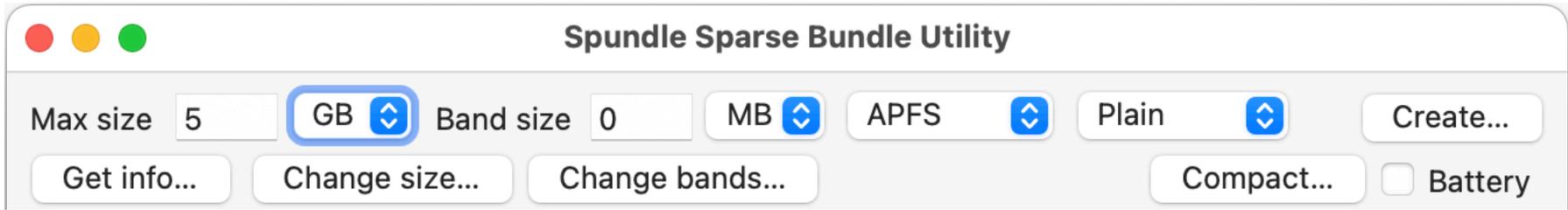
Change the size of the output text using ⌘+ to enlarge and ⌘- to reduce the size.

Each button, including **Get info...**, also has a menu command in the **File** menu, and a command key shortcut shown there.

→ [Create](#)      → [Change size](#)    → [Change bands](#)      → [Compact](#)      → [Technical information](#)

→ [Start](#)

# Change size



To change the **maximum size** of any sparse bundle, first ensure that it's unmounted, then specify its new size by entering the number and units (MB, GB or TB) at the left of the upper row of controls. When you're happy with those settings, click on the **Change size** button below. You'll then be prompted to select the sparse bundle whose size is to be changed using a standard Open File dialog, and the change will be applied.

⚠ Spundle doesn't check whether resizing is wise or possible. If it's not possible, an error will be reported in the lower view in Spundle's window.

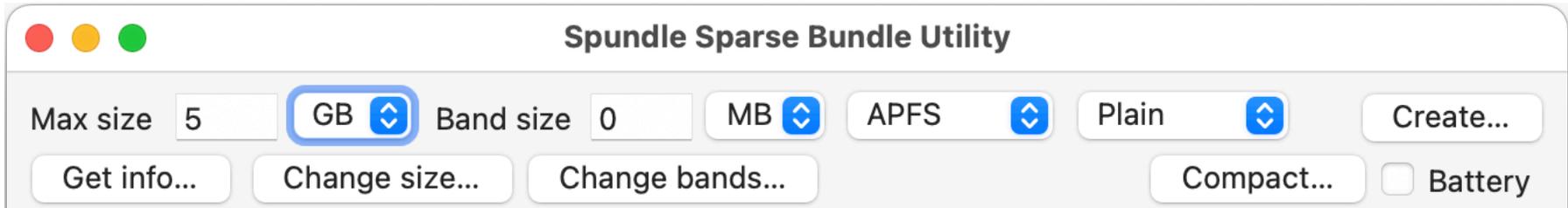
Sizes given for the **maximum size** are megabytes, in which 1 MB = 1,024 KB = 1,048,576 bytes, gigabytes or terabytes. The range permitted is 1–999 MB/GB/TB, enabling you to specify sizes between 1 MB and 999 TB. You can use integer or decimal numbers.

Change the size of the output text using ⌘+ to enlarge and ⌘– to reduce the size.

Each button, including **Change size...**, also has a menu command in the **File** menu, and a command key shortcut shown there.

→ [Create](#)      → [Get info](#)      → [Change bands](#)      → [Compact](#)      → [Technical information](#)

# Change bands



To change the **band size** of any sparse bundle, first ensure that it's unmounted, then specify the new size by entering the number and units (MB or GB) in the middle of the upper row of controls. When you're happy with those settings, click on the **Change bands...** button below. You'll then be prompted to select the sparse bundle whose band size is to be changed using a standard Open File dialog, and the change will be applied. Sizes given in the **band size** are converted to 512 byte sectors as used by `hdiutil`. The conversion process multiplies by 2,000 for MB, so 1 MB is actually 1,024,000 bytes rather than a round million. The range permitted is 1–1,000 MB/GB, enabling you to specify sizes between 1 MB (1.024 MB) and 1 TB (1.024 TB). You can use integer or decimal numbers.

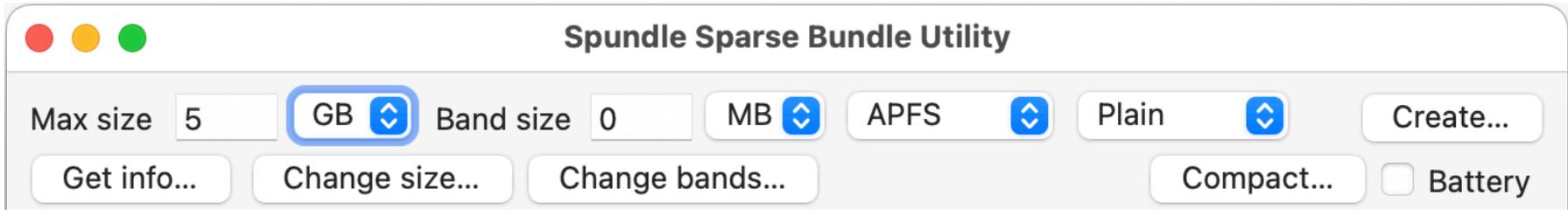
Sparse bundles are unusual in storing their data not in a single file, but in many smaller files known as *bands*. These grow in number to accommodate the data in the sparse bundle, but each band is limited to a maximum size, by default 8.4 MB. Because the bands are all stored in a single folder, if their number of bands exceeds about 100,000 there is an increasing risk of problems. When making sparse bundles with large maximum sizes, consider increasing their band size. A useful guide is to divide the maximum size of the sparse bundle by the maximum number of files, such as 50,000 (or less).

⚠ This feature isn't currently available for encrypted sparse bundles. If you try to apply this to an encrypted original, Spundle will return an error. Otherwise Spundle doesn't check whether resizing is wise or possible. If it's not possible, an error will be reported in the lower view in Spundle's window.

Change the size of the output text using ⌘+ to enlarge and ⌘– to reduce the size.

→ [Create](#)      → [Get info](#)      → [Change size](#)      → [Compact](#)      → [Technical information](#)

# Compact



To compact a sparse bundle, first ensure that it's unmounted, then click on the **Compact...** button. You'll next be prompted to select the sparse bundle which is to be compacted using a standard Open File dialog, and the compaction will be performed.

⚠ Spundle doesn't check whether compaction is wise or possible. Compaction can only be performed at present on HFS+, HFS+J and APFS sparse bundles. Results are reported in the lower view in Spundle's window.

⚠ macOS normally doesn't permit compaction when a notebook is running on battery power, and returns an odd error instead. The **Battery** checkbox allows you to override this behaviour: tick that box if you're confident that compaction can be completed on battery power alone. If compaction results in an error or failure, you'll be warned that this could be the result of running on battery power.

Change the size of the output text using ⌘+ to enlarge and ⌘- to reduce the size.

Each button, including **Compact...**, also has a menu command in the **File** menu, and a command key shortcut shown there.

→ [Create](#)

→ [Get info](#)

→ [Change size](#)

→ [Change bands](#)

→ [Technical information](#)

# Updates

Whenever you open Spundle, it may check to see if an update is available. This *doesn't* use the popular Sparkle mechanism for updating in place, but works as detailed here.

Once Spundle has successfully completed its integrity check, it checks whether update checking has been turned off in its preferences file. If that has, it abandons any attempt to check for updates. If checking is allowed, it then checks when it last checked for updates. If that was more than 12 hours ago, it continues to perform the check. It then connects to my GitHub server, from where it downloads a list of current versions of my apps. It doesn't upload any data to the GitHub server at all, and no statistics beyond GitHub normal connection figures are collected either: no personal identifiers are recorded. If there is an update available, Spundle then checks that its location is on this WordPress blog, and posts a dialog which invites you to download the update.

If you click on the **Download** button, it then points your default browser at that update, which should trigger the update to be downloaded to your normal downloads folder. The update is received as a regular Zip archive, and is exactly the same as you would download from the Downloads page here. It also carries a quarantine flag, so that when you unZip it and install the app inside, it undergoes normal first run 'Gatekeeper' security checks. If you click on the **Ignore** button, Spundle won't remind you about it again for another 12 hours.

An additional item at the end of the **Help** menu explains the update status. If no update check is performed, or the check fails, the last item reads **Update not checked**. If the check is performed and update information is obtained, even when no update is available or you decline to download it, that menu item reads **Checked for update** and is ticked (but still disabled).

You can customise this behaviour by changing Spundle's preferences. The keys to use are:

- `noUpdateCheck`, a Boolean. When set to `true`, this disables all update checking. Default is `false`.
- `updateCheckInt`, a real number (Double). When set to a value greater than 1.0, the minimum time interval between checks, in seconds. Default is 43200, which is 12 hours. If you set it to any value less than 1, Spundle will reset it automatically to that default.

To change either of these, use a Terminal command of the form

```
defaults write co.electiclight.Spundle updateCheckInt '10'
```

which works properly through the preferences server `cfprefsd`.

# Technical Information

Spundle performs its functions using the `hdiutil` command. The basic commands which it uses are

- `hdiutil create -size [maxSize] -type SPARSEBUNDLE [-encryption [AES-128|AES-256] -stdinpass] -imagekey sparse-band-size=[bandSize] -fs [fileSystem] -volname [filename] [filename.sparsebundle]`
- `hdiutil imageinfo [filename.sparsebundle]` and `hdiutil isencrypted [filename.sparsebundle]`
- `hdiutil resize -size [maxSize] [filename.sparsebundle]`
- `hdiutil convert [filename.sparsebundle] -format UDSB -o [filename2.sparsebundle] -imagekey sparse-band-size=[bandSize]`
- `hdiutil compact [filename.sparsebundle]`

where *maxSize* is a size specifier such as 20m for 20 MB, *bandSize* is the number of 512 byte sectors ranging from 2048 to 16777216, *fileSystem* is one of the supported file systems such as APFS, and *filename* is the selected filename from the open/save file dialog.



Not all variations of settings or functions are applicable to any given sparse bundle. For example, Spundle lets you try to compact sparse bundles of all file systems, but that command only works on HFS+ and APFS. No attempt is made to check whether any given resize operation will succeed. Most importantly, Spundle doesn't prevent you from trying to create APFS sparse bundles in macOS Sierra.

When resizing sparse bundles using APFS on High Sierra or Mojave, the container may not be expanded to use the new size. To fix that, mount the sparse bundle, and use the command

```
diskutil list
```

to list all mounted disks. Identify your sparse bundle within that list and within that the container's identifier, such as *disk5s1*. Use that specifier in the command

```
diskutil apfs resizeContainer disk5s1 0
```

to resize its container correctly. This isn't needed in Catalina, where this bug was fixed.

For other functions on sparse bundles and to work with other types of disk image, buy C-Command's excellent [DropDMG](#).

# Change list

1.8:

- added support for Case-sensitive APFS.

1.7:

- fixed a bug entering passwords in Sonoma.

1.6:

- output view now set to non-editable
- rebuilt to support Sonoma.

1.5:

- added Battery checkbox override of compaction failure. Thanks to Bruce for reporting and explaining this.

1.4:

- adjusted window layout and built using Xcode 13 for improved compatibility with recent macOS.

1.3:

- first Universal App version to run on all Macs, including Apple Silicon with Big Sur.
- 

1.2:

- fixed maxima and formatting of bundle and band sizes
- fixed bug in handling size when changing bundle sizes.

1.1:

- added Get info and Change bands commands
- added encryption (currently without Change bands support)
- added menu commands
- changed units for maximum size
- added ... to buttons.

1.0:

- first release.

1 December 2023.