Archiving with tar

The `tar` command can be used to archive files, originally designed for storing data on Tape archives. It allows you to store multiple files and directories as a single file while retaining all the file attributes, such as owner, permissions, and so on. The file created by the `tar` command is often referred to as a tarball. In this recipe, we will learn how to create archives using `tar`.

Getting ready

The `tar` command comes by default with all Unix-like operating systems. It has a simple syntax and is a portable file format. It supports these arguments: `A`, `c`, `d`, `r`, `t`, `u`, `x`, `f`, and `v`. Each of these options can be used independently for different purposes corresponding to it.

How to do it...

We can use tar to create archives, and perform operations on existing archives. Let's see how:

1. To archive files with tar, use the following syntax:

   ```
   $ tar -cf output.tar [SOURCES]
   ```

   For example:

   ```
   $ tar -cf output.tar file1 file2 file3 folder1 ..
   ```

2. To list files in an archive, use the `-t` option:

   ```
   $ tar -tf archive.tar
   file1
   file2
   ```

3. In order to print more details while archiving or listing, use the `-v` or the `-vv` flag. This feature is called verbose (`v`), which for most of the commands will turn on printing more details on the terminal. For example, using verbose you could print more details, such as file permissions, owner group, modification date, and so on:

   ```
   $ tar -tvf archive.tar
   -rw-rw-r-- shaan/shaan       0 2013-04-08 21:34 file1
   -rw-rw-r-- shaan/shaan       0 2013-04-08 21:34 file2
   ```

> The file name must appear immediately after the -f and it should be the last option in the argument group. For example, if you want verbose output, you should use the options like this:
>
> ```
> $ tar -cvf output.tar file1 file2 file3 folder1 ..
> ```

How it works...

In this command, `-c` stands for "create file" and `-f` stands for "specify filename".

We can specify folders and filenames as `SOURCES`. We can use a list of file names or wildcards such as `*.txt` to specify the sources. When finished, `tar` will archive the source files into a file called `output.tar`.

We cannot pass hundreds of files or folders as command-line arguments because there is a limit. So, it is safer to use the append option (see below) if many files are to be archived.

There's more...

Let's go through additional features that are available with the `tar` command.

Appending files to an archive

Sometimes we may need to add files to an archive that already exists, we can use the append option `-r` for this.

In order to append a file into an already existing archive use:

```
$ tar -rvf original.tar new_file
```

Let's create an archive with one text file in it:

```
$ tar -cf archive.tar hello.txt
```

To list the files present in the archive, use:

```
$ tar -tf archive.tar
hello.txt
```

Now add another file to the archive and list its contents again:

```
$ tar -rf archive.tar world.txt
$ tar -tf archive.tar
hello.txt
world.txt
```

The archive now contains both the files.