Dear Lifehacker, can you help me understand the different file compression formats? Looks like there's a ton out there (zip, gzip, bzip1, tar, rar, 7z, etc.), but what are the differences between them, and which one should I use? Signed, CinnamonsterDear Cinnamonster,This is a good question, as it seems that all the other files you download nowadays are in a different compression format, especially if you use multiple operating systems. First, we'll provide a very brief overview of how these compression formats work and the features that matter. Next, let's notice some of the most popular formats and how they compared to each other. Because The WorkArchive File formats are most commonly used for two reasons: the ability to combine multiple files into one (for simple upload or download) and compress them (so they take less space). Many formats also provide other features such as encryption, but let's not get into it today. G/O Media can get a teeth whitening kit from MySmile Commissions, most formats will archive and compress a group of files for you, some formats are designed just to make one or the other. Tar, for example, only archives files, while gzip and bzip only compress files (and are not able to combine multiple files into one). That's why you often see formats like tar.gz, which combine both formats into one to archive multiple files and compress them. In general, unless you have special needs (such as encryption), the three features you really want to look for in a file format are compression capability (how much space you can save), compression time (how fast a particular file can be compressed or decompressed), and the software you will need to use the format. How to choose the right archiving format for your needsIf you are filing files for others, I recommend using simple old zip. Unfortunately, most other formats (at least for Windows and Mac users) will require the other person to download and install extra software, and no one wants to do that. If you absolutely have to, you can use something else, but the harder you do it for other people (especially non-tech sexperienced people), the more they'll wonder if what you're sending them is worth. That said, if you're filing for your own purposes—say moving large files or groups of files between computers—you have a few options. There are many file formats to go through here, so let's list some of our favorites and their pros and cons. Zip: It won't compress your files in a ton, but it's super fast and you don't need to install any extra software. Great for Windows and Mac users. Tar.gz: Similar to zip, tar.gz files are very fast. It is the most available to Linux users, and also compresses files to a slightly smaller size than zip. Great for linux uses. Tar.bzip2: Another great option for Linux users; tar.bzip2 is a little slower than tar.gz, but will compress files to a smaller size.7z: The file file popular by our favorite compression tool, 7-Zip. It's pretty slow, but it compresses files like it's nobody's business. This is the best format to use if you need a very, very small file. 7-Zip is only available for Windows, but you can get a command-line tool for Mac OS X and Linux, too. 7z is one of the only compression formats that can split files into multiple files and put them back together, too. This may seem counterintuitive, but it's useful if you're trying to send too many large files about something like email, which limits the size of the files you send RAR: This format is comparable to 7z. 7z usually has better compression, with some notable exceptions like mp3s, in which case RAR compresses to smaller sizes. However, the only software that creates RAR files (WinRAR) is shareware and only Windows. You can unpack RAR files on Mac and Linux, but not create them. RAR can also split compressed files into multiple parts. So which one should you use? I personally use 7z unless I'm in a hurry because it compresses so well. I don't like RAR because of its price or nature only for Windows, and I'm mostly a mac and Linux user. As always, the best format is the one that best fits your need in any situation. If you are a Windows user and are compressing files to other Windows machines, RAR and 7z are good bets. Mac and Linux users will want to stick with tar.gz, tar.bz2, or 7z (although Mac users need a program like GUI Tar to create these formats unless they want to dive into the command line). Sincerely, Lifehackerp.S. This is a fairly basic summary, and a lot of it is opinion (and some people have some very strong opinions that may disagree with my own!). So if you have your own favorite file format, let us know what it is and why you love it in the comments. By Noah Kain When working on computers, backing up files and storing them can become a problem very quickly. If your computer is about to run out of hard drive space, you should back up and compress the files on your computer. This will ensure that your computer has enough free space to continue running without problems and also that you do not lose any files that you want to save. Compressing files for backup requires only a few easy steps. Find the files you want to compress and place them in a new folder. Before you start compressing your files, you'll need to put them in one place. Make a new folder on your desktop and drag the files you want to compress into the folder. Name your folder. When compressing files for backup, you need to stay as organized as possible. That's because in the future, when you want to unpack your backup files, you'll want to know the is in each compressed file. Make the folder name that you will compress something specific to the files you are putting in the folder. To compress the files in the folder, select the folder and right-click it. A menu should appear Screen. Click the option in the menu titled Compress folder. This will compress the folder into what is known as a zip file suitable for storing on a backup drive. In our previous example, we chose all repeated words and put it in a dictionary. For us, this is the most obvious way to write a dictionary. But a compression program sees it quite differently: it has no concept of separate words - it only looks for patterns. And in order to reduce the file size as much as possible, it carefully selects which patterns to include in the dictionary. If we approach the phrase from this perspective, we end up with a completely different dictionary. Publicity If the compression program scans Kennedy's phrase, the first redundancy i would find would be just a few letters. In do not ask what yours, there is a repeated pattern of the letter t followed by a space - in no and what. If the compression program wrote this for the dictionary, it could write a 1 every time a t was followed by a space. But in this short sentence, this pattern does not occur enough to make it a valuable entry, so the program would eventually replace it. The next thing the program may notice is either, that it appears both in yours and in the country. If this were a longer document, writing this pattern for the dictionary could save a lot of space - or it's a fairly common combination in the English language. But as the compression program worked through this sentence, he would quickly discover a better choice for an entry in the dictionary: Not only is it or repeated, but all his and country's words are repeated, and they are actually repeated together, like the phrase his country. In this case, the program would replace the dictionary entry to or with the entry to your country. The phrase can do by is also repeated once followed by yours and once followed by you, giving us a repeated pattern of can do for you. This allows us to type 15 characters (including spaces) with a numedisle value, while your country only allows us to type 13 characters (with spaces) with a numtile value, so that the program replaces the entry your country as just r country, and then write a separate entry to can do for you. The program proceeds in this way, picking up all the repeated bits of information and then calculating which patterns to write to the dictionary. This ability to rewrite the dictionary is the adaptive part of the lz adaptive dictionary-based algorithm. The way a program actually does this is quite complicated, as you can see from discussions about Data-Compression.com. No matter which specific method you use, this in-depth search system allows you to compress the file much more efficiently than you could just by choosing words. Using the patterns we chose above, and adding __ to spaces, we created this larger dictionary: And that's it phrase: 1not__2345__-___12354 The sentence now occupies 18 units of memory, and our dictionary occupies 41 units. So we compressed the total file size from 79 units to 59 units! This is just a way to compress the sentence, and not necessarily the most efficient. (See if you can find a better way!) So how good is this system? The reason for file reduction depends on a number of factors, including file type, file size, and compression scheme. In most languages of the world, certain letters and words often appear together in the same pattern. Because of this high redundancy rate, text files compress very well. A reduction of 50% or more is typical for a good-sized text file. Most programming languages are also very redundant because they use a relatively small collection of commands, which often come together in a defined pattern. Files that include many unique information, such as graphics or MP3 files, cannot be very compressed with this system because they do not repeat many patterns (more on this in the next section). If a file has many repeated patterns, the reduction rate typically increases with the file size. You can see this just by looking at our example - if we had more of Kennedy's speech, we could refer to the patterns in our dictionary more often, and thus get more of the file space from each entry. In addition, more widespread patterns can arise in longer work, allowing us to create a more efficient dictionary. This efficiency also depends on the specific algorithm used by the compression program. Some programs are particularly suitable for capturing patterns in certain types of files, and so they can compress them more succinctly. Others have dictionaries within dictionaries, which can compress efficiently for larger files, but not for smaller ones. Although all compression programs of this type work with the same basic idea, there is actually a good deal of variation in the way of execution. Programmers are always trying to build a better system. System.