Compress

h1. Component Overview Compress is an API for working with the various archiving and compression formats. h1. Quick Start *compress* is a stream based API. Some of it's original code came from Avalon's Excalibur, but originally from Ant, as far as life in Apache goes. It has migrated via: Ant \-> Avalon-Excalibur \-> Commons-IO \-> Commons-Compress. More credits can be found in NOTICE.txt file. *compress* divides the implementation in Compressors and Archivers. For each one an factory is implemented. Basically you have to get the stream implementation from the factory, create an entry, put this into the stream and stream. Some experimental code makes it possible to modify archiver and compressor files. This means you can delete from for example a zip file. Historical notes: There were discussion of "sponsoring" compress with code from [TrueZip] (https://truezip.dev.java.net/). The result was, not to include any code from this project (see mailinglists). There was a discussion about the complex file based implementation too. The result was to start again with a stream based implementation and move the old one to a branch. More information about the old implementation can be found on the [CompressImplementationDetails] page. h1. Roadmap We're currenlty discussing a 2.0 API that fully embraces generics and tries to address a few pain points discovered over the years Compress 1.x has been around. If you want to help, here is the [CompressRoadmap]. h2. Archiver To pack an archive, you have to get an archiver stream via the [ArchiverFactory]. Add your streams to the archiver and stream. Archiver streams are: ZIP, CPIO, AR, TAR, JAR, DUMP. 7z, ARJ. h3. Creating a ZIP-File {{{final [OutputStream] out = new [FileOutputStream](output); [ArchiveOutputStream] os = new [ArchiveStreamFactory]().createArchiveOutputStream("zip", out); os.putArchiveEntry(new [ZipArchiveEntry]("testdata/test1.xml")); IOUtils.copy(new [FileInputStream](file1), os); os.closeArchiveEntry(); os.putArchiveEntry(new [ZipArchiveEntry]("testdata/test2.xml")); IOUtils.copy(new [FileInputStream](file2), os); os.closeArchiveEntry(); out.finish(); os.close();}}} h3. Unpacking a ZIP-File {{{final [InputStream] is = new [FileInputStream](input); [ArchiveInputStream] in = new [ArchiveStreamFactory]().createArchiveInputStream("zip", is); [ZipArchiveEntry] entry = (ZipArchiveEntry)in.getNextEntry(); [OutputStream] out = new [FileOutputStream](new File(dir, entry.getName())); IOUtils.copy(in, out); out.close(); in.close();}}} h2. Compressor Same goes for Compressor. Compressor streams are: bz2, gz, lzma, xz, pack200, Z. h3. Compressing a file {{{final [OutputStream] out = new [BufferedOutputStream](new [FileOutputStream](output)); [CompressorOutputStream] cos = new [CompressorStreamFactory]().createCompressorOutputStream("bzip2", out); IOUtils.copy(new [FileInputStream](input), cos); cos.close();}}} h3. Decompressing a file {{{final [InputStream] is = new [BufferedInputStream](new [FileInputStream](input)); [CompressorInputStream] in = new [CompressorStreamFactory]().createCompressorInputStream("bzip2", is); IOUtils.copy(in, new [FileOutputStream](output)); in.close();}}} h1. FAQ |Add your questions/answers here.|