# Siard Suite Zip64 tips & tricks

How to use Zip64 to manual unzip and zip extractions
https://github.com/sfa-siard/Zip64File

### Create a SIARD file (always creates Zip64-format)

```
java -jar "C:\prog\siardsuite\zip64-2.1.80\zip64\lib\zip64.jar" n "-d=D:\_demo-temp\002\in"
"D:\_demo-temp\002\out\sakila_mysql5.6.35_scfc20.03.1583_int_zip64.2.1.80.siard"
```

### Unpack a SIARD file (can upack both Zip32- and Zip64-format)

```
java -jar "C:\prog\siardsuite\zip64-2.1.80\zip64\lib\zip64.jar" x "-d=D:\_demo-temp\001\out"
"D:\_demo-temp\001\in\sakila_mysql5.6.35_scfc20.03.1583_int.siard"
```

### Result unpack SIARD-file

```
Microsoft Windows [Version 10.0.18363.720]
(c) 2019 Microsoft Corporation. Med enerett.

C:\WINDOWS\system32>java -jar "C:\prog\siardsuite\zip64-2.1.80\zip64\lib\zip64.jar" x "-
d=D:\_demo-temp\001\out" "D:\_demo-temp\001\in\sakila_mysql5.6.35_scfc20.03.1583_int.siard"
zip64 2.1 - handles ZIP64 archives
Copyright (c) 2010, Hartwig Thomas, Enter AG, Zurich, Switzerland
This program comes with ABSOLUTELY NO WARRANTY.
This is open source software, and you are welcome to redistribute it.
See https://opensource.org/licenses/CDDL-1.0 for details.

Extracting from D:\_demo-temp\001\in\sakila_mysql5.6.35_scfc20.03.1583_int.siard ...
extracting D:\_demo-temp\001\out\content\schema0\table0\table0.xml from
content/schema0/table0/table0.xml
extracting D:\_demo-temp\001\out\content\schema0\table0\table0.xsd from
content/schema0/table0/table0.xsd
extracting D:\_demo-temp\001\out\content\schema0\table1\table1.xml from
content/schema0/table1/table1.xml
extracting D:\_demo-temp\001\out\content\schema0\table1\table1.xsd from
content/schema0/table1/table1.xsd
extracting D:\_demo-temp\001\out\content\schema0\table2\table2.xml from
content/schema0/table2/table2.xml
extracting D:\_demo-temp\001\out\content\schema0\table2\table2.xsd from
content/schema0/table2/table2.xsd
extracting D:\_demo-temp\001\out\content\schema0\table3\table3.xml from
content/schema0/table3/table3.xml
extracting D:\_demo-temp\001\out\content\schema0\table3\table3.xsd from
content/schema0/table3/table3.xsd
extracting D:\_demo-temp\001\out\content\schema0\table4\table4.xml from
content/schema0/table4/table4.xml
extracting D:\_demo-temp\001\out\content\schema0\table4\table4.xsd from
content/schema0/table4/table4.xsd
```

```
extracting D:\_demo-temp\001\out\content\schema0\table5\table5.xml from
content/schema0/table5/table5.xml
extracting D:\_demo-temp\001\out\content\schema0\table5\table5.xsd from
content/schema0/table5/table5.xsd
extracting D:\_demo-temp\001\out\content\schema0\table6\table6.xml from
content/schema0/table6/table6.xml
extracting D:\_demo-temp\001\out\content\schema0\table6\table6.xsd from
content/schema0/table6/table6.xsd
extracting D:\_demo-temp\001\out\content\schema0\table7\table7.xml from
content/schema0/table7/table7.xml
extracting D:\_demo-temp\001\out\content\schema0\table7\table7.xsd from
content/schema0/table7/table7.xsd
extracting D:\_demo-temp\001\out\content\schema0\table8\table8.xml from
content/schema0/table8/table8.xml
extracting D:\_demo-temp\001\out\content\schema0\table8\table8.xsd from
content/schema0/table8/table8.xsd
extracting D:\_demo-temp\001\out\content\schema0\table9\table9.xml from
content/schema0/table9/table9.xml
extracting D:\_demo-temp\001\out\content\schema0\table9\table9.xsd from
content/schema0/table9/table9.xsd
extracting D:\_demo-temp\001\out\content\schema0\table10\table10.xml from
content/schema0/table10/table10.xml
extracting D:\_demo-temp\001\out\content\schema0\table10\lob5\seg0\rec1.bin from
content/schema0/table10/lob5/seg0/rec1.bin
extracting D:\_demo-temp\001\out\content\schema0\table10\table10.xsd from
content/schema0/table10/table10.xsd
extracting D:\_demo-temp\001\out\content\schema0\table11\table11.xml from
content/schema0/table11/table11.xml
extracting D:\_demo-temp\001\out\content\schema0\table11\table11.xsd from
content/schema0/table11/table11.xsd
extracting D:\_demo-temp\001\out\content\schema0\table12\table12.xml from
content/schema0/table12/table12.xml
extracting D:\_demo-temp\001\out\content\schema0\table12\table12.xsd from
content/schema0/table12/table12.xsd
extracting D:\_demo-temp\001\out\content\schema0\table13\table13.xml from
content/schema0/table13/table13.xml
extracting D:\_demo-temp\001\out\content\schema0\table13\table13.xsd from
content/schema0/table13/table13.xsd
extracting D:\_demo-temp\001\out\content\schema0\table14\table14.xml from
content/schema0/table14/table14.xml
extracting D:\_demo-temp\001\out\content\schema0\table14\table14.xsd from
content/schema0/table14/table14.xsd
extracting D:\_demo-temp\001\out\content\schema0\table15\table15.xml from
content/schema0/table15/table15.xml
extracting D:\_demo-temp\001\out\content\schema0\table15\table15.xsd from
content/schema0/table15/table15.xsd
extracting D:\_demo-temp\001\out\header\metadata.xml from header/metadata.xml
extracting D:\_demo-temp\001\out\header\metadata.xsd from header/metadata.xsd
36 matching file entries extracted.
```

**Result create SIARD-file**

```
C:\WINDOWS\system32>java -jar "C:\prog\siardsuite\zip64-2.1.80\zip64\lib\zip64.jar" n "-
d=D:\_demo-temp\002\in" "D:\_demo-
temp\002\out\sakila_mysql5.6.35_scfc20.03.1583_int_zip64.2.1.80.siard"
zip64 2.1 - handles ZIP64 archives
Copyright (c) 2010, Hartwig Thomas, Enter AG, Zurich, Switzerland
This program comes with ABSOLUTELY NO WARRANTY.
This is open source software, and you are welcome to redistribute it.
See https://opensource.org/licenses/CDDL-1.0 for details.

Injecting into D:\_demo-temp\002\out\sakila_mysql5.6.35_scfc20.03.1583_int_zip64.2.1.80.siard ...
inserting content/
inserting content/schema0/
inserting content/schema0/table0/
inserting content/schema0/table0/table0.xml
inserting content/schema0/table0/table0.xsd
inserting content/schema0/table1/
inserting content/schema0/table1/table1.xml
inserting content/schema0/table1/table1.xsd
inserting content/schema0/table10/
inserting content/schema0/table10/lob5/
inserting content/schema0/table10/lob5/seg0/
inserting content/schema0/table10/lob5/seg0/rec1.bin
inserting content/schema0/table10/table10.xml
inserting content/schema0/table10/table10.xsd
inserting content/schema0/table11/
inserting content/schema0/table11/table11.xml
inserting content/schema0/table11/table11.xsd
inserting content/schema0/table12/
inserting content/schema0/table12/table12.xml
inserting content/schema0/table12/table12.xsd
inserting content/schema0/table13/
inserting content/schema0/table13/table13.xml
inserting content/schema0/table13/table13.xsd
inserting content/schema0/table14/
inserting content/schema0/table14/table14.xml
inserting content/schema0/table14/table14.xsd
inserting content/schema0/table15/
inserting content/schema0/table15/table15.xml
inserting content/schema0/table15/table15.xsd
inserting content/schema0/table2/
inserting content/schema0/table2/table2.xml
inserting content/schema0/table2/table2.xsd
inserting content/schema0/table3/
inserting content/schema0/table3/table3.xml
inserting content/schema0/table3/table3.xsd
inserting content/schema0/table4/
inserting content/schema0/table4/table4.xml
inserting content/schema0/table4/table4.xsd
inserting content/schema0/table5/
inserting content/schema0/table5/table5.xml
inserting content/schema0/table5/table5.xsd
inserting content/schema0/table6/
inserting content/schema0/table6/table6.xml
inserting content/schema0/table6/table6.xsd
inserting content/schema0/table7/
inserting content/schema0/table7/table7.xml
inserting content/schema0/table7/table7.xsd
inserting content/schema0/table8/
inserting content/schema0/table8/table8.xml
inserting content/schema0/table8/table8.xsd
inserting content/schema0/table9/
inserting content/schema0/table9/table9.xml
inserting content/schema0/table9/table9.xsd
inserting header/
inserting header/metadata.xml
inserting header/metadata.xsd
inserting header/siardversion/
inserting header/siardversion/2.1/
58 matching file entries injected.
```

**Zip64 v2.1.80**

# Usage

Zip64File implements a command-line *zip64* as well as an API for programmers. The use of the API is documented in the developer's manual and in the javadoc documentation. Here we describe the use of the command-line *zip64*.

*zip64* is started by running *zip64.cmd* (for Windows) or *zip64.sh* (for LINUX) or explicitly starting it using the appropriate JAVA executable.

```
zip64.cmd <arguments>,
zip64.sh <arguments>, or
java -jar <zip64path>/lib/zip64.jar <arguments>
```

where *<zip64path>* refers to the installation folder of *Zip64File* and *<arguments>* refers to the following arguments that can be specified on the command-line:

```
<command> options <zipfile> files...
```

## Command

The command defines the basic operation executed by *zip64*. It must be one of the following:

**n**

for inject overwrites and existing archive or creates a new archive, adding the files,

**x**

for extract extracts the files from the archive, or

**l**

for list lists the files in the archive.

If the first argument is not one of these commands, then **l** is assumed as a default.

## Options

If *zip64* is called with the special option **-h** a short explanation of the usage is displayed and all other arguments are ignored.

Otherwise the following options can be used:

| **option** | **cmd** | **meaning** |
|---|---|---|
| **-d:<ziproot>** | n, x | external directory relative to which the files will be injected or extracted<br>default: current folder |
| **-c** | n | if present, content will be compressed<br>default: uncompressed storage (ZIP64 file is just used as a container) |
| **-r** | n, x | if present, the target files will be replaced/overwritten without warning<br>default: processing skips existing file entries and lists them on *stderr*. |
| **-q** | n, x | quiet, do not display output on *stdout*<br>default: listing on *stdout* |
| **-z:<comment>** | n | adds a comment to the ZIP file |

## zipfile

Archive file in ZIP format. On input this can be either in ZIP32 or in ZIP64 format. The *zip64* utility will only write archives in the ZIP64 format, even if the content could be stored in ZIP32 format.

## files ...

One or more names of files or @ followed by the name of a plain text file containing the names of files to be injected or extracted. (The text file will be read with the JAVA default character set which can be changed by specifying the JVM parameter file.encoding on the command line.)
Each filename (on the command line or in the file list) can contain the wildcard characters ?, * or **.
The ? indicates a single character at this position in the name. The * indicates any number of characters in a file or folder name. It occurs at most once in a file or folder name. ** indicates any number of characters including the folder separator. It occurs at most once in the full file name.
If the *file...* argument is missing, ** is assumed as a default.
N.B.: Unfortunately the command line is parsed in a half-hearted way by the shell or by JAVA. Therefore you cannot enter a file template starting with „**“. You have to avoid the situation by starting the file name template with another character or use the file name list.
Also you need to be careful not to start a file name with „/“ on the Windows platform. On this platform „/“ is treated as a variant of „-“ and introduces command-line options.

# Examples

```
java -jar "C:\Programme\Enter AG\zip64-1.0\lib\zip64.jar" n "-d=C:\Dokumente und
Einstellungen\Hartwig\Eigene Dateien\SourceFolder" "C:\Dokumente und
Einstellungen\Hartwig\Eigene Dateien\Test.zip"
```

recursively packs all files in *SourceFolder* into *test.zip* retaining the path name relative to the source folder in the ZIP file. If *test.zip* already existed, the new content is added, unless its name collides with an existing file entry.

```
java -jar "C:\Programme\Enter AG\zip64-1.0\lib\zip64.jar" x "-d=C:\Dokumente und
Einstellungen\Hartwig\Eigene Dateien\TargetFolder" "C:\Dokumente und
Einstellungen\Hartwig\Eigene Dateien\Test.zip"
```

recursively extracts all content of *test.zip* into the *TargetFolder*. Existing files in the target folder are not overwritten.
If the two examples are executed one after the other where initially the file *test.zip* did not exist and the target folder was empty, then the target folder ends up with the same content as the source folder.