Overview

Opening Files

When working with the SRP Zip Utility, you must first get a handle to a zip file. SRP_Zip_Open gets a handle to an existing file, or creates it if it doesn't exist. SRP_Zip_Create gets a handle to a new empty zip file. The handles are then used by the other SRP Zip Utility functions for manipulation.

You must always be sure to close a file handle using SRP_Zip_Close. This ensures that the resources are released. Forgetting to perform this step can result in resource leaks and, eventually, application hang ups. Additionally, you must always close the handle to one zip file before creating or opening another one.

Once you've retrieved a zip file handle, you are ready to access and/or modify files by index.

File Indexes

Eventually you will access a file within a zip file, and you will need an index to refer to it. The SRP Zip Utility comes with two methods for determining file indexes. SRP_Zip_GetFileList returns an ordered list of all files within the zip file. The positions of the files within the returned list are the same as the positions within the zip file. So, you can use the position of the file as an accurate index. If you already know the file name and just need its index, then use SRP_Zip_FindFile. This method returns the index of the file if found. Once the index is obtained, you're ready to access it.

File Processing

SRP_Zip_ExtractFile is the method you will use to extract files out of a zip file. The file's contents are returned directly into a variable. To remove a file from a zip file, simply pass the file index to SRP_Zip_RemoveFile. SRP_Zip_AddFile adds files to a zip file. Similar to file extraction, the new file's contents are provided via a BASIC+ variable. All three methods can be used with confidence in security.

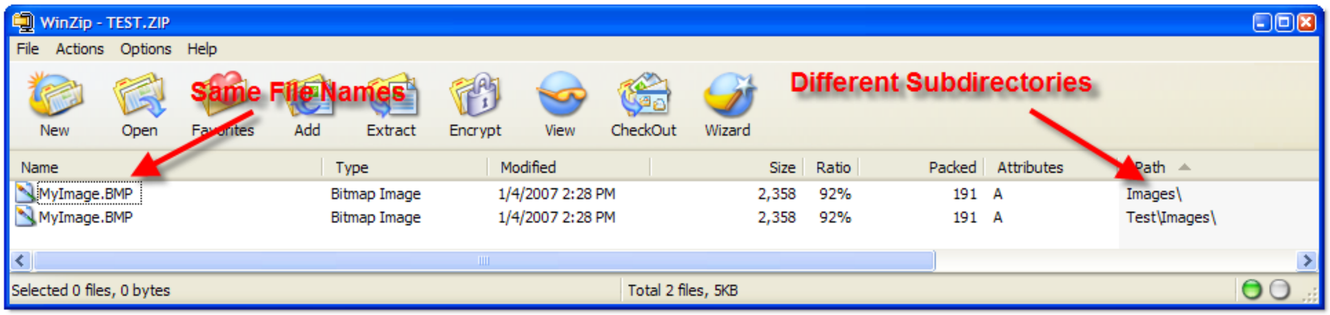
Security

The SRP Zip Utility provides added security in two ways. First, files are extracted into variables or added from variables. The data never has to reside temporarily on the hard drive, which is often a requirement when using command line utilities. This also adds a level of convenience. Once the data is in a variable, you can use an BASIC+ routine to store or process it as needed.

A second level of security comes in the form of a Password parameter. The optional Password parameter is available in SRP_Zip_ExtractFile, SRP_Zip_RemoveFile, and SRP_Zip_AddFile. Zip files store an individual password for each file it contains, although the same password is usually used for all files. If you plan to extract or remove a protected file, you **must** provide the password. When adding files, you may optionally provide a password to secure it.

Subdirectories

Files within a zip file may include directory information, which we refer to as subdirectory associations. It is important to note that, when referencing a file by name, you **must always** include it's subdirectory information. Essentially, subdirectory associations allow two files of the same name to exist within the same zip file, as seen in the following image.



Name	Type	Modified	Size	Ratio	Packed	Attributes	Path
MyImage.BMP	Bitmap Image	1/4/2007 2:28 PM	2,358	92%	191	A	Images\
MyImage.BMP	Bitmap Image	1/4/2007 2:28 PM	2,358	92%	191	A	Test\Images\

Selected 0 files, 0 bytes

Total 2 files, 5KB

If you call SRP_Zip_GetFileList for the above file, you will get the following list:

```
<1>     \Test\Images\MyImage.BMP
<2>     \Images\MyImage.BMP
```

Note that the two files use the same name, but they are stored as two separate files. This is due to the difference in their subdirectory associations. As an example, if you add "MyImage.BMP" to the zip file, it will be created as a third file since there is no subdirectory association. On the other hand, if you add "\Images\MyImage.BMP" to the zip file, then the current file with that subdirectory association will be replaced.

Zip files can store a subdirectory with no file attached to it. For instance, "\Images\" may appear in the file list. This is not a bug with the SRP Zip Utility. This is just a feature of zip files in general. In fact, you can add a subdirectory with no data associated with it using SRP_Zip_AddFile, though there is usually no reason to do so. Most often, you will encounter stand alone directories in zip files generated by third party utilities or command line driven zip programs.

Any attempt to extract a subdirectory using SRP_Zip_ExtractFile results in an empty variable.

Error Handling

All routines in the SRP Zip Utility return a value. Sometimes, the value will indicate directly that an error has occurred. Usually, a zero is returned. In other cases, an empty value is return (""), which means an error returns or there was no data to retrieve. Refer to the reference documentation for any particular function's mode of operation. In all cases, use SRP_Zip_GetLastError to get a readable error message that can be displayed to the user.

Examples

```
// extract all files and save to a table (assuming table was previously opened)
hZip = SRP_Zip_Open("C:\MyFile.zip")
If hZip EQ 0 then
   Call Msg(@Window, SRP_Zip_GetLastError())
end else
   FileList = SRP_Zip_GetFileList(hZip)
   NumFiles = Count(FileList, @FM) + (FileList NE "")
   For i = 1 to NumFiles
       Data = SRP_Zip_ExtractFile(hZip, i)
       Write Data to hTable, FileList<i> then NULL
   Next i
   SRP_Zip_Close(hZip)
end

 // extract and save a single known file to a table (assuming table was previously opened)
hZip = SRP_Zip_Open("C:\MyFile.zip")
If hZip EQ 0 then
   Call Msg(@Window, SRP_Zip_GetLastError())
end else
   FileIndex = SRP_Zip_FindFile(hZip, "MyImage.BMP")
   If FileIndex NE 0 then
       Data = SRP_Zip_ExtractFile(hZip, FileIndex)
       Write Data to hTable, "MyImage.BMP" then NULL
   end
   SRP_Zip_Close(hZip)
end

 // remove a single known file
hZip = SRP_Zip_Open("C:\MyFile.zip")
If hZip EQ 0 then
   Call Msg(@Window, SRP_Zip_GetLastError())
end else
   FileIndex = SRP_Zip_FindFile(hZip, "MyImage.BMP")
   If FileIndex NE 0 then
       Data = SRP_Zip_RemoveFile(hZip, FileIndex)
   end
   SRP_Zip_Close(hZip)
end

 // load an OS file into a variable and add it to a new zip file
hZip = SRP_Zip_Create("C:\MyFile.zip")
If hZip EQ 0 then
   Call Msg(@Window, SRP_Zip_GetLastError())
end else

   // notice how we can add the same data as 3 files with different subdirectories
   OSRead Data from "C:\sample.bmp" then
       rv = SRP_Zip_AddFile(hZip, "Test\Images\MyImage.BMP", Data)
       rv = SRP_Zip_AddFile(hZip, "Images\MyImage.BMP", Data)
       rv = SRP_Zip_AddFile(hZip, "MyImage.BMP", Data)
   end

   SRP_Zip_Close(hZip)
end

 // add a file whose contents come directly from a table
Open "MY_TABLE" to hTable then
   Read Data from hTable, "TEST_DATA" then
       hZip = SRP_Zip_Create("C:\MyFile.zip")
       If hZip EQ 0 then
           Call Msg(@Window, SRP_Zip_GetLastError())
       end else
           rv = SRP_Zip_AddFile(hZip, "Test.dat", Data)
           SRP_Zip_Close(hZip)
       end
   end
end
```