Contents

Overview

Introduction
Registration

Installation

Requirements
Installing the ZipServer

How to

Select an archive
Add files to an archive
Update an archive
Delete files from an archive
Verify files in an archive
Extract files from an archive
Create a Multi Volume archive
Extract a Multi Volume archive
Recurse Subdirectories
Store Path Names
Restore Directories
Keep Existing Zip Date
Overwrite existing files
Freshen existing files
Extracting Newer files
Protect archive with Password
List File
Associate menu command
View Zip
Commenting an archive
Launch on double-click
Stay on Top

Calling ZipServer from another Application

Using Dynamic Data Exchange (DDE)
Using the ZipServer dynamic-link library

RMV Archive Series

About RMV Archive Series
Limitations of RMV archive series
Working with RMV Archive series
Compatibility

Introduction

ZipServer is a tool for creating, updating, viewing and extracting PKZIP 2.04g compatible archives by loading ZipServer as a visible Windows application or using it remotely and invisibly from another Windows application, or using ZIPSERV.DLL API **Zip** and **Unzip** calls from a Visual Basic program. ZipServer is a Visual Basic program using Compression Plus. It is seamless bound to the client application when used remotely. The ZipServer installation package includes sample source code for Visual Basic 3.0 and for MsWord 6.0 that show how to access the ZipServer. The sample codes are also helpful for users who want to link ZipServer to MSExcel or MSAccess. The sample Visual Basic program presents the use of the Zip, ZZip and Unzip as well as the MultiZip, MultiUnzip API calls.

ZipServer can produce and extract Multi Volume archives and be used as a tool for backup. Projects can be archived based on the project file including associated files that are actually not listed in the project file.

Registration

Before we begin discussing ZipServer, please take a moment to fill out the registration card. Doing this entitles you to free technical support by phone, as well as ensuring that you are notified of possible enhancements and new products.

The Registration Card can be completed by selecting the Register Command, and can either be printed and mailed, or faxed.You can register via Compuserve (GO SWREG - SHAREWARE REGISTRATION - SEARCH BY REGISTRATION ID 5205), but in order to send you your registration code you must EMail the serial number of your installed ZipServer.

After registration you will receive a code. Click the Register menu command on the Registration Form to enter the code.

You are entitled to experience ZipServer without registration and payment. However, you must register in order to use the software. The free trial will end after the tenth usage.

You can operate in one computer one registered copy of the ZipServer Single User Edition, or in one server one registered copy of the ZipServer Network Edition. ZipServer can not be distributed as a part of an application. However, you can distribute the installation package of ZipServer, which your user can register.

The **ZipServer Development Kit** is available for programmers. The kit provides the exact same functionality that is described in the *Using the ZipServer dynamic-link library* section of this manual, but allows royalty free distribution of parts of the ZipServer.

Requirements

Hardware
Operation System

Hardware

- Min. 486 processor
- Min. 4MB RAM
- Min. 2MB free space on the hard drive
- Min. half space of a floppy disk must be free when the archive resides there

Operation System

Microsoft ®Windows™ 3.1 or later

Novell NetWare all versions up to 4.1 (Network Edition only)

Installing the ZipServer

General
Special requirements for Network installation
Installing the Single User Edition on Network
Running Setup
Installing the ZipServer Development Kit

General

To install the ZipServer you must use the Setup program, SETUP.EXE on the distribution disk or in the directory to where ZipServer was downloaded from Compuserve, America Online, or a BBS.

Special requirements for Network installation

The installation of the ZipServer Network Edition normally is performed by the Network Supervisor or Administrator. Before you start to install the Network Edition you need to prepare the environment for the ZipServer. This process includes the following steps:

1. Assign a network location to where the ZipServer will be installed and map it. The location must provide read/write access for all users.
2. Assign another network location to where the ZipServer library files will be installed and map it. This location must be in the Network path for all users. In addition to that you also need to add an environ variable to the current table pointing to the library path. Users will also need Read/Write access to this location. The line you need to add to the Environ table should look like:

   ZipServPath=H:\DIRECTORY\SUBDIR

   if the mapping of the H drive will be identical for all users.
3. Check the Environ table with the SET DOS command before you start to install the ZipServer.

Note:

The installation package is the same for the Single User and for the Network Edition. If the Network is not available on the computer you use or the ZipServPath environ variable does not exist or does not point to a valid Network location with write access, the Single Edition will be installed. When the Network is not Novell you will need to consult with Redei Enterprises before the installation.

Installing the Single User Edition on Network

Usually no special action is required when the ZipServer Single User Edition is installed on a station of a Network. However, when the \WINDOWS\SYSTEM directory is installed on a Network drive, you must assign the **ZipServPath** environment variable as it was explained in the prior section, and only the Network Edition can be installed.

Running Setup

1. Place the Installation Diskette in the floppy disk drive.
2. From the Program Manager or from the File Manager click **File** and choose **Run**.
3. When the dialog box appears, type as shown below:

   A:SETUP

   (Assuming that, drive A: is used. If drive B: is used type B: instead of A:.)
4. Click **OK**. Several seconds later you will be asked where you want to install ZipServer to. Select the default directory or any other by your choice and press **Enter**.
5. You will be notified when **Setup** finished the installation.
6. Setup will start ZipServer. If you are performing a Network installation ZipServer will check the circumstances and might displays messages when unable to proceed. At the end of the installation procedure the About ZipServer screen will appear. Press the OK button. An Open Zip File dialog box will appear. You can either select an existing. ZIP file, or enter a new one.

Installing the ZipServer Development Kit

1. Install the ZipServer as described above
2. Upon registration you will receive additional files and instructions.

Select an archive

When you first start ZipServer, you will be asked to select the name of a ZIP archive. When you have finished with that archive, you may pick another by clicking on the **Zip File Name** label or by selecting **Open Zip File** from the File menu.

Add files to an archive

The **Add** button is used to add the files in the Source Files **on Disk** list box to the current Zip File.

In order to add files to an archive, you must be in **Zip mode** (choose the Zip option). A list box of **Source Files on Disk** will be displayed in the right-hand corner. You can add files to this list box by clicking on the **Browse** button, which allows you to select one or more files at a time from a given path. You may remove individual entries from the list by clicking on **Untag**, or clear the entire list with **Untag All**. There is a text box over the list box. You can edit in the text box the Source Files, including wild card characters. The content of the list box will be added to the text box. When a large number of files, including subdirectories need to be archived the text box offers a faster method. For example, you might prefer to write in the text box ***C:\VB*.**** and mark the Recursive and Store Path check boxes to archive your whole Visual Basic than add all files to the list box.

Click on the **Add** button by the **Files in Zip** list box, to the left when you are satisfied with your selections.

Update an archive

The **Freshen** and **Update** buttons are used to update an archive according to its current contents and the files in the Source Files **on Disk** list box. The **Freshen** button causes the files in the **Source Files on Disk** list box to replace any corresponding files in the **Files in Zip** list box, if the files on disk are newer than the files in the archive. The **Update** button causes the files in the **Source Files on Disk** list box to replace any corresponding files in the **Files in Zip** list box, if the files on disk are newer than the files in the archive, or if the files on disk do not exist in the archive.

In order to update an archive, you must be in **Zip mode** (choose the Zip option). A list box of **Source Files on Disk** will be displayed in the right-hand corner. You can add files to this box by clicking on the **Browse** button, which allows you to select one or more files at a time from a given path. You may remove individual entries from the list by clicking on **Untag**, or clear the entire list with **Untag All**. There is a text box over the list box. You can edit in the text box the Source Files, including wild card characters. The list box and the text box can be used together. When a large number of files, including subdirectories need to be archived the text box offers an easier and faster method. For example, you might prefer to write in the text box ***C:\VB*.**** and mark the Recursive and Store Path check boxes to archive your whole Visual Basic than add all files to the list box.

Click on the **Freshen** or **Update** buttons by the **Files in Zip** list box, to the left when you are satisfied with your selections.

Delete files from an archive

The **Delete** button is used to delete files tagged in the **Files in Zip** list box from the current archive.

In order to delete from the archive, you must be in **Zip mode** (choose the Zip option). Tag the files that you want to delete in the **Files in the Zip** list box. When you are satisfied with your selections, click on the **Delete** button.

Verify files in an archive

The **Verify** button is used to verify the good condition of files tagged in the **Files in Zip** list box.

You may verify files regardless of whether Zip or Unzip mode is chosen. Tag the files you want to verify in the Files in Zip list box. When you are satisfied with your selections, click on the Verify button.

Extract files from an archive

The **Extract** button is used to extract files tagged in the **Files in Zip** list box from the current archive. The extracted files are placed in the directory specified in the Destination Path area.

In order to extract from an archive, you must be in **Unzip mode** (choose the Unzip option). Tag the files that you want to extract in the Files in Zip list box. Select the destination drive and directory in the Destination Path area to the right. When you are satisfied with your selections, click on the Extract button.

Use the **Tag All** and **Untag All** buttons for speedier selection.

Create a Multi Volume archive

Place an empty formatted diskette in the floppy disk drive you wish to use. Select an archive on your floppy disk drive. Choose the **Multi Volume** check box. Select the type of diskette to be used in the **Diskette** option box. Select **Special** from the option box if the size of the diskette is not listed. ZipServer will promt you to enter the size of the diskdrive in Megabytes you want to use. The **Add** button is used to add the files in the Source Files **on Disk** list box or in the text box above it to the current Zip File.

In order to add files to an archive, you must be in **Zip mode** (choose the Zip option). A list box of Source Files on Disk will be displayed in the right-hand corner. You can add files to this list box by clicking on the **Browse** button, which allows you to select one or more files at a time from a given path. You may remove individual entries from the list by clicking on **Untag**, or clear the entire list with **Untag All**. Click on the **Add** button by the **Files in Zip** list box, to the left when you are satisfied with your selections.

Follow the instructions while creating a multi volume archive is in progress.

ZipServer can backup huge files with the multi volume method. The size of the archived file can exceed the capacity of the diskette.

Extract a Multi Volume archive

Place the first archived diskette in the floppy disk drive you wish to use. Select the archive on your floppy disk drive. Choose the **Multi Volume** check box (If the first diskette was placed in the diskdrive ZipServer should automatically detect that the archive is a multivolume backup and swich to Unzip mode). The **Extract** button is used to extract files tagged in the **Files in Zip** list box from the current archive. The extracted files are placed in the directory specified in the Destination Path area.

In order to extract from an archive, you must be in **Unzip mode** (choose the Unzip option). Multi Volume archives are NOT update able. Select the destination drive and directory in the Destination Path area to the right. When you are satisfied with your selections, click on the Extract button.

Follow the instructions while extracting a multi volume archive is in progress.

Recurse Subdirectories

It is available only in Zip mode. When this option is specified, ZipServer will search the source directory for subdirectories. If any subdirectories are found, ZipServer searches them for files. If further level of subdirectories is found in a subdirectory, ZipServer will search these as well. ZipServer will enter as many subdirectory levels as exist. In order to preserve the path structure use it together with the Store Path Names option.

Store Path Names

It is available only in Zip mode. When the **Store Path Names** check box is checked the path will be stored in the archive and can be restored at extraction.

Restore Directories

Available only in Unzip mode. When the **Restore Directories** check box is checked and the archive was created with the Store Path Names option, the path will be restored in the extraction process. Non existing directories will be created.

Keep Existing Zip Date

It is available only in Zip mode. When the **Keep Existing Zip Date** check box is checked the original date of the archive file will be kept while updating.

Overwrite existing files

It is available only in Unzip mode. When the **Overwrite** check box is checked ZipServer will overwrite existing files in the extraction process without asking it. If this option is unchecked ZipServer will ask every time if an existing file is to be overwritten unless one of the Freshen or the Newer check boxes is checked. Only one of the Freshen, Overwrite and Newer check boxes can be checked at a time.

Freshen existing files

It is available only in Unzip mode. When the **Freshen** check box is checked ZipServer will overwrite existing files if the archived version is newer. The Freshen option will not extract files that are non-existing. Only one of the Freshen, Overwrite and Newer check boxes can be checked at a time.

Extracting Newer files

It is available only in Unzip mode. When the **Newer** check box is checked ZipServer will overwrite existing files if the archived version is newer and in addition to that it will also extract those files that are non-existing. Only one of the Freshen, Overwrite and Newer check boxes can be checked at a time.

Protect archive with Password

When the Password check box is checked, the text box on the right side allows you to enter a Password. In Unzip mode those files updated or added with password can be extracted only with entering the same password prior extraction.

List File

It is available only in Zip mode. When the List File checkbox is checked ZipServer will archive the content of the file named in the Source Files **on Disk** list box additional to the file named. The list file must be the only listed source file and must be a text file where all files are listed in separate lines. ZipServer will verify the content of the list file and return a message if an error found. ZipServer will disregard any text in the file that can not be considered as a file name.

Use the List File option to archive file packages, such as a .MAK file for Visual Basic. Project files often do not list included files. For example, Visual Basic .MAK files do not list files with .FRX extensions, although they are part of the project. They are automatically included by Visual Basic if the leftside name of an FRX file matches with the leftside name of an .FRM file. ZipServer offers an Associate menu command with which users can instruct ZipServer to do the same. Do not forget the use of **Recurse** and **Store Path names**!

Associate menu command

Use the Associate menu command to add or delete file extension associations for the List File option.

The Associate screen is pretty much self explanatory. ZipServer will verify your entries and except them if they follow these rules:

- The total length of an entry must be minimum three, maximum seven characters
- The entry must have three zones. The middle zone is an equal '=' sign, the length of the left and right zones must be minimum one, maximum three characters
- Both the left and the right zones can consist of numbers (0 to 9) and the lower and upper case characters of the English alphabet, the underscore '_' and the dash '-'.

Example:

Write the line

FRM=FRX

to associate in a Visual Basic project file the files with FRM extension with the similarly named unlisted FRX files (for example, if there is a REDEI.FRM file in the .MAK file and in the same directory where REDEI.FRM resides there is a REDEI.FRX file as well, both will be archived).

View Zip

You can view the content of the selected archive by clicking on the **View Zip** button. Click the Control Box at the left upper corner of the View Window and select Close to unload the View Zip Window.

Commenting an archive

Once the archive file exists click on the **Zip Comment** box and type the comment you want. It will automatically be saved when you leave the editing field.

The same Comment box will display any existing comments on the current Zip file.

Launch on double-click

When the **Options - Lauch on double-click** menu item is checked a **Save Launched** checkbox will appear. The checkbox will be disabled (grayed). Clicking the menu item will toggle the On/Off status. When Launch on double-click is 'On' you can launch an executable or any file listed in the open archive, that associated with an application in the File Manager.

When ZipServer is launching an archived application it attempts to extract the file. If the file exists and Overwrite is unchecked it will NOT overwrite the existing file and will ask you if you want to launch the existing file instead.

When the ZipServer launched an application the **Save Launched** checkbox becomes enabled and checked. This will remind you to update the archive after the launched application is terminated. The checkbox will remain in this status until you either clicked the checkbox or unloaded the ZipServer. Clicking the Save Launched check box will update the archive and you will be asked if the original is to be deleted. The situation will be the same when you launch an existing application while the Overwrite check box is unchecked, but you will be asked if you want to update the file in the archive.

The **Launch on double-click** option will be disabled while the Save Launched checkbox is marked. You must finish your launching business (click Save Launched) prior starting another one to make sure you saved what you needed to.

As it was stated above you can launch an executable or any file listed in the open archive, that associated with an application in the File Manager. If you attempt to launch a file that is not an application itself and not associated with any application the ZipServer will display a message, but will not be able to start the application.

Stay on Top

When the **Options - Stay on Top** menu item is checked ZipServer will always be visible and appear on the top of other windows. Clicking the menu item toggles the On/Off status.

Using Dynamic Data Exchange (DDE)

Visual Basic 3.0
MsWord 6.0
Other Applications

Visual Basic 3.0

ZIP_RMV.MAK and SEARCH.MAK fully commented sample client sources are included in the installation package. ZIP_RMV.MAK will load ZIP_RMV.FRM as the main module, and DDEZIP.BAS that contains the data conversion routines for communication with the ZipServer.

MsWord 6.0

Edit WORDCODE.TXT with Notepad and copy it into a Macro in Word.

Other Applications

Edit ZIP_RMV.FRM and DDEZIP.BAS and WORDCODE.TXT files with a text editor (Notepad), and use the same approach as sample in the application you want to link to ZipServer. Note that the code used in the sample should be understandable for those are familiar with Access Basic or Excel Basic.

You can also using the File Manager - Associate command to associate all files having .ZIP extension with ZipServer. It will result in that clicking any file with ZIP file extension will automatically invoke ZipServer.

Using the ZipServer dynamic-link library

ZipServer v.1.5 includes ZIPSERV.DLL a Dynamic-link library for Visual Basic programmers. The sample Visual Basic application ZIP_RMV.MAK demonstrates the way that its functions can be used. It also includes ZIPSERV.RVB that should always be added to a Visual Basic program that calls ZIPSERV functions. Note that, the ZIPSERV.DLL library will NOT work without the ZipServer software is being installed on the computer or Network. The library extends the Visual Basic programming language with the following functions:

Related Topics:

AddZipComment
AppIsRunning
DisplayProcess
DisplayProcessLTW
DisplayTextFile
DriveType
DropCombo
ExtractBase
ExtractFile
ExtractPath
FoundInSearch
FromIni
FromWinIni
GetWindowPos
GetZipComment
MultiUnzip
MultiZip
MultiZipCount
MultiZipped
ReadOnlyTextBox
RemoveIniLine
RemoveIniSection
SearchFor
StayOnTop
Unzip
WinPath
WinSysPath
WriteIni
WriteWinIni
Zip
ZipCount
ZipError
Zipped
ZZip

AddZipComment

Declare Sub ***AddZipComment*** Lib "ZIPSERV.DLL" (ByVal *ZipFile* As STRING, ByVal *Comment* As STRING)

Purpose: Adding a Comment to an existing Zip file

VARIABLE	DESCRIPTION
ZipFile	The name of the Zipfile including the complete path, like "C:\ZIPFLS\BACKUP.ZIP"
Comment	A string to add as a comment

AppIsRunning

Declare Function AppIsRunning Lib "ZIPSERV.DLL" (ByVal *Application* As STRING) As INTEGER

Purpose: Answer the question the application is running or not

VARIABLE	DESCRIPTION
Application	The name of the executable including file extension. Example: Application = "CLOCK.EXE"
ReturnValue	The hWnd property value of the main form of the application if the application is running or False (zero)

DisplayProcess

Declare Sub ***DisplayProcess*** Lib "ZIPSERV.DLL" (ByVal *status* As INTEGER)

Purpose: Displaying the Zipping and Unzipping process

VARIABLE		DESCRIPTION
status	0	do not display
	1 (or non zero)	display

DisplayProcessLTW

Declare Sub ***DisplayProcessLTW*** Lib "ZIPSERV.DLL" (ByVal *status* As INTEGER, ByVal *dLeft* As INTEGER, ByVal *dTop* As INTEGER, ByVal *dWidth* As INTEGER)

Purpose: Displaying the Zipping and Unzipping process in sized

form in a predefined location

VARIABLE	DESCRIPTION	
status	0	do not display
	1 (or non zero)	display
dLeft	the position of the left side of the form in Twips	
dTop	the position of the top of the form in Twips	
dWidth	the width of the form in Twips	

DisplayTextFile

Declare Sub ***DisplayTextFile*** Lib "ZIPSERV.DLL" (ByVal *Textfile* As STRING, ByVal *Title* As STRING, ByVal *hWndParent* As INTEGER, ByVal *ReadOnly* As INTEGER)

Purpose: Displaying a small text file and print it as needed

VARIABLE	DESCRIPTION
Textfile	The name including the full path of the textfile to be displayed. If the file does not exist nothing will be displayed, but no error will occur
Title	The Caption property of the Form to be displayed
hWndParent	The hWnd property value of the form in the calling application
ReadOnly	True (non zero) value results in read only display

DriveType

Declare Function ***DriveType*** Lib "ZIPSERV.DLL" (ByVal *DriveLetter* As STRING) As INTEGER

Purpose: Obtaining information about the type of a drive

VARIABLE	DESCRIPTION	
DriveLetter	The drive letter, like "A" or "Z"	
Return value	0	drive not found
	2	floppy disk drive
	3	hard disk drive
	4	remote (network) drive

DropCombo

Declare Function ***DropCombo*** Lib "ZIPSERV.DLL" (ByVal *hWnd* As

INTEGER) As INTEGER

Purpose: Displaying a combo box in dropped condition

VARIABLE	DESCRIPTION
hWnd	The hWnd property value of the combo box
Return value	Non zero value for failure

ExtractBase

Declare Function ***ExtractBase*** Lib "ZIPSERV.DLL" (ByVal *File* As STRING) As STRING

Purpose: Retrieving the base of the file name

VARIABLE	DESCRIPTION
File	The name of a file with full path
Return value	The basename of the file in the File string, like if File = "C:\VSR\SOHAJ.INI" the return value will be "SOHAJ"

ExtractFile

Declare Function ***ExtractFile*** Lib "ZIPSERV.DLL" (ByVal *File* As STRING) As STRING

Purpose: Retrieving the complete File name without the path

VARIABLE	DESCRIPTION
File	The name of a file with full path
Return value	The name of the file in the File string, like if File = "C:\VSR\SOHAJ.INI" the return value will be "SOHAJ.INI"

ExtractPath

Declare Function ***ExtractPath*** Lib "ZIPSERV.DLL" (ByVal *File* As STRING) As STRING

Purpose: Retrieving the path of a file

VARIABLE	DESCRIPTION
File	The name of a file with full path
Return value	The path section of the File string, like if File = "C:\VSR\SOHAJ.INI" the return value will be "C:\VSR\"

FoundInSearch

Declare Function ***FoundInSearch*** Lib "ZIPSERV.DLL" (ByVal *item* As INTEGER) As STRING

Purpose: Retrieving the result of a file search (see SearchFor function)

VARIABLE	DESCRIPTION
item	The number of the file found in the SearchFor function. The value must be between one(1) and the number of files found. Use it to populate an array or list box.
Return value	The name of the file including the complete path

FromIni

Declare Function ***FromIni*** Lib "ZIPSERV.DLL" (ByVal *Ininame* As STRING, ByVal *Section* As STRING, ByVal *Request* As STRING) As STRING

Purpose: Retrieving data from an INI file

VARIABLE	DESCRIPTION
Ininame	Name of the .INI file including the full path
Section	The name of the section in the .INI file that will be in rectangular brackets, like [Section]
Request	The name of the variable in the .INI file
Return value	The value of the Request variable from the .INI file

FromWinIni

Declare Function ***FromWinIni*** Lib "ZIPSERV.DLL" (ByVal *Appname* As STRING, ByVal *Request* As STRING, ByVal *Default* As STRING) As STRING

Purpose: Retrieving data from the WIN.INI file

VARIABLE	DESCRIPTION
Appname	The name of the section in the .INI file
Request	The name of the variable in the .INI file
Default	The default value of Request if the value is missing it will be added
Return value	The value of the Request variable from the .INI file

GetWindowPos

Declare Sub ***GetWindowPos*** Lib "ZIPSERV.DLL" (*dLeft* As INTEGER, *dTop* As INTEGER, *dWidth* As INTEGER)

Purpose: Retrieving the position and width of the DisplayProcessLTW form at design time

VARIABLE	DESCRIPTION
dLeft	The position of the left side of the form in twips
dTop	The position of the top of the form in twips
dWidth	The width of the form in twips
Special Note	There is no mistake, do not use ByVal in the declaration

GetZipComment

Declare Function ***GetZipComment*** Lib "ZIPSERV.DLL" (ByVal *ZipFile* As STRING) As STRING

Purpose: Retrieving a Comment from an existing Zip file

VARIABLE	DESCRIPTION
ZipFile	The name of the Zipfile including the complete path, like "C:\ZIPFLS\BACKUP.ZIP"
Return value	The comment of the Zipfile

MultiUnzip

Declare Function ***MultiUnzip*** Lib "ZIPSERV.DLL" (ByVal *ZipFile* As STRING, ByVal *Destination* As STRING, ByVal *FilesToUnzip* As STRING, ByVal *RestorePath* As INTEGER, ByVal *Overwrite* As INTEGER, ByVal *WPath* As STRING, ByVal *ZPath* As STRING) As INTEGER

Purpose: Extract files from a multi volume archive

VARIABLE	DESCRIPTION
ZipFile	the name of the ZipFile with the complete path. Note: RMV archive is not allowed in the same directory where the multivolume archive resides.
Destination	The Destination directory
FilesToUnzip	The name of the file including path to be unzipped. Wild cards are allowed.
RestorePath	True (non zero) when the stored path in the Zip file is

	to be restored
Overwrite	True (non zero) when existing files are to be overwritten
WPath	A Path for the ZipServer that can be used as working area (Read/Write access)
ZPath	The path to the ZIPSERV.DLL and ZIPSERV.LIC files. If a Null string is passed the Windows\System directory will be taken
ReturnValue	Zero (0) for success, or call ZipError(ReturnValue) for more details

MultiZip

Declare Function ***MultiZip*** Lib "ZIPSERV.DLL" (ByVal *ZipFile* As STRING, ByVal *FilesToZip* As STRING, ByVal *Recurse* As INTEGER, ByVal *StorePath* As INTEGER, ByVal *FirstSize* As LONG, ByVal *LaterSize* As LONG, ByVal *WPath* As STRING, ByVal *ZPath* As STRING) As INTEGER

Purpose: Creating a multi volume archive

VARIABLE	**DESCRIPTION**
ZipFile	the name of the ZipFile, like "A:\CONRAD.ZIP", with the complete path. Note: RMV archive is not allowed in the same directory where the multivolume archive resides.
FilesToZip	The name of the files to zip including the path, wild cards allowed, multiple files can be passed separated by an empty space
Recurse	True (non zero) when recursive subdirectories in the Zip file
StorePath	True (non zero) when the path is to be stored in the Zip file
FirstSize	The number of bytes allowed to write on the first diskette
LaterSize	The number of bytes allowed to write on the later diskettes
WPath	A Path for the ZipServer that can be used as working area (Read/Write access)
ZPath	The path to the ZIPSERV.DLL and ZIPSERV.LIC files. If a Null string is passed the Windows\System directory will be taken
ReturnValue	Zero (0) for success, or call ZipError(ReturnValue) for more details

MultiZipCount

Declare Function ***MultiZipCount*** Lib "ZIPSERV.DLL" (ByVal *ZipFile* As STRING) As LONG

Purpose: Retrieving the number of files in a multi volume archive

VARIABLE	DESCRIPTION
ZipFile	The name of the Zipfile including the complete path. Note: RMV archive is not allowed in the same directory where the multivolume archive resides.
Return value	The number of files in the multi volume archive

MultiZipped

Declare Function ***MultiZipped*** Lib "ZIPSERV.DLL" (ByVal *ListIndex* As LONG) As STRING

Purpose: Retrieving the name of a file from a multi volume archive

VARIABLE	DESCRIPTION
ListIndex	The number of the file in the directory from one (1) to the value returned by the MultiZipCount function. Note: RMV archive is not allowed in the same directory where the multivolume archive resides.
Return value	The name of the file in the multi volume archive

ReadOnlyTextBox

Declare Sub ***ReadOnlyTextBox*** Lib "ZIPSERV.DLL" (ByVal *hWnd* As INTEGER)

Purpose: Displaying text in a Text box but restricting the user for reading it only

VARIABLE	DESCRIPTION
hWnd	The hWnd property value of the text box

RemoveIniLine

Declare Function ***RemoveIniLine*** Lib "ZIPSERV.DLL" (ByVal *Ininame* As STRING, ByVal *Section* As STRING, ByVal *Request* As STRING) As INTEGER

Purpose: Removing a line from an .INI file

VARIABLE	DESCRIPTION
Ininame	Name of the .INI file including the full path
Section	The name of the section in the .INI file that will be in rectangular brackets, like [Section]
Request	The name of the variable in the .INI file
Return value	Non zero value for failure

RemoveIniSection

Declare Function ***RemoveIniSection*** Lib "ZIPSERV.DLL" (ByVal *Ininame* As STRING, ByVal *Section* As STRING) As INTEGER

Purpose: Removing a whole section from an .INI file

VARIABLE	DESCRIPTION
Ininame	Name of the .INI file including the full path
Section	The name of the section in the .INI file that will be in rectangular brackets, like [Section]
Return value	Non zero value for failure

SearchFor

Declare Function ***SearchFor*** Lib "ZIPSERV.DLL" (ByVal *SearchString* As STRING, ByVal *Recursive* As INTEGER, ByVal *Display* As INTEGER) As INTEGER

Purpose: Executing a file search

VARIABLE	DESCRIPTION
SearchString	A string, containing the search requirements. Wild cards are allowed. The search might include several requirements separated by spaces. If the drive letter is missing the current drive will be searched.
Recursive	True (non zero) when recursive subdirectories in the search
Display	True (non zero) value will display a box in the center of the screen displaying the progress of the search
Return value	The number of files found

StayOnTop

Declare Sub ***StayOnTop*** Lib "ZIPSERV.DLL" (ByVal *hWnd* As INTEGER, ByVal *status* As INTEGER)

Purpose: Toggle if a form can be covered or not

VARIABLE	DESCRIPTION	
hWnd	The hWnd property value of the Form	
status	0	do not stay on top
	1	stay on top

Unzip

Declare Function ***Unzip*** Lib "ZIPSERV.DLL" (ByVal *ZipFileName* As STRING, ByVal *Dest* As STRING, ByVal *F2Unzip* As STRING, ByVal *Rest* As INTEGER, ByVal *OW* As INTEGER, ByVal *Pword* As STRING, ByVal *WPath* As STRING, ByVal *ZPath* As STRING) As INTEGER

Purpose: Unzipping a PKZip 2.04g compatible archive from a Visual Basic application

VARIABLE	DESCRIPTION
ZipFileName	the name of the ZipFile, like "C:\MY\CONRAD.ZIP", with the complete path
Dest	The Destination directory where the file(s) to be unzipped
F2Unzip	The name of the file including path that to be unzipped. Wild cards are allowed.
Rest	True (non zero) when the stored path in the Zip file is to be restored
OW	True (non zero) when existing files are to be overwritten
Pword	Password (if any)
WPath	A Path for the ZipServer that can be used as working area (Read\Write access)
ZPath	if a Null string is passed the Windows\System directory will be taken
ReturnValue	Zero (0) for success, or call ZipError(ReturnValue) for more details

Important: If the passed value of ZPath is incorrect the ZIPSERV.DLL's About screen will be displayed with the notice that the Free trial period expired.

WinPath

Declare Function ***WinPath*** Lib "ZIPSERV.DLL" () As STRING

Purpose: Retrieving the path of the Windows directory

VARIABLE	DESCRIPTION
Return value	The path to the Windows directory. The path ends with backslash, like "C:\WINDOWS\"

WinSysPath

Declare Function ***WinSysPath*** Lib "ZIPSERV.DLL" () As STRING

Purpose: Retrieving the path of the Windows\System directory

VARIABLE	DESCRIPTION
Return value	The path to the Windows\System directory. The path ends with backslash, like "C:\WINDOWS\SYSTEM\"

WriteIni

Declare Function ***WriteIni*** Lib "ZIPSERV.DLL" (ByVal *Ininame* As STRING, ByVal *Section* As STRING, ByVal *Request* As STRING, ByVal *Value* As STRING) As INTEGER

Purpose: Writing into an INI file

VARIABLE	DESCRIPTION
Ininame	Name of the .INI file including the full path
Section	The name of the section in the .INI file that will be in rectangular brackets, like [Section]
Request	The name of the variable in the .INI file
Value	The value of Request
Return value	Non zero value for failure

WriteWinIni

Declare Function ***WriteWinIni*** Lib "ZIPSERV.DLL" (ByVal *Appname* As STRING, ByVal *Request* As STRING, ByVal *Value* As STRING) As INTEGER

Purpose: Writing into the WIN.INI file

VARIABLE	DESCRIPTION
Appname	The name of the section in the .INI file
Request	The name of the variable in the .INI file
Value	The default value of Request if the value is missing it will be added
Return value	Non zero value for failure

Zip

Declare Function ***Zip*** Lib "ZIPSERV.DLL" (ByVal *ZipName* As STRING, ByVal *FileToZip* As STRING, ByVal *Keep* As INTEGER, ByVal *Store* As INTEGER, ByVal *Recur* As INTEGER, ByVal *Pword* As STRING, ByVal *WPath* As STRING, ByVal *ZPath* As STRING) As INTEGER

Purpose: Zipping files to a PKZip 2.04g compatible archive from a Visual Basic application

VARIABLE	DESCRIPTION
ZipName	the name of the ZipFile, like "C:\MY\CONRAD.ZIP", with the complete path
FileToZip	the name of the file to zip including the path, wild cards allowed
Keep	True (non zero) when the original date at update to be kept in the Zip file
Store	True (non zero) when the path is to be stored in the Zip file
Recur	True (non zero) when recursive subdirectories in the Zip file
Pword	Password (if any)
WPath	A Path for the ZipServer that can be used as working area (Read\Write access)
ZPath	if a Null string is passed the Windows\System directory will be taken
ReturnValue	Zero (0) for success, or call ZipError(ReturnValue) for more details

Important: If the passed value of ZPath is incorrect the ZIPSERV.DLL's About screen will be displayed with the notice that the ***Free trial period expired.***

ZipCount

Declare Function ***ZipCount*** Lib "ZIPSERV.DLL" (ByVal *ZipFile* As STRING) As LONG

Purpose: Retrieving the number of files in an archive

VARIABLE	DESCRIPTION
ZipFile	The name of the Zipfile including the complete path
Return value	The number of the files in the archive

ZipError

Declare Function ***ZipError*** Lib "ZIPSERV.DLL" (ByVal *ErrCode* As INTEGER) As STRING

Purpose: Displaying zipping/unzipping error messages

VARIABLE	DESCRIPTION
ErrCode	The error number as returned by the Zip and Unzip functions
Return value	The description of the error that occured

Zipped

Declare Function ***Zipped*** Lib "ZIPSERV.DLL" (ByVal *ListIndex* As LONG, ByVal *Volume* As INTEGER) As STRING

Purpose: Retrieving the name of a file from an archive

VARIABLE	DESCRIPTION
ListIndex	The number of the file in the archive from one (1) to the value returned by the ZipCount function
Volume	The number of the file of an RMV series in which the file resides
Return value	The name of the file

ZZip

Declare Function ***ZZip*** Lib "ZIPSERV.DLL" (ByVal *ZipName* As STRING, ByVal *FileToZip* As STRING, ByVal *pkSwitch* As STRING, ByVal *Password* As STRING, ByVal *WPath* As STRING, ByVal *ZPath*

As STRING) As INTEGER

Purpose: Zipping files to a PKZip 2.04g compatible archive or to an RMV archive from a Visual Basic, Application Basic, C++ application using PKZIP compatible switches

VARIABLE	DESCRIPTION
ZipName	the name of the ZipFile, like "C:\MY\CONRAD.ZIP", with the complete path. If the file extension is ".RMV" an RMV archive series will be created. Note: Multivolume archive is not allowed in the same directory.
FileToZip	the name of the file to zip including the path, wild cards allowed
pkSwitch	A PKZIP compatible switch string might contains the following instructions: -k keep original date of the archive at updating -r recursive (archives all matching files in the subdirectories) -p same as -n -n store subdirectories -f freshen (updates all matching files that are already in the archive in case if the source file is newer than the one in the archive) -u update (updates all matching files that are already in the archive in case if the source file is newer than the one in the archive and in addition to that adds all matching files to the archive which are not there) -d delete matching files from the archive
Password	Password (if any)
WPath	A Path for the ZipServer that can be used as working area (Read\Write access)
ZPath	if a Null string is passed the Windows\System directory will be taken
ReturnValue	Zero (0) for success, or call ZipError(ReturnValue) for more details

Important: If the passed value of ZPath is incorrect the ZIPSERV.DLL's About screen will be displayed with the notice that the ***Free trial period expired.***

About RMV Archive Series

The RMV Archive Series is introduced by Redei Enterprises and as of today is unique to the ZipServer. However, RMV is not a new compression algorithm. It is 100% PKZIP 2.04g compatible. The naming stands for how the ZipServer handles a series of archive files rather than for the structure of the archive itself.

The RMV Archive Series is used for a series of PKZIP 2.04g compatible archive files following a naming convention. All files must have the same filename. The file extension of the first file must be RMV, while the second's extension is 001', the third's 002', etc. The minimum number of the files in a RMV Archive series is one, in which case it is identical with a .ZIP file. The maximum number of the files is one thousand. All members of the series must reside in the same directory. The purpose of the RMV Archive series is:

- greatly extend the limitations of the archive size
- increase the safety
- provide the same handling for the user as it would be a single file

Limitations of RMV archive series

ZipServer can not open a normal ZIP archive over 30MB, and the maximum number of files in the archive is also limited up to one thousand. The same limitations apply for each member of a RMV Archive Series. The ZipServer will automatically open a new archive and name it with the next number when the size of the archive exceeds 20 MB. When updating files in an existing archive, this size can grow up to ZipServers limit.

Considering these limitations the maximum size of an RMV Archive series is between 20 to 30 Gigabytes!

Working with RMV Archive series

Creating RMV Archive series
Converting ZIP files to RMV Archive series
Extracting files from RMV Archive series
Updating RMV Archive series

Creating RMV Archive series

Using your own program with calls to the ZIPSERV.DLL an RMV archive series is created by simply naming the archive with an RMV file extension. For example, naming a new archive as ***SAMPLE.RMV*** will automatically create the subsequent members of the series when it is needed.

Please note that, the executable file (ZIPS.EXE) of the ZipServer does not create, update and extract RMV archive series.

Converting ZIP files to RMV Archive series

Name the first file with RMV extension instead of the common ZIP extension. Rename all other archives with the same filename and with extensions 001, 002, 003, etc. Make sure that, none of the files exceeds the maximum of 20 MB size limitation. It does not make any difference that, which file is named as the first RMV file, or what is the sequence in the series.

Extracting files from RMV Archive series

The ZIPSERV.DLL provides functions for listing all files in a RMV Archive series. You can select files for extraction just if they would reside in one single archive.

Updating RMV Archive series

You can freshen or update RMV Archive series exactly the same way as you would do if they reside in one single archive.

Compatibility

Using PKZIP for RMV Archive series
Multi volume and RMV archives

Using PKZIP for RMV Archive series

PKZIP can maintain and extract members of an RMV Archive series as individual archives. While it is more comfortable using the ZipServer, PKZIP can be used as well.

Multi volume and RMV archives

The similarity of the naming convention requires that multi volume archives and RMV archives can not share the same directory. The ZIPS.EXE file will assume multi volume archives when detecting numberized members of a series and will return "Unknown archive type" error message if the members are actually part of an RMV series instead.

Glossary of Terms

Compression Plus
Destination Path
List File
Multi Volume archives
PKZIP 2.04g
Source Files
Visual Basic

Compression Plus

Product of EllTech Development, Inc., 4374 Shallowford Industrial Parkway, Marietta, GA 30066

Destination Path

The Path to where the content or the designated part of the archive will be extracted. The Destination Path becomes the root when the original path is restored .

List File

A text file that lists other files in separate lines.

Multi Volume archives

The content of the archive file is too large for one floppy diskette. Typically is used for backup or installation package purposes.

PKZIP 2.04g

PKWARE's widely used utility program for DOS.

Source Files

Files to be archived.

Visual Basic

Microsoft -- Visual Basic V.3.0