

## **ZIPSHHELL 2.47 HELP INDEX**

**Commands**

**Procedures**

**Compression / Decompression Options**

**Explanation of Errors**

**Registration Information**

## ZipShell Commands

### File Menu:

- New Archive
- Open Archive/(New or Current Window)
- Close Archive
- Delete Archive
- Rename Archive
- Make EXE
- Exit

### Setup Menu:

- Options
- Internal Extraction
- Register ZipShell

### Sort Menu

ARJ Menu: (visible only when an ARJ file is open)

- Test Archive Integrity
- Join Archive
- Extended File Info

ZIP Menu: (visible only when a ZIP file is open)

- Test Zipfile Integrity
- Fix Corrupted Zipfile
- Extended File Info

### Help Menu:

- Help Index
- Help On Opening An Archive
- ZIPSHELL.WRI
- Order Form
- About

**NOTE:** The ARJ and ZIP menus are only visible when an archive of the appropriate format is open.

## Sort Menu

The Sort Menu contains several commands which let you sort the files displayed in the main listbox according to various criteria. When you open an archive, the list is initially displayed in the "original order," the order that the files occur within the archive. You can choose to sort the list by filename, extension, length, or last modification date simply by selecting the appropriate command from the Sort Menu.

## Internal Extraction

Selecting this command from ZipShell's **Setup** menu will bring up a window containing ZipShell's internal extraction options.

ZipShell has the ability to extract and test most ZIP and ARJ archives internally, without shelling to DOS to use the appropriate unarchiver. To enable this feature, check both the "Use Internal Unzip When Possible" and "Use Internal Unarj When Possible" boxes.

When extracting files internally, ZipShell uses a percent bar to display its progress. However, for small files, displaying this percent bar will slow down the extraction process considerably. Therefore, for each archive type, a list of different behaviors for the percent bar is listed. Selecting "Always Use Percent Bar" will cause ZipShell to display the percent bar regardless of file size, and selecting the "Never Use Percent Bar" will cause ZipShell to extract files without using the percent bar. The third choice, "Use Percent Bar for Files > 20K", causes ZipShell to show the percent bar only when extracting files greater than 20K in length. This threshold value (20K in this case) is also modifiable.

If an error occurs during an internal extraction process, ZipShell will attempt to extract the affected file using either PKUNZIP.EXE or ARJ.EXE.

**NOTE:** A 386 or better processor is required for internal ARJ and ZIP extraction.

## Extended File Info

This command is available under both the ARJ and ZIP menus when an archive of the appropriate type is open. Selecting this command will open a window which will display an extended listing of the current archive, such as might be displayed when using PKUNZIP.EXE or ARJ.EXE from the DOS prompt. It includes information not displayed in ZipShell's main listing, such as time stamp, CRC, etc.

## **Compression / Decompression Options**

**ARJ Compression Options**

**ARJ Extraction Options**

**ZIP Compression Options**

**ZIP Extraction Options**

## Register ZipShell

After registering ZipShell, you will receive a Serial Number from NewVision. After you install ZipShell, type in your name and Serial Number (**exactly** as it appeared in the letter sent to you from NewVision) in the spaces provided in this window. ZipShell will then behave as a "registered" program. That is, your name and serial number will be displayed in the About box, and the About box will not appear on startup.

You may also have to repeat this process if you damage or erase the ZIPSHELL.INI file, located in your /windows directory.

## Registration Information

ZipShell is not public domain software, nor is it free. It is ShareWare. Non-licensed users are granted a 21-day period during which they may use ZipShell on a trial basis, to see if it is suitable for their needs. After the 21-day trial period, users are required to register the program or discontinue using it.

The single-license registration fee for ZipShell is \$25.00 (plus \$2.00 S/H in the US). In return you will receive a diskette containing the latest version of ZipShell (without the nag screen!) and accompanying documentation files, along with a unique registration number. Additionally, your name will be displayed in the "About" box as the registered user. Registering ZipShell enables us to continue to develop high-quality, inexpensive software for you to use, so please support the Shareware concept and register this program if you find it of use! Also, if you register you will be eligible for discounts on futures upgrades.

Commercial site licenses are also available. Please refer to ZSORDER.TXT for rates.

To order, please fill out the accompanying order form (ZSORDER.TXT), and send to:

NewVision  
800 5th Ave #382  
Seattle, WA 98104

To print out the order form, use an editor such as NOTEPAD or select the **Order Form** command from ZipShell's **Help** menu.

### **Online Registration Via CompuServe**

You may also register ZipShell by charging it to your CompuServe account. To do this, type GO SWREG at any CompuServe ! prompt, and use ID 153. As soon as you do this, NewVision is notified, and you should receive your copy of ZipShell in the mail within a week or so.

Additionally, if you have any questions, problems, comments, or suggestions regarding ZipShell, contact us at the above address, or on CompuServe at 76020,732. All comments and suggestions are appreciated and welcome.



## Enter Password

The file you have selected to view is scrambled with a password. You must enter the password in the box provided and press OK in order for the file to be extracted successfully. If you specify an incorrect password, you will receive an error.

## Make EXE File / Module Size

Using ZipShell, you can create a stand-alone executable (EXE) file from any ZIP, ARJ, or LZH archive. Code is added to the archive which enables it to be extracted without the use of a separate unarchiving program (i.e., PKUNZIP, ARJ, or LHA). Thus, archives modified in this manner may be extracted by people who do not possess the original archiver, simply by executing the file.

When creating this self-extracting archive (sometimes referred to as an SFX file), some decompression code is attached to the file to enable it to run by itself. Therefore, the size of the archive is increased. When creating an EXE file with ZipShell, you may specify either the large or small SFX module to be attached to the archive. They both contain the code necessary to decompress the archive; however, the larger archive contains code which gives the EXE file greater capabilities, such as the creation of subtrees. The large SFX module usually adds about 10K extra to your archive.

It is suggested that unless you need one of the capabilities provided only by the large SFX module, you use the small SFX module instead. This will result in a more compact EXE file. To determine what capabilities are supported by the different SFX modules, refer to the documentation that comes with the appropriate archiver.

## Make EXE File

Using ZipShell, you can create a stand-alone executable (EXE) file from any ZIP, ARJ, or LZH archive. Code is added to the archive which enables it to be extracted without the use of a separate unarchiving program (i.e., PKUNZIP, ARJ, or LHA). Thus, archives modified in this manner may be extracted by people who do not possess the original archiver, simply by executing the file.

To create an EXE file from an existing archive, simply open the archive and choose the **Make EXE** command from ZipShell's **File** menu. ZipShell will activate the appropriate archiver, which will attach the necessary code to make the archive a self-extracting EXE file. When the process is complete, the newly created EXE file will be opened in a new window for your inspection.

ZipShell treats self-extracting archives just like any other archive. You can perform any action on such a file that you can on any archive, such as extracting files, adding files, deleting files, etc.

## ZIP Extraction Options

When you extract files from a ZIP file, a window will appear with the following extraction options.

### **-o: Overwrite Existing Files**

If this box is checked, files in the archive will overwrite existing files with the same names when they are extracted. If this box is not checked, ZipShell will check for the presence of files before extracting, and if duplicate names are found, will ask you whether to delete the existing file, or skip the file's extraction.

### **-d: Recreate Stored Paths**

If this box is checked, any files stored with path information will be extracted using this information. That is, they will be extracted to the subdirectories stored in the ZIP file.

### **-\$: Extract Volume Label**

If this box is checked, any volume label stored in the ZIP archive will be extracted to the drive that the files are being extracted to.

### **-s: Unscramble With Password**

Checking this box will cause PKUNZIP to descramble any scrambled files with the password given in the text box below the option. You must specify a password in order to extract garbled files. Specifying and password for non-scrambled files will have no effect on them.

## ZIP Compression Options

When you add files to a ZIP file or create a new one, a window will appear with the following compression options.

### Compression Options

ZipShell lets you specify the method of compression used to compress your files. The choices shown depend on which version of PKZIP you are using.

#### Version 1.10

The first option, "**-ex: Use Imploding**", is the default, and generally results in better compression and uses less memory than the other options. However, there are three more available options. The "**-es: Use Shrinking**" option uses more memory and generally is not as efficient. The other two options are combination of the above options. The "**-ea: Implode ASCII, Shrink Binary**" option will Implode all ASCII files (generally documents) and Shrink all binary files (programs, etc). The "**-eb: Implode Binary, Shrink ASCII**" options does just the opposite.

#### Version 2.04

The first option, "**-ex: Max compression**", results in the best compression ratios, but takes the most time. The last option, "**-es: Super-fast compression**", takes the least time but also compresses the least. The middle two options are compromises between speed and compression. The second option, "**-en: Normal compression**", is the default when using PKZIP from the DOS command line.

#### **-P: Include Pathnames**

Checking this box will cause PKZIP to include pathnames inside the ZIP file. Files can then be restored to their original locations during extraction.

#### **-!: Use Authenticity Verification**

This option is available only to registered users of PKZIP who have inserted an authenticity code into their copy of PKZIP.EXE using the PUTAV.EXE program provided with PKZIP. Checking this box will insert this authenticity code into a ZIP file. If a file created with this code is subsequently modified, a Warning Error will be reported when the file is tested or files are extracted from it.

#### **-b: Store Temp Zipfile on Drive**

When PKZIP updates an archive, it creates a temporary file which can consume considerable disk space. Normally, the file is created on the same drive as the ZIP archive which is being updated is located. This can cause Disk Full errors to occur if you are working with a ZIP archive on a floppy disk, even if it seems there is plenty free space. To fix the problem, check this option, and set the drive box below it to a local hard drive. This will cause PKZIP to store the temporary file on the specified disk, freeing up space on the floppy.

#### **-\$: Add Volume Label Of Drive**

Checking this option will cause the volume of the drive indicated by the drive box below to be added to the ZIP file. It can then be restored during extraction.

#### **-s: Scramble With Password**

Checking this option will cause all files added to a ZIP file in the current operation to be scrambled using the password specified in the text box below. **REMEMBER THIS PASSWORD**, because neither ZipShell nor PKZIP retains it. It is necessary to specify the password in order for the files to be successfully extracted.

## ARJ Extraction Options

When you extract files from an ARJ archive, a window will appear with the following extraction options.

### **-jyo: Overwrite Existing Files**

If this box is checked, files in the archive will overwrite existing files with the same names when they are extracted. If this box is not checked, ZipShell will check for the presence of files before extracting, and if duplicate names are found, will ask you whether to delete the existing file, or skip the file's extraction.

### **-\$: Extract Volume Label**

If this box is checked, any volume label stored in the ARJ archive will be extracted to the drive that the files are being extracted to.

### **-jr: Recover Broken Files**

Checking this option will allow ARJ to extract files with CRC errors, as well as extract files from archives with more serious errors. It should be noted that files extracted with this switch on may contain errors. Also, after the extraction process is complete, a Fatal Error will be reported. This is normal and does not indicate a serious problem.

### **-g: UnGarble With Password**

Checking this box will cause ARJ to descramble any scrambled files with the password given in the text box below the option. You must specify a password in order to extract garbled files. Specifying and password for non-scrambled files will have no effect on them.

## ARJ Compression Options

When you add files to an ARJ file or create a new one, a window will appear with the following compression options.

### Compression Method

There are five available compression methods to choose from. The first, **-m0**, simply stores files in an ARJ archive and does not compress them. The remaining options, from **-m1** to **-m4**, are each faster, compress less, and consume less memory than the previous option. The option **-m1**, which delivers maximum compression, is the default option.

### -k: Keep a .BAK of ARJ Archive

If this box is checked, and an ARJ archive is about to be modified, a backup copy of the archive will be created with a BAK extension before the original archive is changed. If a BAK file already exists, it will be deleted.

### -e: Exclude Pathnames

If this box is checked, ARJ will NOT include a file's path in the ARJ archive. ARJ's default method is to include pathnames, so if you do not want them, it is essential to keep this box checked. If this box is clear, files added to an ARJ archive will include their paths.

### -w: Use Working Drive

When ARJ updates an archive, it creates a temporary file which can consume considerable disk space. Normally, the file is created on the same drive as the ARJ archive which is being updated is located. This can cause Disk Full errors to occur if you are working with an ARJ archive on a floppy disk, even if it seems there is plenty of free space. To fix the problem, check this option, and set the drive box below it to a local hard drive. This will cause ARJ to store the temporary file on the specified disk, freeing up space on the floppy.

### -\$: Add Volume Label Of Drive

Checking this option will cause the volume of the drive indicated by the drive box below to be added to the ARJ file. It can then be restored during extraction.

### -g: Garble With Password

Checking this option will cause all files added to an ARJ file in the current operation to be scrambled using the password specified in the text box below. **REMEMBER THIS PASSWORD**, because neither ZipShell nor ARJ retains it. It is necessary to specify the password in order for the files to be successfully extracted.

## Help Index

Selecting the **Help Index** command from ZipShell's **Help** menu brings up this help file and displays its index.



## About

Selecting the **About** command from ZipShell's **Help** menu will cause ZipShell to displays a window containing information such as version number, copyrights, NewVision's address, etc. Also displayed are your computer's CPU type, and the mode in which Windows is currently running.

## Order Form

Selecting the **Order Form** command from ZipShell's **Help** menu will cause ZipShell to open the file ZSORDER.TXT using NOTEPAD. You may then view and print this file, if you wish.

For information about registering ZipShell, see [Registration Information](#).

## **ZIPSHELL.WRI**

Selecting the **ZIPSHELL.WRI** command from ZipShell's **Help** menu will cause ZipShell to launch WRITE.EXE and open the file ZIPSHELL.WRI for you to view, which contains important information about ZipShell.

## Help On Opening An Archive

Selecting the **Help On Opening An Archive** command from ZipShell's **Help** menu brings up this help file and displays the help screen available for the **Open Archive** command.

## **File: Exit Command**

Selecting the **Exit** command from the **File** menu closes ZipShell. It is not necessary to close any open archive with the **Close Archive** command first.

It is best not to exit ZipShell while one of the archivers it launched is still running. If you do, ZipShell will leave several temporary files on your hard drive that it normally would have cleaned up.

## **ZipShell Procedures**

**Adding Files To An Archive**

**Closing An Archive**

**Creating A New Archive**

**Deleting File(s) Within An Archive**

**Deleting An Entire Archive**

**Drag And Drop Operation**

**Error Detection**

**Extracting Files From An Archive**

**Opening An Archive**

**Renaming An Archive**

**Viewing A File Within An Archive**

**Working With Archive Comments**

**ZipShell Setup**

## Error Detection

ZipShell performs error detection when working with ZIP and ARJ archives. This means that when ARJ.EXE, LHA.EXE, PKZIP.EXE, or PKUNZIP.EXE reports an error, ZipShell intercepts it and delivers the error to you. If you receive an error, a message box will appear and ask if you want an explanation for the error. If you reply 'yes', you will be shown an explanation for it as well as possible solutions.

If you receive the message "**Error Detection Failed**" on the status bar after some operation, it means that ZipShell was unable to determine whether or not an error occurred. There are two possible reasons for this:

- 1) You are using ZipShell with a version of PKZIP other than 1.10 or 2.04c or a version of ARJ other than 2.30. If this is the case, the archiver may report errors with which ZipShell is not familiar.
- 2) One of the files created by ZipShell on the fly was moved, deleted, or renamed during a compression/decompression operation. If this is the case, simply try the operation again.

## Working With Archive Comments

**NOTE:** ZipShell only supports archive comments in archives of ZIP or ARJ format. ZipShell does not currently support comments for individual files within an archive, only for comments for the archive itself.

When you open a ZIP or ARJ archive, the comment button will be enabled. If the button shows the word "NOTE" written in red on the picture of the piece of paper, it means that the file contains an archive comment. If the word "NOTE" is missing, the file does not contain a comment.



Indicates no comment included in archive.



Indicates a comment is included in the archive.

If a comment is present, click on the comment button to view it. Whether or not a comment is present, you may click on the comment button to bring up the comment editor. Here you may view the file's comment (if any), and make any changes to it you wish. You may copy text into the clipboard with another application and then paste it into the comment with the paste button. When you are done viewing/editing the comment, press OK. If the comment has changed, ZipShell will ask you if you wish to save the new comment back into the archive. If you reply 'yes', the new comment will be inserted into the archive.

**NOTE:** Many archives downloaded from BBS's contain BBS advertisements which may look rather strange in Windows proportional font. To better view these, check the "Fixed Pitch Font" box at the bottom of the window.



## **File: New Archive Command**

Selecting the **New Archive** command from ZipShell's **File** menu opens up a new (untitled) window, which you may then add files to.

**NOTE:** ZipShell allows a maximum of 6 archives to be open at once. If you have already have this number of files open when selecting the File...New command, ZipShell will request that you close an existing archive before trying to create a new one.

There are two methods of adding files to a new archive:

### **Method 1**

You may select and drag any number of files from the File Manager and drop them anywhere on ZipShell's window or icon. You will be prompted for a filename for the new archive. You may choose to create an ARJ, ARC, LZH or ZIP file, merely be using the appropriate file extension when specifying the name for your archive. You will then be asked for confirmation, and if the file is a ZIP or ARJ file, you will be presented with a list of available compression options. See **Drag And Drop Operation** for more information on using ZipShell's drag and drop capabilities, or **ZIP Compression Options** or **ARJ Compression Options** for more information on options available during compression.

### **Method 2**

Press the **Add Files** button, which will bring up a window that enables you to navigate your system looking for files you wish to add. When you press this window's OK button, you will be prompted for the new archive's name and location. Enter any valid name and ZipShell will activate the appropriate archiver to compress the files. When the archiver is finished, ZipShell will open the newly created archive and display its contents.

## ARJ: Test Archive Integrity Command

**NOTE:** The ARJ menu is available only when an ARJ archive is open.

Selecting the **Test Archive Integrity** command from the **ARJ** menu will cause ZipShell to launch ARJ.EXE to test the current archive's integrity. You should use this command if you suspect an archive has been corrupted. If an error is found in the archive, it will be reported; otherwise, it will be stated that no errors were found.

If you have internal extraction for arjfiles turned on, this process will be done internally and will not require ARJ.EXE.

If an error is found using this command, you should extract all the files in the archive using the **"-jr: Recover Broken Files"** option, and then make a create a new archive with the extracted files. Be careful, though, because some of the extracted files may contain errors.

## ZIP: Test Zipfile Integrity

**NOTE:** The ZIP menu is available only when a ZIP archive is open.

Selecting the **Test Zipfile Integrity** command from the **ZIP** menu will cause ZipShell to launch PKUNZIP.EXE to test the current zipfile's integrity. You should use this command if you suspect an archive has been corrupted. If an error is found in the archive, it will be reported; otherwise, it will be stated that no errors were found.

If you have internal extraction for zipfiles turned on, this process will be done internally and will not require PKUNZIP.EXE.

If a correctable error is found using this command, you should repair the zipfile using the Fix Corrupted Zipfile command.

## ZIP: Fix Corrupted Zipfile Command

**NOTE:** The ZIP menu is available only when a ZIP archive is open.

Selecting the **Fix Corrupted Zipfile** command from the **ZIP** menu will cause ZipShell to launch PKZIPFIX.EXE (which comes with PKZIP.EXE) to fix the current zipfile. You should use this command to reconstruct a damaged zipfile so files can safely be extracted from it. Note that even after a file has been fixed using this command, you may still receive Bad CRC errors.

PKZIPFIX.EXE reconstructs the zipfile and names it PKFIXED.ZIP. After PKZIPFIX is done, ZipShell automatically opens up PKFIXED.ZIP and displays it's contents.

## **ARJ: Join Archive Command**

**NOTE:** The ARJ menu is available only when an ARJ archive is open.

Selecting the **Join Archive** command from the **ARJ** menu will let you combine your current ARJ archive with another ARJ archive. After selecting this command, you will be presented with two dialog boxes. The first allows you to choose which other ARJ file you wish to join to the current ARJ file. The second allows you to define the name of the new ARJ file to be created from the two others. Note that the new ARJ file cannot have the same name as one of the files it is being created from.

After you press 'OK' on the second dialog box, ZipShell will activate ARJ.EXE to join the archives. When ARJ.EXE is finished, ZipShell will open the newly created archive and display its contents.

## **File: Close Archive Command**

To close an open archive, choose the **Close Archive** command from the **File** menu. If no archive is open, selecting this command will have no effect.

The primary use for this command is to clear the screen so that a new archive may be created in the current window. It is not necessary to close one archive before opening another, or before exiting the program.

## Drag And Drop Operation

One of ZipShell's most useful features is its use of the Windows 3.1 File Manager's drag and drop capabilities. This provides an easy, intuitive way to add files to an existing archive, or to create a new one.

**To add files to an existing archive** using the drag and drop method, first use ZipShell to open the archive to which you wish to add more files. Next, open the File Manager (if it is not already open), and select the files you wish to add. Then drag these files and drop them anywhere on the main ZipShell window (or its icon, if it's minimized). A dialog box will appear, listing the names of the files you have dropped on ZipShell, and ask you to confirm their addition to the current archive. In addition, you are asked if you wish to delete the original files after they are zipped. The default answer (if you press "Enter" in response to the question) is "No," which means that the files will be left in their original location after zipping. If you decide you do not want to add any files at this time, press the Cancel button.

After selecting "Yes" or "No" (depending on whether you want to delete the original files after zipping or not), ZipShell will activate the appropriate archiver to update the archive. When the archiver is done, ZipShell will re-read the archive and update its statistics.

**To create a new archive** using the drag and drop method, you follow similar steps. The only difference is that before you drop any files on ZipShell, no archive should currently be open in the active window. (See Closing An Archive for more information.) Then the files will be added to a new archive. As soon as you drop the files onto ZipShell, a dialog box will appear prompting you for a filename and directory for the new archive. Choose the directory you want it to be created in, type in the name of the archive, and press OK. If you do not specify an extension, ZipShell assumes you wish to create a ZIP file. If you wish to create an ARJ, LZH or ARC file instead, be sure to include its extension when specifying its file name. After you do this, the next dialog box will appear, listing the names of the files about to be added to the new archive. Proceed as outlined above.

## **File: Delete Archive Command**

ZipShell allows you to delete an entire archive. To do so, open the file you wish to delete, then choose the **Delete Archive** command from the **File** menu. A message box will pop up asking you to confirm the deletion, and if you reply "yes", the file will be deleted. The Delete Archive command is only available while an archive is open.

Note that this command deletes an **ENTIRE** archive. If you wish to delete particular file(s) within an archive, see [Deleting Files Within an Archive](#).



## File: Rename Archive Command

If you wish to rename an archive, it must first be open. (See [Opening An Archive](#) for more information.) Then select the **Rename Archive** command from the **File** menu of ZipShell's main window. You will be shown a dialog box with which you can specify the new name and path for the archive.

Note that you can use this command to move the file from one directory to another, AS LONG AS BOTH DIRECTORIES ARE ON THE SAME DRIVE. You cannot use this command to copy or move an archive from one drive to another. For example, if your current archive is "c:\windows\newstuff\utility.zip", you may rename it to "c:\dos\utility\new.zip." The file will then be renamed and moved from one location to the other. You can not, however, rename the above file to "d:\temp\utility.zip," because that involves a different drive. If you attempt to do this, ZipShell will generate a "Could not rename file" error.

If you specify a filename which already exists, you will be asked if you wish to overwrite the existing file. If you reply "yes," the existing file will be deleted so that the new file may be moved into its place.



## Extracting Files From An Archive

In order to **extract** files from a archive, you must have first **opened** it. Then click on the Extract button at the bottom of the screen. A dialog box will appear with two list boxes: the first (which initially contains all of the files within a archive) lists files which are to be extracted, and the second (which is initially empty) lists files which are NOT to be extracted. To move a file from one box to the other, double-click on it, or select it and choose the Move button. All the files in one box can be moved to the other by choosing the appropriate Move All button.

Once you have placed all the files in the appropriate list box (according to whether or not you wish to extract them) and examined the extraction path (see below), choose the Extract button. After you do this, ZipShell will then activate the appropriate archiver to decompress whichever files you have selected for extraction.

**NOTE:** If the archive you are working with is a ZIP or ARJ file, an additional window will appear before ZipShell activates the appropriate unarchiver, listing available extraction options. See [ARJ Extraction Options](#) or [ZIP Extraction Options](#) for more information.

### Extraction Path

At the very bottom of the dialog box is a text box containing the Extraction Path. This is the directory in which ZipShell will place the extracted files. If the directory shown is not the directory into which you wish to extract files, either type the name of the correct directory or choose the Browse button. This will bring up a dialog box which enables you to navigate your system, looking for a directory you wish to choose for extraction. It also allows you to create a new subdirectory if you so desire. Once you have selected the appropriate directory, choose the OK button, and that directory's name will be placed in the Extraction Path box in the extraction dialog box.

**NOTE:** If you wish to save your Extraction Path into ZIPSHELL.INI so that you do not have to type it in again, use the **Paths** command from ZipShell's **Options** window. See [ZipShell Paths](#) for more information.

The term "extracting files" (also called "unzipping" when in reference to ZIP files) is the process of expanding them, or decompressing them, from their compressed state into their original, uncompressed state, in order to make them usable.



## Viewing A File Within An Archive

To view a file within an archive, you must first **open the archive** which contains the file you wish to view. Then simply select the file in the listbox, and press the View button. Alternatively, you may double-click on the filename. In either case, ZipShell will execute the appropriate de-archiver to extract the selected file to a temporary location. After the de-archiver has finished, ZipShell will launch the editor you have specified under the "**Default Editor**" section of the **Options** window. If you haven't specified any editor, ZipShell uses Windows' NOTEPAD.EXE. After the editor is launched, it will open up the extracted file for you to view.

If the file you want to view is associated with a program in the [Extensions] section of your WIN.INI file, ZipShell will retrieve this information. For information on how to specify which viewer to use in this case, see the "**Viewer Behavior**" section of the **Options** window.

**NOTE:** All files unzipped for viewing will be deleted when ZipShell is closed. If you wish to retain any of these files, use the "**Save As**" command from the editor you are using to save the file to a different location.

## File: Open Archive Command

The process "opening an archive" refers to reading its directory in order to determine its contents. An archive must be opened before any other operations can be performed on it. There are two methods you may use to open an archive with ZipShell.

### Method 1

To open an archive, choose the **Open / Current Window** command (to open the archive in the current window, replacing any currently open archive), or the **Open / New Window** command (to open the archive in a new, empty window) from the **File** menu. Select the file you wish to open from the directory and file lists provided, and ZipShell will open it for you.

After ZipShell has finished reading the file, it will display its contents on the screen. The file's name, **length**, **size**, compression ratio, and last modification date are displayed. (For ZIP and ARJ files, additional information may be displayed, such as the fact that a particular file is encrypted.) From this point, you can go on to perform any operations on the archive (deleting, extracting, adding, viewing, etc.) you wish to.

### Method 2

You may also open an archive by dragging it from the File Manager and dropping it onto ZipShell. It will then automatically be opened for you in the current window. If you do this, make sure you only drag one file at a time, and that it has a ZIP, ARJ, LZH or ARC extension. If these conditions are not met, ZipShell will begin the process of adding the dropped file(s) to the current archive. (See **Drag And Drop Operation** for more information.)

### See Also

#### Extracting An Archive

The length of a file within an archive is its uncompressed size, or in other words, its original size before it was compressed.

The size of a file within a an archive is its compressed size; the amount of space it takes up inside the archive.



## Deleting File(s) Within An Archive

To delete files from within an archive, you must first have opened the archive. Then simply select the file or files you wish to delete from the list, and press the Delete button. ZipShell will ask you if you are sure you wish to proceed with the deletion, and if you reply "yes", ZipShell will execute the appropriate archiver to delete the file(s) from within the archive. After the archiver has finished this operation, ZipShell will re-read the archive in order to update the file list.

**NOTE:** This operation should be used with caution, since files deleted in this fashion are unrecoverable.





## Adding Files To An Archive

There are two main ways to add files to an archive. To add files to an existing archive, you must first have **opened** it.

### Method 1

You may add files to an archive using drag and drop. To do this, simply drag any number of files from the File Manager and drop them on top of ZipShell while an archive is open. ZipShell will ask you to confirm whether you wish to add the files to the archive. It will also ask you if you wish to delete the files after they have been added. After you reply to this query, you will be presented with the ARJ or ZIP compression options (if the current archive is of ZIP or ARJ format), and then ZipShell will activate the appropriate archiver to add your files to the archive. See **Drag and Drop Method** for more information.

### Method 2

Click on the Add Files button at the bottom of the screen. A dialog box will appear prompting you for the files you wish to add to your archive. On the left are the drive/directory/file boxes which enable you to navigate through your system looking for files. When you see a file you wish to add, simply double-click on it, or select it and click on the Add Selected File button. The name of the file will appear in the right-hand box, which contains names of all the files to be added. If you wish to add all the files in the currently displayed subdirectory, click on the Add All button, and a line something like c:\dos\\*. \* will appear in the list box on the right, indicating that all the files in the c:\dos directory will be added to the archive. When you are finished selecting files, click the OK button. If you are adding files to a ZIP or ARJ file, an additional window will be displayed allowing you to choose from various compression options. See **ZIP Compression Options** and **ARJ Compression Options** for more information.

When you have completed these steps, ZipShell will activate the appropriate archiver to add the files to the current archive. When the archiver is finished, ZipShell will re-read the archive in order to update its display.

## Font Command

Selecting the Font command from the Display Options window brings up a dialog box allowing you to select a font to be used in the listbox in ZipShell's main window. Only fixed-pitch fonts are shown, and the size is restricted to between 8 and 16 pts. The default font is Windows' FixedSys font.

**NOTE:** At larger point sizes with some fonts, the display may appear garbled. If this happens, either reduce the size of the font (to 10 or 12 points) or change which font you are using.

## Tab Settings

With certain display drivers, the columns in the main listbox may not be aligned with their headings. If this is the case, select the Options command from ZipShell's Setup menu, then go to the Display Options window. This will bring up a dialog box allowing you to change the tabs in the main listbox so that the columns will be aligned correctly.

The five tabs are measured in pixels. To change the current settings, type in new numbers and press OK. Make sure there are five numbers, separated by commas, and that the numbers get progressively larger, or else the display may become garbled. Try experimenting with different values until the display meets with your satisfaction. The values you input are saved in ZipShell's INI file so that once you get the numbers right, you won't have to type them in again.

You may press the Default button at any time to return the tabs to their default state. Pressing the Cancel button will abort the operation and leave the tabs unchanged.

**NOTE:** It is probably easiest to work with one tab setting at a time; i.e. work with the first until it is at the right position, move to the second, and so on. If a comma appears in the listbox it means that the tab for the item following the comma is not valid. Try increasing the value of its tab, or decreasing the value of the tab for the item preceding the comma.

## **Color Command**

Selecting this command from the Display Options window brings up a dialog box allowing you to select any of Windows' standard colors. The color you select will be applied as the background color to all list and text boxes within the program.



## Paths

The "paths" window displays a list of directory paths ZipShell uses for various operations, and allows you to edit and/or browse them. To make a set of paths active, just enter them and press OK. If you want them to be the default paths everytime ZipShell starts, select the Save button.

If you wish to select a path from the directory listing to the right (instead of typing it in manually), first select the path you wish to work with by clicking on it with the mouse. You will know a path has been selected when a black box appears around its frame. After selecting a path, just double-click on the appropriate directory in the list. Below is a description of each of the paths.

### **Extraction Path**

This is the path ZipShell uses when extracting an archive. Regardless of where the archive is located, the files residing within the archive will be extracted to this location. You may also edit the Extraction Path during the extraction process; you don't need to bring up the paths window to change the Extraction Path, only to save it as the default.

### **Open Archive Path**

When ZipShell is given the **Open Archive** command, it displays a list of archives for you to choose from. The directory it first looks in is the Open Archive Path. This is especially useful if you regularly download compressed files to a particular directory. Simply use the browse button to locate that directory, then press the Save button, and everytime ZipShell displays the open archive window, it will first display the contents of that directory.

### **Add Files Path**

The Add Files Path is simply the path that ZipShell first displays in the Add Files window (i.e. when you are adding files to an existing archive, or creating a new one).

## ZipShell Options

Selecting the Options command from ZipShell's Setup menu brings up a window which contains options and preferences which you can configure to make ZipShell work best for you. There are four different types of options:

General Options

Display Options

ZipShell Paths

Archivers



## Archivers

The Archivers window (accessed from the Options command from ZipShell's Setup menu), lets you specify location of all the archivers which ZipShell supports. Simply type in the full name and path of the archivers in their appropriate locations, and press OK to save the information to disk. Alternatively, click on one of the fields to activate it, then press the "?" key to bring up a dialog box allowing you to browse for the appropriate archiver.

Setting up these paths before using ZipShell is not necessary. If ZipShell is unable to find an archiver which it requires to complete an operation, it will prompt you for its location and save the information to disk.

**NOTE:** It is important that you accurately indicate which version of PKZIP you are using, version 1.10 or version 2.04. You select the version by checking the appropriate option button. If you indicate an incorrect version, you will receive many spurious error messages when attempting to work with ZIP files.



## Display Options

Going to the Display Options window (using the Options command from ZipShell's Setup menu), brings up a window with a "sample" listbox on the top. Use the commands available to alter the appearance of the listbox, and when you are satisfied, press the OK button to save the changes and apply them to ZipShell's main window. Below are explanations of the different options.

Color Command

Font Command

Tab Settings





## General Options

### Display Open ZIPFile Window On Startup

If this option is checked, the Open Archive Window will be displayed immediately after startup, just as if you had selected the **Open Archive** command from the **File** menu. This is for people who primarily use ZipShell to unzip programs downloaded from BBS's, etc.

### Exploding Windows

If this option is checked, every time ZipShell creates a new window, it will appear to "explode" outwards from its central point. This is an interesting special effect, but on computers with slow graphics cards, the process may be slow or annoying. In that case, you may turn it off by unchecking the option and pressing the "OK" button to save your preference to ZIPSHELL.INI.

### Disable On Startup

This option, just below the **Exploding Windows** option, is only enabled when the Exploding Windows option is checked. It allows you to disable the explosion of ZipShell's main window. All other windows will be exploded as usual. It's main purpose is for users who start ZipShell in a minimized mode (via a Program Manager group, for instance). If this box is NOT checked, and ZipShell is started in a minimized mode, the exploding effect will cover part of the screen with a gray rectangle that is more or less permanent until something else is put on top of it.

### Viewer Behavior

This is a group of three option buttons, which tell ZipShell which viewer to use when viewing files.

If the option "**Always Use Default Editor**" is checked, the editor specified in the **Default Editor** box below will always be used to view files.

If the option "**Always Use Associated File, If Any**" is checked, ZipShell will always use whatever program is associated with the type of file you are trying to view. If no program is associated with the file you are viewing, ZipShell will use the Default Editor instead.

If the option "**Prompt For Viewer To Use**" is checked, ZipShell will ask you which viewer to use each time you try to view a file, providing it can find an association for the file you are viewing. If it can not, it will use the Default Editor without asking you for confirmation.

### Default Editor

Use this section to specify which editor you wish ZipShell to launch when it unzips files for you to **view**. The default is Windows' NOTEPAD.EXE. If you would like to add a program to the list, select the **Add To List** item at the bottom of the drop-down box. A dialog box will appear allowing you to select a file to add to the editors list.

### Associate Command

Clicking on this button will cause ZipShell to edit the [Extensions] section in your WIN.INI file. After this is done, all ARJ, ARC, LZH and ZIP files are said to be "associated" with ZipShell. What this means is that when you try to run one of these archives (by double-clicking on it in the File Manager, or using the "Run" command from the Program Manager, for example), ZipShell will be launched first and will open the archive automatically and display its contents.

## **Error 108: Not enough memory (ARJ)**

This error occurs when ARJ.EXE does not have enough memory to work with in its DOS window. If you get this error repeatedly, try the following suggestions to see if they will fix the problem:

- 1) Try freeing up more conventional memory by removing TSR's (Terminate and Stay Resident programs) from your CONFIG.SYS and AUTOEXEC.BAT files. Remember, if you modify these files, you will have to reboot your computer for the changes to take effect.
- 2) If you are performing compression (i.e. adding files to an ARJ file), try using compression method -m4, which uses the least memory of the available methods.
- 3) Edit the settings in the file ARJ001.PIF (in your /windows directory) for settings more appropriate for your computer.

## **Error 107: Bad parameters (ARJ)**

This error will occur most likely for one of two reasons:

- 1) You are using ZipShell with a version of ARJ other than 2.30.
- 2) One of the files used by ZipShell has been corrupted.

If you receive this error repeatedly, contact NewVision for assistance.

## **Error 106: Can't open archive or file (ARJ)**

This error will occur in the following situations:

- 1) One of the files to be processed has been deleted, renamed, or moved so that ARJ is unable to find it.
- 2) The ARJ file you are working with is marked read-only, and you are trying to modify it by adding files or changing its comment.

## **Error 105: Disk full or write error (ARJ)**

This error will occur in the following situations:

- 1) The disk you are working with is full. This condition usually occurs when working with ARJ files on floppy disks. If this is the case, try using the -w (Use Working Drive) option (when adding or extracting files), and assign your hard drive as your working drive. This will prevent ARJ from writing its temporary files on the floppy, and thus free up more room.
- 2) You are working with an ARJ file on a floppy disk, and the disk is write-protected.

**Error 104: ARJ SECURITY error or attempt to update an ARJ-SECURED archive (ARJ)**

This error will occur when you try to update an ARJ-SECURED archive, or when the security envelope of the ARJ file you are working with has been violated.

This version of ZipShell does not support the manipulation of ARJ-SECURED archives, although you may extract files from such archives.

### **Error 103: CRC error (ARJ)**

This error occurs when ARJ detects a CRC error in the archive and most likely indicates that the archive is corrupted. If you receive this error, you may try using the "-jr: Recover Broken Files" to extract files containing errors. ARJ will attempt to extract damaged files, but may be unable to, depending on the extent of damage.

## **Error 102: Fatal error (ARJ)**

This error indicates that ARJ had a serious internal error from which it was unable to recover. If you receive this error, you should check to see that the ARJ file you were working with has not been corrupted.

This error also occurs when extracting files using the "-jr: Recover Broken Files" option. This is normal and does not indicate a problem.



## **Error 101: Warning (file not found) (ARJ)**

This error has one of two probable causes:

- 1) One of the files to be processed by ARJ could not be found (it was probably moved, deleted or renamed). Processing continues without the file.
- 2) An attempt was made to extract a garbled file without specifying a password, or specifying an incorrect password. The file is skipped, and processing continues.

## **Error 201: Warning error (Bad CRC) (PKUNZIP)**

This error occurs in the following situations:

- 1) The ZIP file you are working with is corrupted. If this is the case, reconstruct the ZIP file using PKZIPFIX.EXE to recover as much of the file as possible. Even after running PKZIPFIX, it is possible to receive this error. If you do, be cautious when extracting the files as one or more may be corrupted.
- 2) The ZIP file you are working with was created with authenticity verification information, and the file has subsequently been modified. In this case, you should be aware that any file in the ZIP file may have been tampered with and may be unsafe to use (i.e. it may contain a virus).

## **Error 202: Error in ZIP file. (PKUNZIP)**

This error message indicates a more serious error in a ZIP file than the Bad CRC message. If this occurs, PKUNZIP will be unable to process the ZIP file. Use PKZIPFIX.EXE to reconstruct the ZIP file and recover some of the data, although some will most likely be lost.

## **Error 204: Insufficient Memory (PKUNZIP)**

This error occurs when PKUNZIP.EXE does not have enough memory to work with in its DOS window. If you get this error repeatedly, try the following suggestions to see if they will fix the problem:

- 1) Try freeing up more conventional memory by removing TSR's (Terminate and Stay Resident programs) from your CONFIG.SYS and AUTOEXEC.BAT files. Remember, if you modify these files, you will have to reboot your computer for the changes to take effect.
- 2) If you are performing compression (i.e. adding files to a ZIP file), try using Imploding as your method of compression, since it uses less memory than Shrinking.
- 3) Edit the settings in the file ZIP001.PIF (in your /windows directory) for settings more appropriate for your computer.

## **Error 209: File not found (PKUNZIP)**

If you receive this error, it means that PKUNZIP was unable to find the ZIP file specified. The most likely cause is that the ZIP file was moved, deleted or renamed (with the File Manager, for example) after it had been opened by ZipShell.

## **Error 210: Bad or illegal parameters specified (PKUNZIP)**

This error will occur most likely for one of two reasons:

- 1) You are using ZipShell with a version of PKUNZIP other than 1.10 or 2.04c, or ZipShell is configured for a different version of PKUNZIP than you are using. See **Options** for information on configuring ZipShell for use with a particular version of PKUNZIP.
- 2) One of the files used by ZipShell has been corrupted.

If you receive this error, quit ZipShell, restart it, and try your operation again. If you receive this error repeatedly, contact NewVision for assistance.

## Error 211: No files found to extract, view etc. (PKUNZIP)

The most likely cause for this error is that one of the files created on the fly by ZipShell was deleted, moved or renamed by the user. If you receive this error, simply try your operation again.

This error also occurs when you try to view a scrambled file using an incorrect password.

If you receive this error again, it is possible your Zipfile is corrupted. In this case, select the **Test Zipfile Integrity** command from the ZIP menu. If you still get the same error, run PKZIPFIX to reconstruct your Zipfile.

## **Error 250: Disk Full (PKUNZIP)**

This error is caused when the target drive of any ZIP operation is full. If you receive this error, try freeing up space on the target drive, or extract files to a different disk.



## **Error 251: Unexpected EOF in ZIP file (PKUNZIP)**

This error indicates that a ZIP file terminated unexpectedly. Run PKZIPFIX.EXE to reconstruct the ZIP file to recover as much data as possible.

### **Error 301: Bad file name or file specification (PKZIP)**

This error indicates that an illegal filename or illegal wildcards were passed to PKZIP. The most probable cause of this error is that one of the files created on the fly by ZipShell was corrupted. If you receive this error, try your operation again.

## **Error 302: Error in ZIP file (PKZIP)**

This error indicates that there is a serious error in the ZIP file you are processing, and PKZIP may not have been able to complete its operation. Use PKZIPFIX.EXE to reconstruct the ZIP file as much as possible, and then try your operation again.

## **Error 304: Insufficient Memory (PKZIP)**

This error occurs when PKZIP.EXE does not have enough memory to work with in its DOS window. If you get this error repeatedly, try the following suggestions to see if they will fix the problem:

- 1) Try freeing up more conventional memory by removing TSR's (Terminate and Stay Resident programs) from your CONFIG.SYS and AUTOEXEC.BAT files. Remember, if you modify these files, you will have to reboot your computer for the changes to take effect.
- 2) If you are performing compression (i.e. adding files to a ZIP file), try using Imploding as your method of compression, since it uses less memory than Shrinking.
- 3) Edit the settings in the file ZIP002PIF (in your /windows directory) for settings more appropriate for your computer.

**Error 312: No files were found to add, or no files were specified for deletion (PKZIP)**

The most likely cause of this error is that one of the temporary files created on the fly by ZipShell was corrupted, renamed, moved or deleted. If you receive this error, try your operation again.

## **Error 313: The specified ZIP file or list file was not found (PKZIP)**

The error has one of two likely causes:

- 1) The ZIP file you are working with was deleted, renamed or moved after it had been opened by ZipShell, so that the filename listed by ZipShell as being open no longer exists.
- 2) One of the temporary files created by ZipShell on the fly (in this case, the list file) was deleted, renamed or moved before PKZIP could complete its operation. If this is the case, try your operation again.

### **Error 314: Disk full (PKZIP)**

This error occurs when the target drive of a ZIP operation is full. If you are modifying an existing ZIP file in any way, try using the -b (Store Temp Zipfile on Drive) option and assign a local hard drive as the working drive. This will free up space on floppies during compression operations.

### **Error 315: ZIP file is read-only and cannot be modified (PKZIP)**

This error indicates that you tried to modify a ZIP file marked as read-only. If you receive this error, remove the read-only attribute from the ZIP file and try your operation again.



## **Error 316: Bad or illegal parameters specified (PKZIP)**

This error will occur most likely for one of two reasons:

- 1) You are using ZipShell with a version of PKUNZIP other than 1.10 or 2.04c, or ZipShell is configured for a different version of PKUNZIP than you are using. See **Options** for information on configuring ZipShell for use with a particular version of PKUNZIP.
- 2) One of the files used by ZipShell has been corrupted.

If you receive this error, quit ZipShell, restart it, and try your operation again. If you receive this error repeatedly, contact NewVision for assistance.

### **Error 317: Too many files (PKZIP)**

This error message indicates that you tried to add or delete too many files at once. If you receive this error, try your operation again, this time specifying fewer files for deletion or compression.

### **Error 318: Unable to open specified file(s) (PKZIP)**

This error indicates that PKZIP was unable to open a specified file. It is a warning condition only; PKZIP continues to process the ZIP file even if you receive this message.

A possible cause of this error is that one of the files you had specified to add to a ZIP file was deleted, moved or renamed before it could be processed by PKZIP.

## **Error 401: CRC Error**

This error indicates that the LZH file in use contains internal errors and may be unreliable. Be wary of any files extracted from an archive which produces this error, as they may contain serious errors.

## **Error 402: Fatal Error**

This error indicates that LHA was unable to process the requested command due to some unresolvable error. The original LZH file is left unchanged. Possible sources of this error:

- 1) User pressed CTRL-C during processing
- 2) No files were found to add, extract, delete, etc.

### **Error 403: Unable to write temporary files to archive**

This error occurs when LHA is unable to write a temporary file to the archive in use. Possible reasons for this error: the LZH file may be marked read-only, or the disk the LZH file is on may be full. If you receive this error, there may be a file called LHTEMP)2(.LZH on your hard drive. You may rename this to an LZH file and use it as an LZH archive.

## ZipShell Error Index

Click on any of the following errors to see a full explanation of it and possible solutions.

### **ARJ Errors**

[Error 101: Warning \(file not found\)](#)  
[Error 102: Fata error](#)  
[Error 103: CRC error](#)  
[Error 104: ARJ-SECURITY error](#)  
[Error 105: Disk full or write error](#)  
[Error 106: Can't open archive or file](#)  
[Error 107: Bad parameters](#)  
[Error 108: Not enough memory](#)

### **PKUNZIP Errors**

[Error 201: Warning error \(Bad CRC\)](#)  
[Error 202: Error in ZIP file](#)  
[Error 204: Insufficient Memory](#)  
[Error 209: File not found](#)  
[Error 210: Bad or illegal parameters specified](#)  
[Error 211: No files found to extract, view etc.](#)  
[Error 250: Disk full](#)  
[Error 251: Unexpected EOF in ZIP file](#)

### **PKZIP Errors**

[Error 301: Bad file name or file specification](#)  
[Error 302: Error in ZIP file](#)  
[Error 304: Insufficient Memory](#)  
[Error 312: No files were found to add or specified for deletion](#)  
[Error 313: The specified ZIP file or list file was not found](#)  
[Error 314: Disk full](#)  
[Error 315: ZIP file is read-only and cannot be modified](#)  
[Error 316: Bad or illegal parameters specified](#)  
[Error 317: Too many files](#)  
[Error 318: Unable to open specified file\(s\)](#)

### **LHA Errors**

[Error 401: CRC Error](#)  
[Error 402: Fatal Error](#)  
[Error 403: Failed to write temporary files in archive](#)





