Soroban Support Guide

Soroban Support

PeaZip installation and using for sharing data

PeaZip is a useful, and free tool, which can assist in sharing data over the Internet for example using the Web service WeTransfer. It significantly extends the features provided by the inbuilt Zip functionality now available on Operating Systems to make sharing data over the Internet safer and easier to manage.

Safe sharing over a public service implies that the data must be encrypted.

Large files, even when compressed using Zip tools, may be too large to share with services such as WeTransfer there is a limit on the free service of 2 Gbyte for a simple transfer. Splitting the file into smaller chunks eliminates this constraint.

PeaZip can compress the data in a similar manner to the familiar Zip programs that are available on most platforms. It in addition provides options to:

- Encrypt the data before uploading to a public server (**which is essential if the data being shared contains any private information**) and to to decrypt when received by the Recipient
- Split an archive file, that is too large so be sent as one file, into chunks that can be shared using WeTransfer and then recombined by the Recipient

PeaZip is available for Windows, MAC and Linux platforms.

Original Author:	**John Steele**
Revised by:	**John Steele**
Version:	**Draft**
Date:	**10 Sep 2023**

Copyright Notice

Revisions

Version	Date	Changed by	Summary of change
Draft	10 Sep 2023	John Steele	Initial draft

Table of Contents

1 PeaZip Introduction 4
1.1 What is a Zip file? 4
1.2 What is PeaZip 4
1.3 Scope of this document 4
2 Usage – Overview 6
2.1 Small archive file 6
2.1.1 Basic use to prepare files to be sent to another user 6
2.1.2 Simple Zip operation 6
2.1.3 Simple Zip + encryption 6
2.1.4 Simple file transfer – WeTransfer steps 6
2.2 Large archive file 7
2.2.1 Overview 7
2.2.2 Zip file – split into several chunks 8
3 PeaZip – Installation 10
3.1 Installation 10
3.1.1 Overview 10
3.1.2 Download and Install PeaZip 10
3.2 Notes on screenshots 13
4 Using PeaZip to create an archive 14
4.1 Overview 14
4.2 Preparing the data for uploading 14
4.2.1 Initial view 14
4.3 Choosing the right options 16
4.3.1 Annotation 1 – preparing files to be archived 16
4.3.2 Annotation 2 – configuring where the archive will be created 18

4.3.3 Annotation 3 – Clicking on the Set password / Keyfile 18
4.3.4 Annotation 4 19
4.3.5 Annotation 5 – type of compression 19
4.3.6 Annotation 6 – New or existing archive 19
4.3.7 Annotation 7 – file splitting choices 20
4.3.8 On completion 20
5 On receipt of the files 21
5.1 Archive files 21

Table of Figures

Figure 1: PeaZip in Start Menu.....12
Figure 2: Modified PeaZip column selection used for screen capture.....13
Figure 3: PeaZip - Initial default view.....15
Figure 4: PeaZip - Prepare files to be archived prior to transmission.....15
Figure 5: Right click menu.....16
Figure 6: Selecting files.....17
Figure 7: Added files.....17
Figure 8: Adding two folders.....18
Figure 9: Apply security to archive.....18
Figure 10: Compression options.....19
Figure 11: Standard file splitting options.....20
Figure 12: Custom size.....20
Figure 13: Received files as chunks.....21
Figure 14: PeaZip has identified the Zip file and shows its content.....22
Figure 15: Extract view.....22
Figure 16: Choose folder to receive the extracted files.....23
Figure 17: Create folder to receive the files.....23
Figure 18: File/folder as received.....24

Index of Tables

Table 1: Sharing data – small archive file.....7
Table 2: Sharing data that has been divided into Chunks.....9
Table 3: PeaZip installation.....10

1 PEAZIP INTRODUCTION

1.1 What is a Zip file?

The term "Zip", when related to computer files, is now almost a generic term that refers to a single computer file that can contain one of more files and/or folders my merging each of these into a single file.

While combining the files/folders it also "compresses" them so that the resulting file is generally smaller than the sum of the files/folders being consolidated into the single file. The amount of file size reduction does depend on what files are being "zipped". In some cases the size reduction is minimal, in others it is dramatic. It depends on the format of the data being compressed.

The Zip format has become a common standard and it is natively supported by Windows and MAC Operating systems for example. This form of compression is called "lossless" as not information is lost. The data, when decompressed, is identical to the original.

This is distinct from other forms of compression, often used for images and video, where the image resolution is reduced but this does result in data loss and this process is not reversible!

Zip is not the only standard compression method and several others also exist. Examples include 7zip, RAR and many others but Zip is the most common. PeaZip does also support other compression methods but the use of the Zip method is assumed in this document.

1.2 What is PeaZip

PeaZip is a free tool that not only provides file compression tools but also provides additional features that provides greatly exceed the functionality to that provided by the native Windows and MAC inbuilt standard ZIP file compassion. It is the only such tool known to the author that provides versions that can run on a variety of platforms and in particular on:

- Windows
- Apple MAC
- Linux

It can also process many of the alternative compression formats n addition to Zip although these are not described in this document.

See the PeaZip website https://peazip.github.io/index.html for full details.

1.3 Scope of this document

This guide does not attempt to replace the detailed documentation that is provided on the PeaZip web site.

PeaZip is actually an easy program to use in practice but it may appear to be complex to use.

This guide explains what features exist in PeaZip that are useful for file sharing and why. This s probably the largest section.

*This guide then attempts to explain **exactly** how to use the features that assist in sharing data over the Internet.*

Each of these steps are explained in detail using screen shots to assist the explanation.

The intention/hope is that you should not need a computer expert to be able to share data. If you work through the steps once you will discover it is actually very easy and quite intuitive!

***This guide is NOT intended to be a detailed and comprehensive guide to all the extensive features of PeaZip!** See the PeaZip web site for additional information.*

This guide will therefore cover the following:

- An overview of the options that can be used to exchange data over the Internet and specifically those needed for using the WeTransfer web based service.
- Installation of PeaZip (on Windows systems) for use for data sharing
    - For installation on MAC or Linux it is hoped that this document will act as a guide.
        - Note: If someone is able to provide the author with specific details and screen shots for other platforms then this guide could be updated.

Specifically this guide will describe **in detail** how to use PeaZip to create a Zip format compressed archive file from a set of files and/or folders on your computer to prepare them for sharing via WeTransfer. The features in PeaZip that are particularly useful include:

- Consolidate the folders/files into a single Zip file in a similar way to the inbuilt Zip functionality provided with the native operating system. **Where this differs from the native Zip functionality is that it also provides the following enhancements:**
    - Ability to optionally encrypt the zip file so that the data can be passed to another user over an insecure path e.g. the Internet if the source contains any private information.
        - This is **essential** if the data is to be passed via an insecure route such as an email attachment or a public file sharing service such as WeTransfer
        - **This feature uses an encryption password/key (a text string) that must be shared with the other person in a safe manner e.g. via a telephone call. It must NEVER be shared via the same route as the data it is protecting**
    - The ability to split the Zip file I creates into smaller parts referred to as “chunks” typically the Zip file created prior to sharing over a service such as WeTransfer and then recombining the chunks when received by the Recipient
        - This specifically enables larger data sets to be shared using services such as WeTransfer or email where there is a limit on the size of individual transfers
            - Note: WeTransfer is limited to a maximum of 2 Gbyte chunks for the free version (Use of this service is described in a separate guide).
                - It will usually be more efficient to use smaller chunks so that the data transfers can be overlapped
            - Email typically has a limit of 10 Mbytes per attachment but this may vary with email providers

The steps for all of these features are described, together with screen shots, in some detail. The process is actually quite simple despite the length of this document!

2 USAGE – OVERVIEW

2.1 Small archive file

2.1.1 Basic use to prepare files to be sent to another user

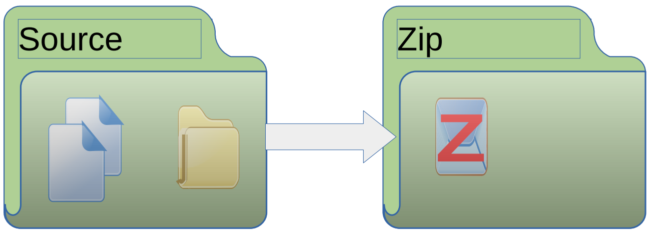
PeaZip can create a simple archive file. Many different archive file formats are supported but in this guide we will only describe the use of the well known Zip format. It is claimed that this is compatible with other Zip programs including the Zip processing built into Windows and MAC.

PeaZip however can do far more but the basic Zip format is a good place to start. The following sections show a number of possible encoding scenarios.

2.1.2 Simple Zip operation

Any number of files and folders are compressed and then consolidated into a single Zip file.

The amount of compression depends on the options chosen and on the nature of the source data.

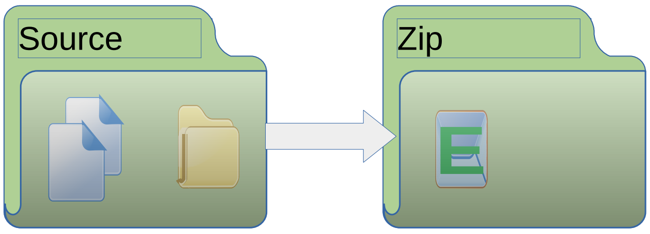


Several compression options are available when the Zip file is created A high degree of compression takes more time to create the zip file but could reduce the time to send/receive the Zip file over the Internet.

Some files are already compressed and there will be little difference between the uncompressed and maximum compression.

2.1.3 Simple Zip + encryption

The process is the same as a simple Zip operation but the Zip file is encrypted using a password.



This file can only be decrypted using the same password which must obviously be transferred to the recipient of the file in an independent, but sufficiently secure, manner. Examples would include

- Telephone call
- Text message
- Etc.

2.1.4 Simple file transfer – WeTransfer steps

The Zip file can be sent unencrypted or encrypted depending on the privacy requirements.

It must be small enough to be sent by the chosen method of delivery . For WeTransfer this must be less than 2 Gbytes.

The steps here are very simple and are shown in Table 1 below.

Table 1: Sharing data – small archive file

Step	Sender	Recipient
1.	Upload the archive file to WeTransfer and send an email to Recipient with the download link provided by WeTransfer. This assumes that the file is less than 2Gbytes and there are no other files stored on WeTransfer	Receive the download link by email and imitate the download to their computer Note that the WeTransfer link is time limited!
2.	Wait to receive notification that the Receiver has received the file, or monitor the WeTransfer status which shows that the recipient has downloaded the file.	Notify the Sender when the download has been completed
3.	Once the Sender receives confirmation that the recipient has successfully received the file it can be deleted on the WeTransfer server	

2.2 Large archive file

2.2.1 Overview

A larger Zip file, which can optionally be encrypted, may be too large and will need to be split up into chunks that can be shared over WeTransfer or email. Each chunk ha to be uploaded separately and must be separately notified to the recipient.

Assuming that your WeTransfer online file store is initially empty you have two basic choices:

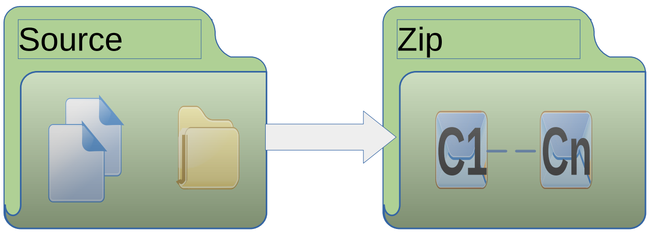
1. Divide the archive file into chunk sizes that match the WeTransfer limit of 2 Gbyte
   a) With the this option the first chunk is uploaded to WeTransfer, the recipient is notified, the recipient downloads the chunk and informs the Sender. The sender then deletes the first chunk and uploads the next chunk.
2. Divide the archive file into chunks of half the WeTransfer limit i.e. 1 Gbyte or less
   a) With this option there is more interaction between Sender and Recipient but the data exchange is better optimised.
      i The Sender can be uploading the next chunk while the receiver is downloading a previous chunk effectively halving the time taken to exchange the data
      ii There is however more interaction required by both Sender and Recipient

Either approach will result in the Recipient having an identical archive file that can be extracted to a mirror copy of the Sender's data. Option 2 requires twice the number of interactions between the participants as there a twice as many files to send but the *sending/receiving can be overlapped so that potentially the transfer can be done in half the time!*

Your Choice!

2.2.2 Zip file – split into several chunks

With this option the, possibly encrypted, Zip file is split into several equal sized "chunks". The file extension for each file is .001, .002, etc. These are shown as C1 to Cn on the diagram. The final chunk may be smaller!



The chunk size is configurable to enable it to be sent using several steps. This approach is appropriate when there is a size limit for individual files e.g. by email or by a web based file transfer service such as WeTransfer.

Assuming you are using WeTransfer each chunk will typically be sent as a separate transfer over to WeTransfer. This enables you to optimise the time it takes to send a large file set. See the WeTransfer guide for more details.

Hint: The zip file chunks each have a numeric file extension. In practice it does not matter what order the chunks are sent. It is suggested that:

- ➢ The chunks sizes are set to half or a third of the transfer limit. This limit is typically 2 Gbyte.
- ➢ The chunks are sent in reverse order – largest fie extension number first.

Manual intervention would be required if you are using a service such as WeTransfer to exchange a large file set if the data exceeds the 2 Gbytes data limit on their free service..

Each chunk would need to be passed to the recipient as a separate transfer and then deleted from WeTransfer server server before the next chunk can be uploaded.

If the archive file is larger than the 2 Gbyte limit here is a suggested sequence of steps to follow:

- ➢ Create (optionally encrypted) and archive file and split it into chunks. These will have file extensions .000, .001,...nnn.
  - ◆ The chunk size should be half of the WeTransfer size limit (or smaller)
  - ◆ These will be equal size apart from possibly the final one which may be smaller. Assume that the chunk size is half of the WeTransfer
  - ◆ Proceed according to Table 2 below

This suggestion of using half the available WeTransfer space for each chunk enables the chunk uploads to the server and the chunk downloads from the server to be overlapped. This can reduce the total time taken but requires twice as many steps at each end. The choice is yours!

Starting the download from the highest numbered chunk means that the resipient will know when all the chunks have been received.

Once all chunks have been downloaded, demerged, decrypted if necessary, and expanded, the Recipient will have an identical copy of the files that were on the Sender's computer. The time stamps on each file is retained but not for any folders.

Table 2: Sharing data that has been divided into Chunks

Step	Sender	Recipient
1.	Upload the last file chunk to WeTransfer and send email to Recipient with the download link	Receive the download link by email and imitate the download to their computer
2.	While the transfer is being read (or even before) upload the next lower numbered chunk to WeTransfer and send the new link by email.	Use the link to initiate the download of the chunk to a local disk folder.
3.	Monitor the WeTransfer web site to watch for completed download, or wait for an email from the Recipient	When the download of the chunk has completed inform the Sender e.g. by email
4.	When you know the first chunk has been downloaded (either by receiving an email of by monitoring WeTransfer status) delete the first chunk and initiate upload of the next chunk in the sequence – the chunk number immediately before that uses in step 2	Wait for the next email and repeat from Step 3
5.	Repeat from Step 3 until all chunks have been sent	Once all chunks have been received confirm to the sender. Use PeaZip to recombine the archive file and unencrypt if appropriate. The receiver will then have an identical file set to the senders original data

3 PEAZIP – INSTALLATION

3.1 Installation

3.1.1 Overview

PeaZip should be installed at both sender and receiver if being used for exchanging data over the Internet using encryption and/or archive file splitting. It is not needed if the data does not need encryption or is small enough to be sent as a single transfer.

A "portable" version is available that has not yet been evaluated. If anyone wants to experiment and provide brief instructions that mode can be added to a future version of this guide.

The author does not have access to other Operating Systems but if screenshots can be obtained from other Operating Systems, and especially if there are significant differences, then these can be added.

Installation of PeaZip is quite straightforward. The detailed installation instructions here apply to Windows systems.

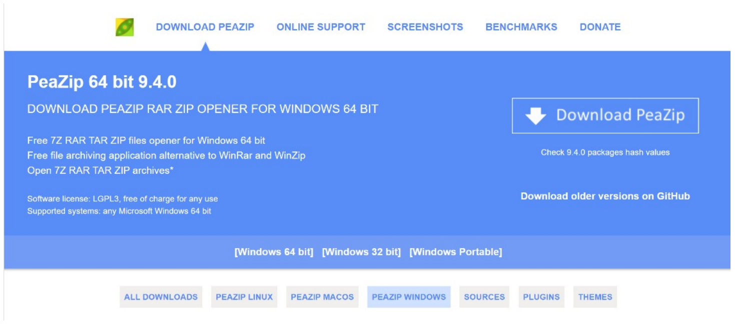
3.1.2 Download and Install PeaZip

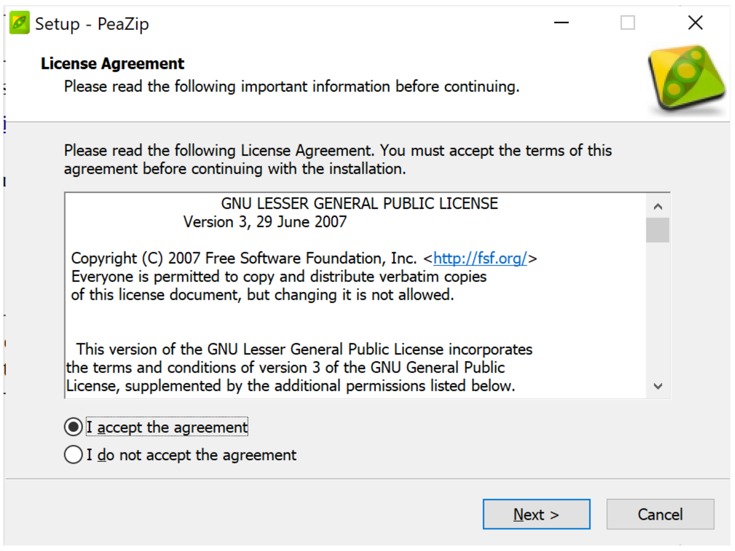
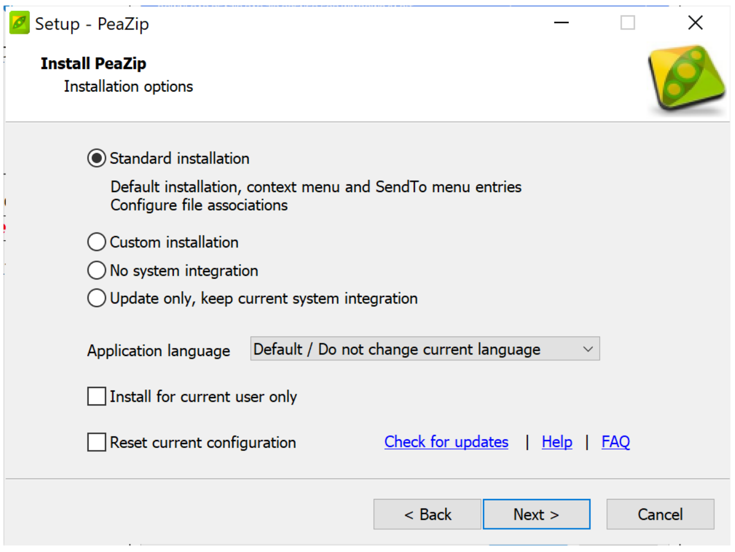
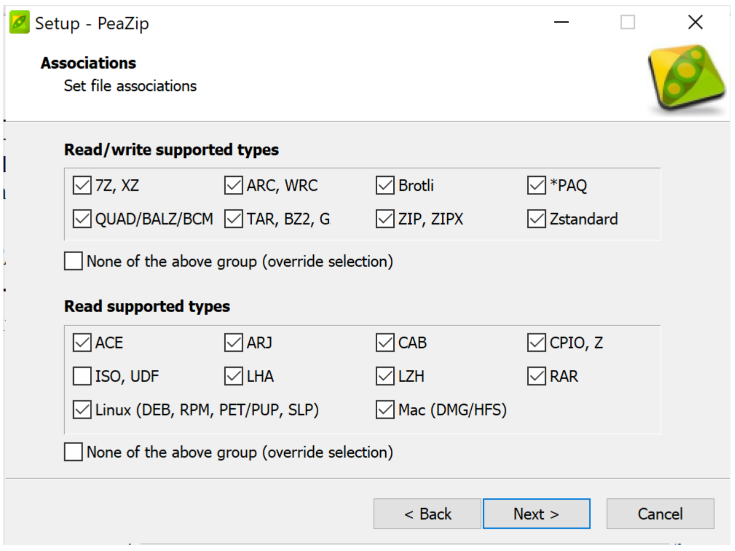
PeaZip should be downloaded from the PeaZip website. Here is the generic download page where you can choose the download option you require.

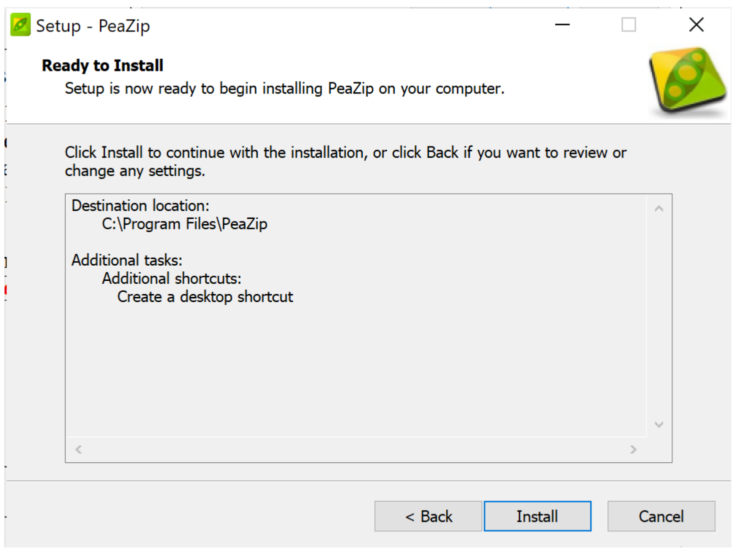
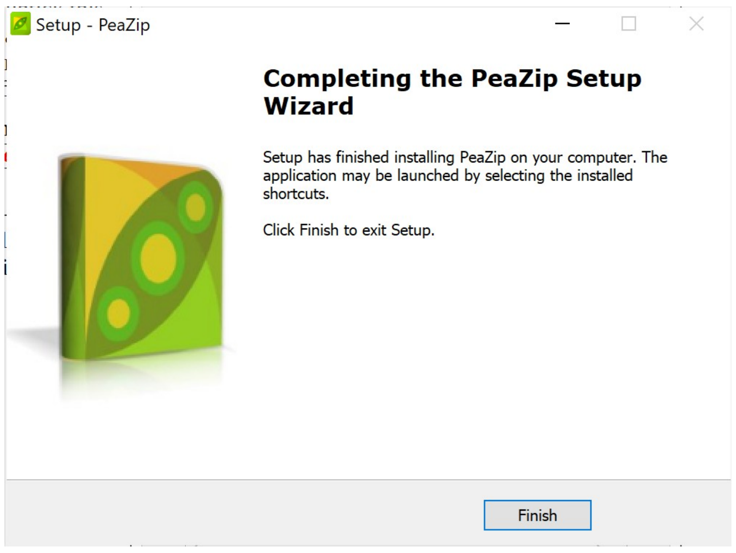
- https://peazip.github.io/index.html

Installation for Windows is shown in the following table!

Table 3: PeaZip installation

Comments	Screen shots
Download and Installation https://peazip.github.io/peazip-64bit.html Select download option, in this case Windows was selected.	 DOWNLOAD PEAZIP ONLINE SUPPORT SCREENSHOTS BENCHMARKS DONATE **PeaZip 64 bit 9.4.0** DOWNLOAD PEAZIP RAR ZIP OPENER FOR WINDOWS 64 BIT Free 7Z RAR TAR ZIP files opener for Windows 64 bit Free file archiving application alternative to WinRar and WinZip Open 7Z RAR TAR ZIP archives* Software license: LGPL3, free of charge for any use Supported systems: any Microsoft Windows 64 bit Download PeaZip Check 9.4.0 packages hash values Download older versions on GitHub [Windows 64 bit] [Windows 32 bit] [Windows Portable] ALL DOWNLOADS PEAZIP LINUX PEAZIP MACOS PEAZIP WINDOWS SOURCES PLUGINS THEMES

Review the license agreement and select ***I accept..*** option Click ***Next*** button to continue	 Setup - PeaZip **License Agreement** Please read the following important information before continuing. Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation. GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. [✓ I accept the agreement] I do not accept the agreement
The default installation options are normally sufficient. Review Help or FAQ for more information. Click ***Next*** button to continue	 Setup - PeaZip **Install PeaZip** Installation options [✓ Standard installation] Default installation, context menu and SendTo menu entries Configure file associations Custom installation No system integration Update only, keep current system integration Application language: Default / Do not change current language ☐ Install for current user only ☐ Reset current configuration Check for updates | Help | FAQ < Back Next > Cancel
Review the file associations. If you leave the ZIP/ZIPX option selected then this program will replace the standard inbuilt ZIP functionality. I would probably untick this box to leave the native Operating function but it makes no real difference. When you have made your choice the click ***Next***	 Setup - PeaZip **Associations** Set file associations **Read/write supported types** ☑ 7Z, XZ ☑ ARC, WRC ☑ Brotli ☑ *PAQ ☑ QUAD/BALZ/BCM ☑ TAR, BZ2, G ☑ ZIP, ZIPX ☑ Zstandard ☐ None of the above group (override selection) **Read supported types** ☑ ACE ☑ ARJ ☑ CAB ☑ CPIO, Z ☐ ISO, UDF ☑ LHA ☑ LZH ☑ RAR ☑ Linux (DEB, RPM, PET/PUP, SLP) ☑ Mac (DMG/HFS) ☐ None of the above group (override selection)

Review the installation and click ***Install*** to install PeaZip.	 Setup - PeaZip **Ready to Install** Setup is now ready to begin installing PeaZip on your computer. Click Install to continue with the installation, or click Back if you want to review or change any settings. Destination location: C:\Program Files\PeaZip Additional tasks: Additional shortcuts: Create a desktop shortcut
Almost there. Click ***Finish*** to exit the installer	 Setup - PeaZip **Completing the PeaZip Setup Wizard** Setup has finished installing PeaZip on your computer. The application may be launched by selecting the installed shortcuts. Click Finish to exit Setup.

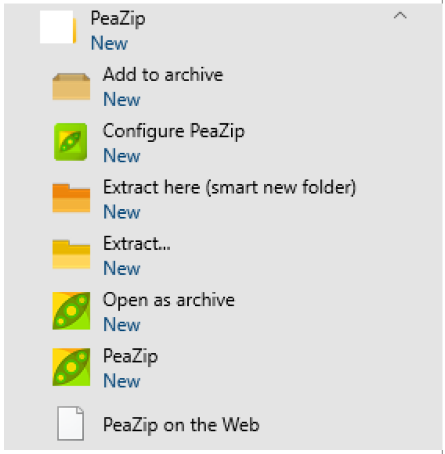


Figure 1: PeaZip in Start Menu

You should now have PeaZip in your Start Menu with a whole list of options! See Figure 1.

Note that the “New” is a Windows feature showing this has recently been installed. It will eventually go away!

The one we will be using is ***PeaZip***.

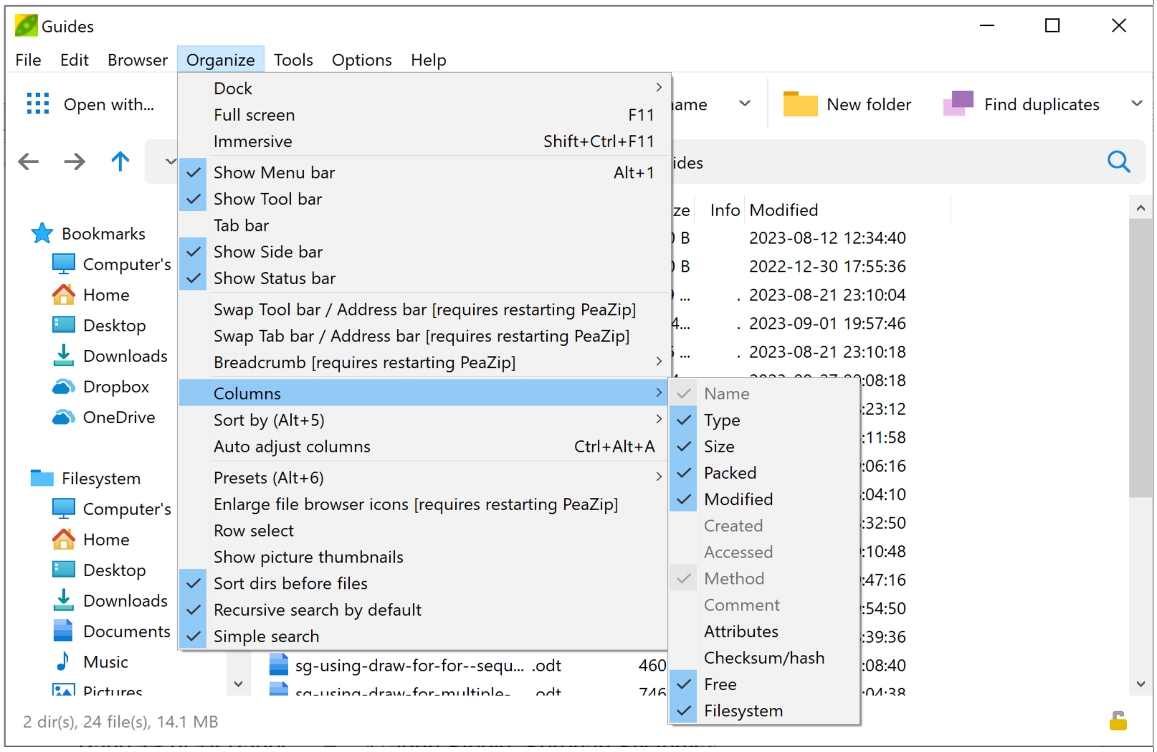
When this is selected PeaZip will open with a default layout.

3.2 Notes on screenshots

In the screenshots, to reduce the size of the captured screenshots in this document, the number of columns has been changed from the default installation as shown in Figure 2 below.

This has no effect on the behaviour of the program.

It is an example of the flexibility of this program to adjust many aspects to suit your personal preferences.



- Dock
- Full screen F11
- Immersive Shift+Ctrl+F11
- ✓ Show Menu bar Alt+1
- ✓ Show Tool bar
- Tab bar
- ✓ Show Side bar
- ✓ Show Status bar
- Swap Tool bar / Address bar [requires restarting PeaZip]
- Swap Tab bar / Address bar [requires restarting PeaZip]
- Breadcrumb [requires restarting PeaZip]
- Columns
  - ✓ Name
  - ✓ Type
  - ✓ Size
  - ✓ Packed
  - ✓ Modified
  - Created
  - Accessed
  - ✓ Method
  - Comment
  - Attributes
  - Checksum/hash
  - ✓ Free
  - ✓ Filesystem
- Sort by (Alt+5)
- Auto adjust columns Ctrl+Alt+A
- Presets (Alt+6)
- Enlarge file browser icons [requires restarting PeaZip]
- Row select
- Show picture thumbnails
- ✓ Sort dirs before files
- ✓ Recursive search by default
- ✓ Simple search

2 dir(s), 24 file(s), 14.1 MB

Figure 2: Modified PeaZip column selection used for screen capture

4 USING PEAZIP TO CREATE AN ARCHIVE

4.1 Overview

In this section we will look at the various options associated with creating an archive that can be shared with another person. This description is targeted at the situation where a set of files is to be shared via the WeTransfer service over the Internet. Other uses could be envisaged.

An outline of the steps is as follows:

1. The data to be transferred must be consolidated into a single archive file to make the transfer easy to manage
   a) This can include multiple files, multiple folders, or a combination of both
2. If the size of this resulting archive file is larger than the file size limit for the method of file sharing method chosen then it must be fragmented into chunks that each meet the criteria imposed by the file sharing method. For example if using WeTransfer
   a) The file chunk size limit (assuming the free version is used) is 2 Gbyte
   b) Each chunk must be sent as a single transfer over the Internet and for large files the sender will have to either:
      i Wait until the receiver has copied the chunk from the server
         - WeTransfer web access status shows this has happened but you could also ask the Receiver to notify you by email.
      ii The sender must then delete the chunk to make space
      iii Upload the next chunk and inform the receiver that the next chunk is ready.
      iv Note that while does not actually matter what order the chunks are sent it is suggested that the chunks are send from the largest numbered chunk with the smallest chunk number (001) sent last. This helps the receiver to know when all have been received
   c) OR
      i It would be more efficient, for very large archives, for the sender:
         - to split the archive file into chunks that are half the maximum size or less than the maximum that can be handled by WeTransfer i.e. 1 Gbyte
         - upload the a chunk and notify the receiver who can then start to download
         - the sender can immediately upload the next chunk while the first is being downloaded so that it is ready to start downloading typically immediately
         - Wait until a chunk has been received and continue sending chunks until all have been sent
         - This sounds more complicated than it is!

4.2 Preparing the data for uploading

4.2.1 Initial view

When you initially run the PeaZip the initial display will look something like Figure 3 below. In all cases the screen shots are from a Windows 10 computer.

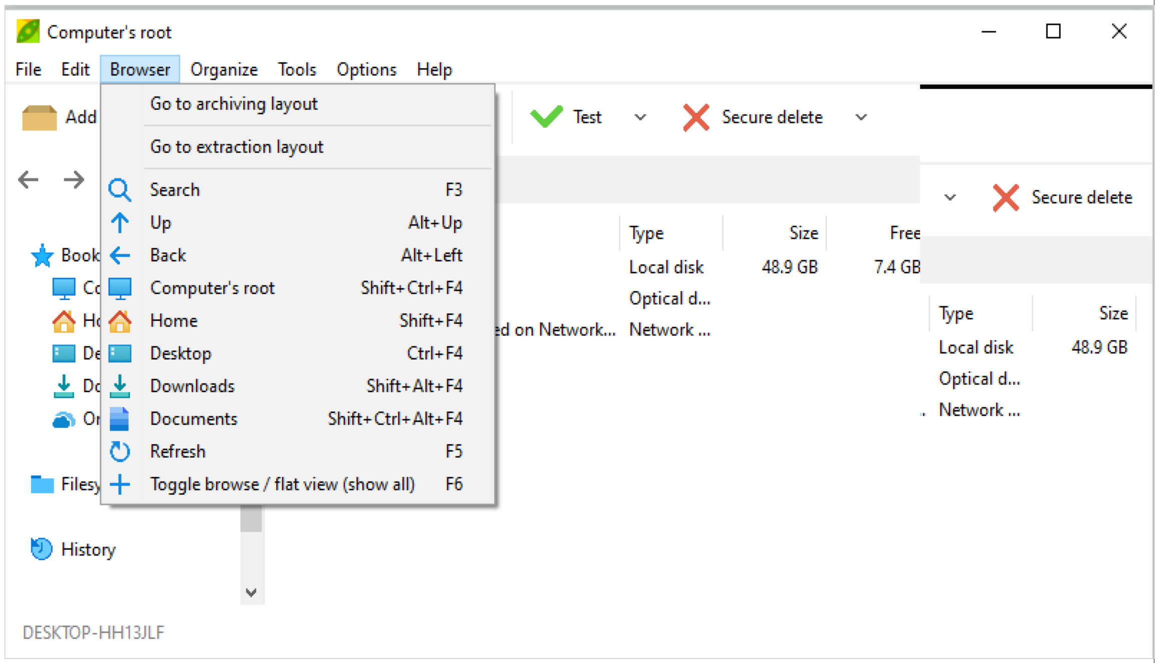


Figure 3: PeaZip - Initial default view

To get to the view we need select **Browser** and the **Go to archiving layout**.

Figure 4 below shows the most useful view for PeaZip for preparing the files.

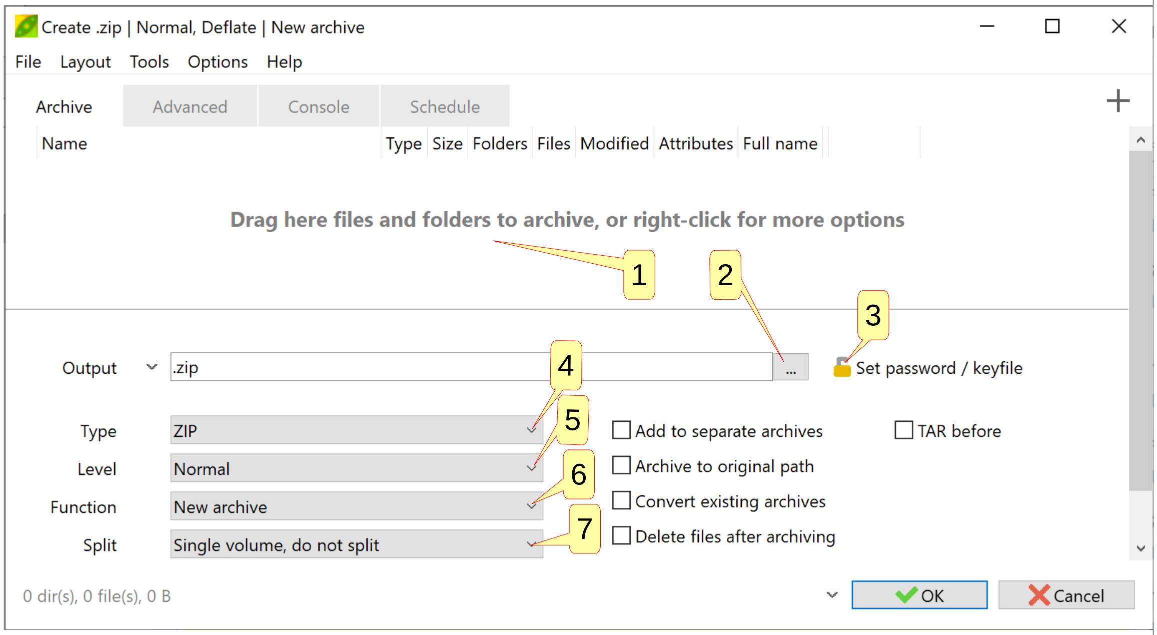

Create .zip | Normal, Deflate | New archive
File Layout Tools Options Help
Archive Advanced Console Schedule
Name Type Size Folders Files Modified Attributes Full name
Drag here files and folders to archive, or right-click for more options
1
2
3
4
5
6
7
Output .zip
Set password / keyfile
Type ZIP
Level Normal
Function New archive
Split Single volume, do not split
Add to separate archives
TAR before
Archive to original path
Convert existing archives
Delete files after archiving
0 dir(s), 0 file(s), 0 B
OK
Cancel

Figure 4: PeaZip - Prepare files to be archived prior to transmission

The numbered annotations have been added to explain the more important items you may need to use and are described below.

To simplify the description that follows it is assumed that if multiple files/folder are to be sent they are already in the same source folder. PeaZip does not impose this constraint!

Once you have selected all the relevant options then PeaZip will prepare a local zip file, or a set of zip file chunks.

The following sections go through the essential options that will be required to prepare an archive for sending over a transfer service such as WeTransfer.

4.3 Choosing the right options

4.3.1 Annotation 1 – preparing files to be archived

The area pointed to by annotation 1 is where the files that will be archived are listed.

You can "drag and drop" files into this area, or you can right click in the file area where you will see a number of options to select files to be included.

- Add files
- Add folder
- Add files and folders
- Plus several more

There are many options here but as an example see Figure 5: Right click menu below.

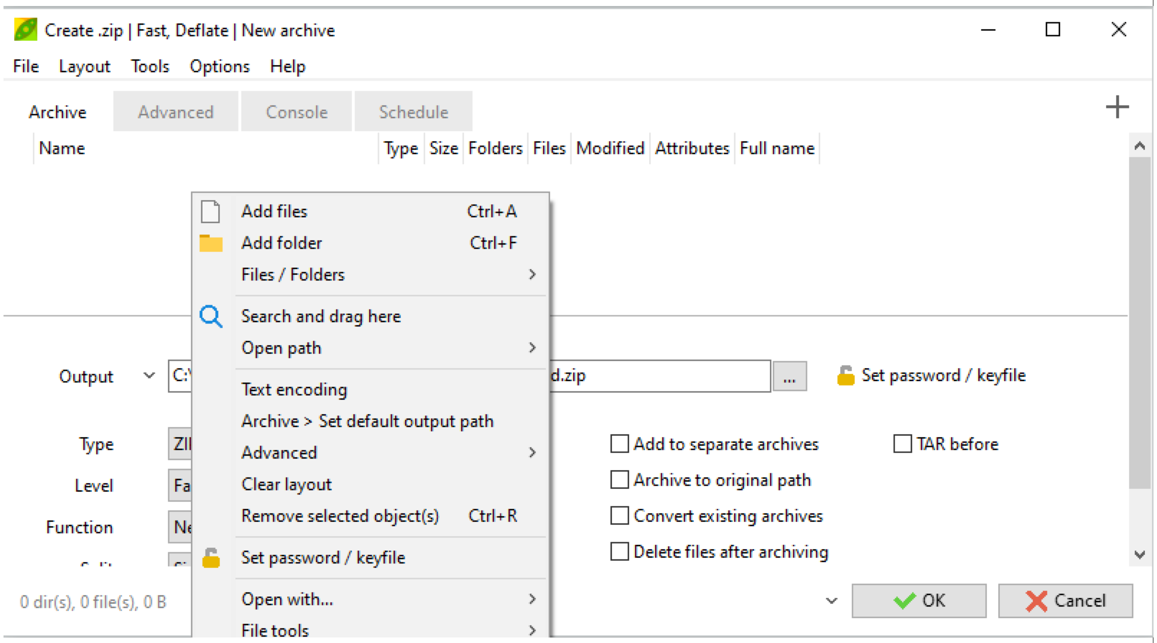
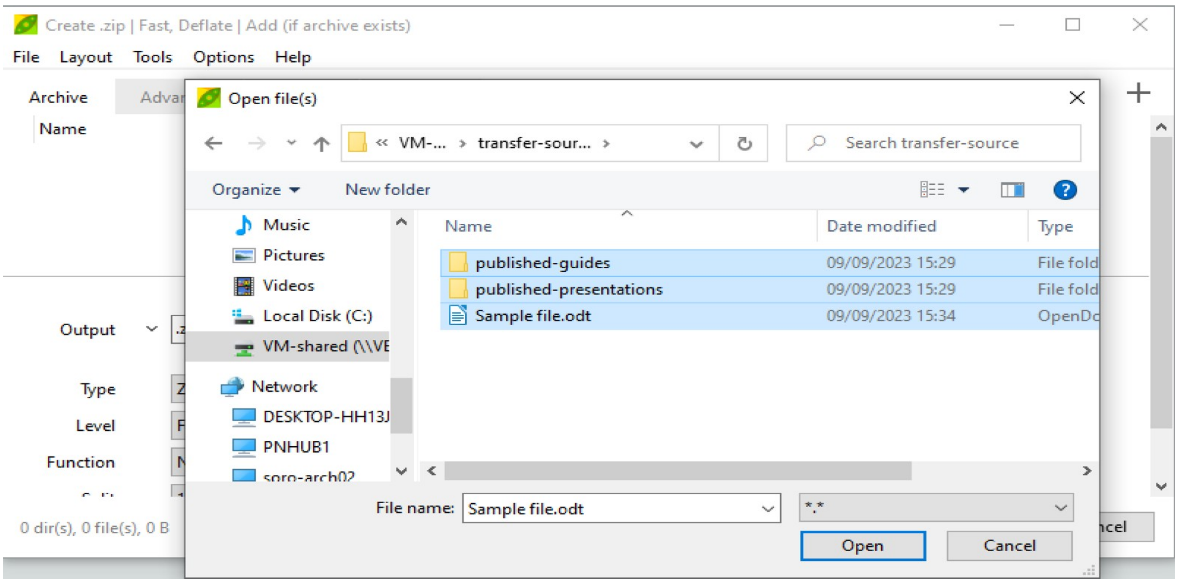


Figure 5: Right click menu

When you select **`Add files`** or **`Add Folder`** an explorer view of your filing system will appear and you can navigate to the area you want to share. This is illustrated in Figure 6: Selecting files.below.



Create .zip | Fast, Deflate | Add (if archive exists)

File Layout Tools Options Help

Open file(s)

« VM-... › transfer-sour... ›

Search transfer-source

Organize ▾ New folder

Name	Date modified	Type
published-guides	09/09/2023 15:29	File fold
published-presentations	09/09/2023 15:29	File fold
Sample file.odt	09/09/2023 15:34	OpenDo

File name: Sample file.odt

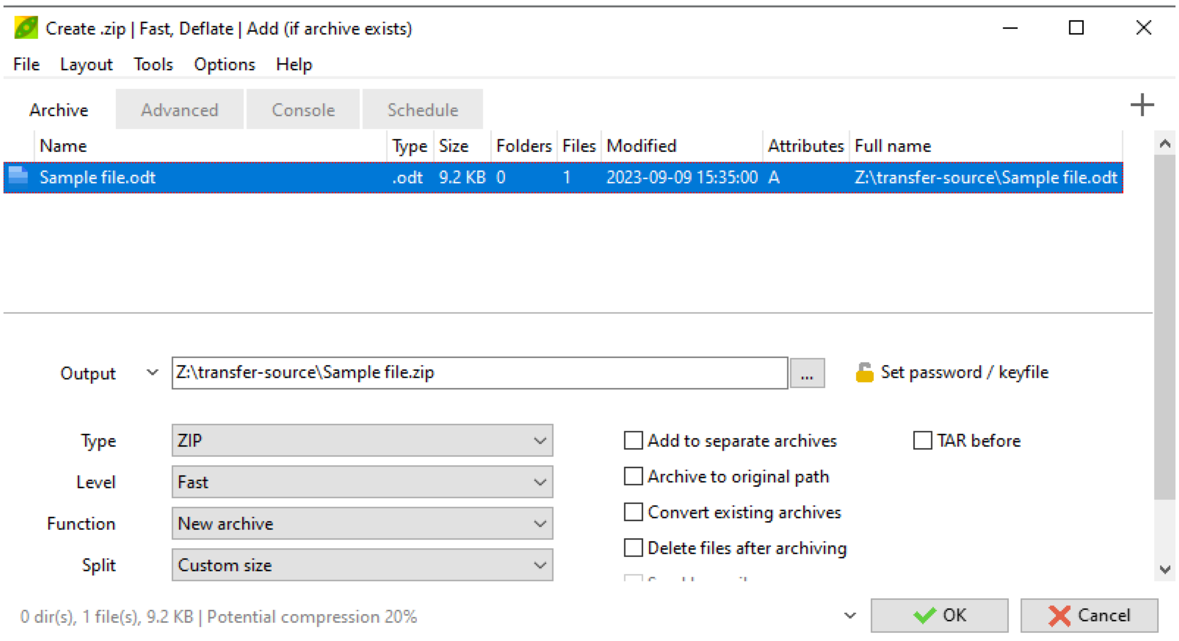
.

Open Cancel

Figure 6: Selecting files

You add multiple files by holding down the CTRL key. In this case two folder and a file are selected which will result in an update to the archive area as shown in Figure 7: Added files. Click on Open to add the files to the PeaZip file area.

Note that although two folders were **apparently** selected in Figure 6 this has not been included in the list of files. To add folders you need to select the `Add folder` option in which case any files that are NOT contained in folders will be ignored.



Create .zip | Fast, Deflate | Add (if archive exists)

File Layout Tools Options Help

Archive Advanced Console Schedule

Name	Type	Size	Folders	Files	Modified	Attributes	Full name
Sample file.odt	.odt	9.2 KB	0	1	2023-09-09 15:35:00	A	Z:\transfer-source\Sample file.odt

Output: Z:\transfer-source\Sample file.zip

Set password / keyfile

Type: ZIP

Level: Fast

Function: New archive

Split: Custom size

☐ Add to separate archives
☐ TAR before
☐ Archive to original path
☐ Convert existing archives
☐ Delete files after archiving

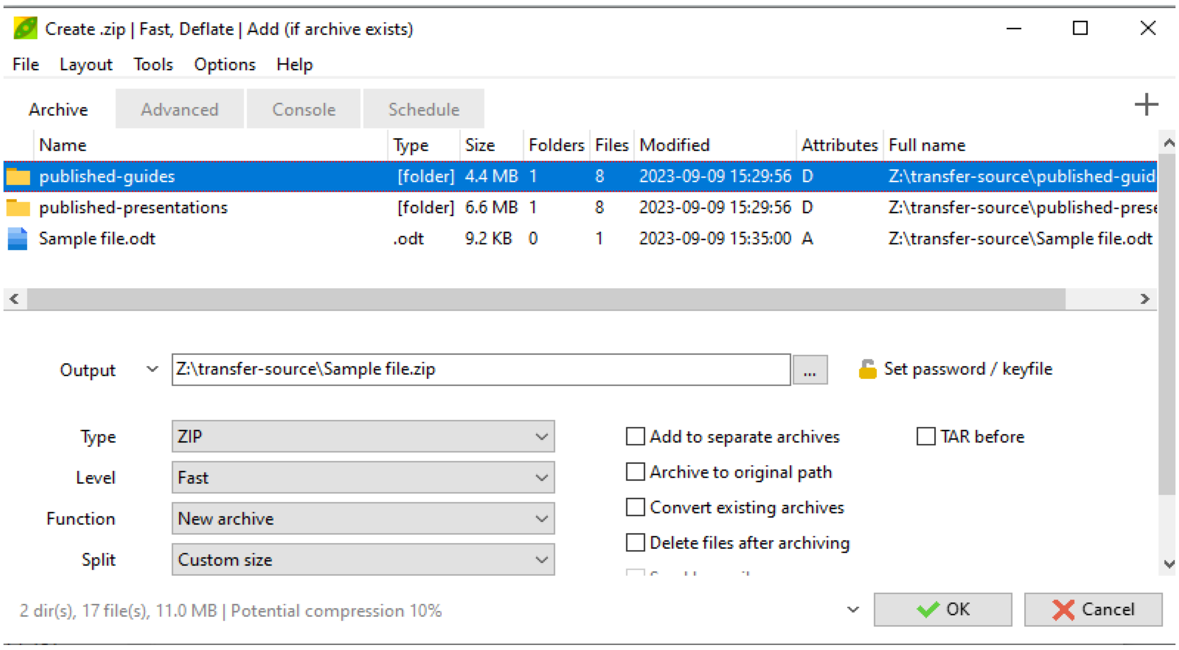
0 dir(s), 1 file(s), 9.2 KB | Potential compression 20%

OK Cancel

Figure 7: Added files

Figure 8 shows the result of adding two folders.

Note that while you can select multiple files but when adding folders these have to be added one at a time.



Create .zip | Fast, Deflate | Add (if archive exists)

File Layout Tools Options Help

Archive Advanced Console Schedule

Name	Type	Size	Folders	Files	Modified	Attributes	Full name
published-guides	[folder]	4.4 MB	1	8	2023-09-09 15:29:56	D	Z:\transfer-source\published-guid
published-presentations	[folder]	6.6 MB	1	8	2023-09-09 15:29:56	D	Z:\transfer-source\published-prese
Sample file.odt	.odt	9.2 KB	0	1	2023-09-09 15:35:00	A	Z:\transfer-source\Sample file.odt

Output Z:\transfer-source\Sample file.zip ... Set password / keyfile

Type ZIP

Level Fast

Function New archive

Split Custom size

☐ Add to separate archives ☐ TAR before

☐ Archive to original path

☐ Convert existing archives

☐ Delete files after archiving

2 dir(s), 17 file(s), 11.0 MB | Potential compression 10%

OK Cancel

Figure 8: Adding two folders

Note that there is a summary at the bottom left of the display showing how many folders(dirs) and files. The total compressed size is an **estimate** of the potential compression that **might** be achieved.

4.3.2 Annotation 2 – configuring where the archive will be created

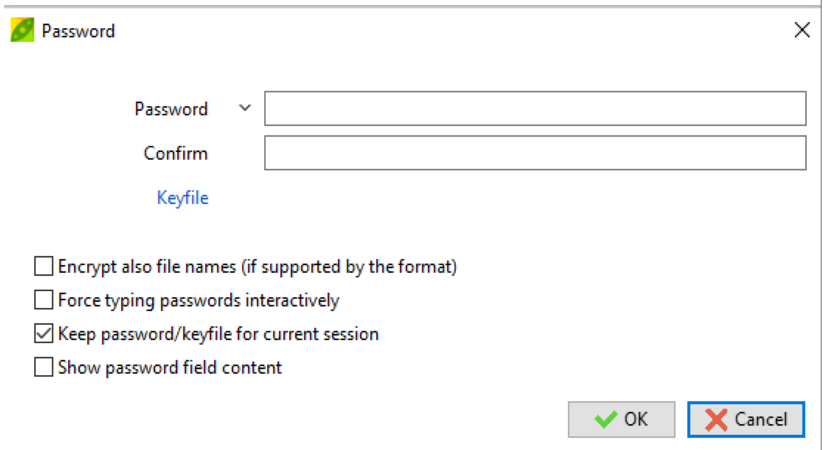
Clicking on the three dots at the end of the Output text box opens a new dialog box to create an archive file.

You can select a file path plus file name. In this case a filename and path has already been created. This is still the case if you are splitting the archive file.

Note that if the archive file is split then the file extension will be replaced by a numeric chunk number.

4.3.3 Annotation 3 – Clicking on the Set password / Keyfile

This **option** will open a new dialog box to enable you to apply security to the archive. See Figure 9.



Password

Password

Confirm

Keyfile

☐ Encrypt also file names (if supported by the format)

☐ Force typing passwords interactively

☑ Keep password/keyfile for current session

☐ Show password field content

OK Cancel

Figure 9: Apply security to archive

This gives the option of adding security by applying encryption to the archive file. This cannot then be opened without having the security credentials.

Note however that a simple password will normally be sufficient, especially for one-off transfers.

If you choose to use a password you enter your chosen password and then retype it in the Confirm box. A strong password should be typically at least 10 to 15 random characters to be of benefit and include some upper and lower case characters, numbers and special characters e.g. @&%# etc. to be secure. 20 characters is probably overkill!

This password needs to be safely shared with any remote partner as they will need to enter it to access the data. This password must be shared via a secure route. Simple email is NOT secure!

Optionally a Keyfile can also be added. This adds another level of security. This is any file, of any length and any type, but if used, must be securely shared with any other party that needs to access the data. A Keyfile could, if required, replace the password.

To add worthwhile security via a Keyfile it is very important to understand that the Keyfile MUST be sent to your partner by some other transport mechanism such as post, text message, email, telephone etc. It defeats the purpose of additional security if it is shared using the same method as the data transfer!

It is recommended that you do NOT add this additional complexity of a Keyfile if it can be avoided

4.3.4 Annotation 4

This option specifies the type of encryption and compaction method to be used. It is recommended that the default ZIP format is used unless there is good reason to use an alternative. It is beyond the scope of this document to expand on the extensive range of options available.

If the data exchange between parties is a regular occurrence, and involves a significant volume of data, it may be worth comparing the ether options to see if any time can be saved. Encoding takes time and for large files this could be significant. ZIP is the default.

4.3.5 Annotation 5 – type of compression

This controls the amount of compression that will be applied. Figure 10 shows the options.

In most cases Normal will be appropriate but, as with the choice of compaction method, some experimentation might be appropriate if regularly used to exchange similar data.

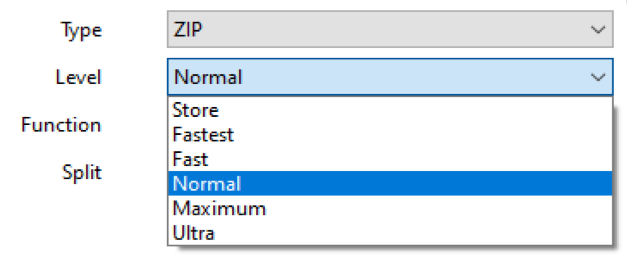

Type: ZIP
Level: Normal

Figure 10: Compression options

Note that some data files are highly compressible whereas some file types are not compressible at all and attempts at compression may sometimes actually make the file larger rather than smaller!

With the sample data used in this document the estimate for all the options is 10% for all options! Each file however is small and much larger files are more likely to benefit.

4.3.6 Annotation 6 – New or existing archive

This allows existing archives to be modified. It is assumed in this document that a new archive will always be created.

4.3.7 Annotation 7 – file splitting choices

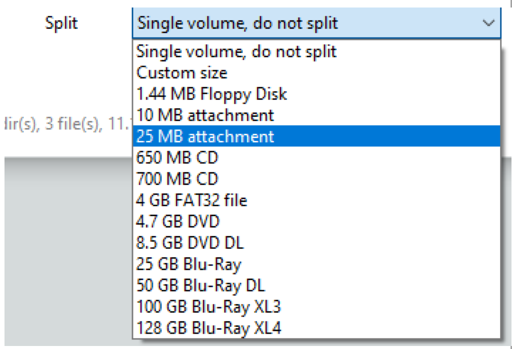
It is this feature that makes PeaZip unique as other similar archiving programs require an additional program to perform this splitting and recombine function or do not work across different platforms.

This option can be very important for file sharing over the Internet as it allows a large archive file to be split into "chunks" removing any limitations on size although it does require some manual steps to be added as will be seen as the chunks need to be separately transported.

When transporting a large file over the Internet using a service such as email or WeTransfer there will be a limit on the maximum size of a data block or chunk that can be handled.

With email this is typically 10 Mbytes but can vary between email providers. With WeTransfer this limit is is 2 Gbytes but for efficiency a 1 Gbyte chunk might be preferable.

PeaZip offers the chunk sizes listed in Figure 11 as standard options but for optimum use by WeTransfer then we need to select the **Custom size** size option. See Figure 12.


Split: Single volume, do not split

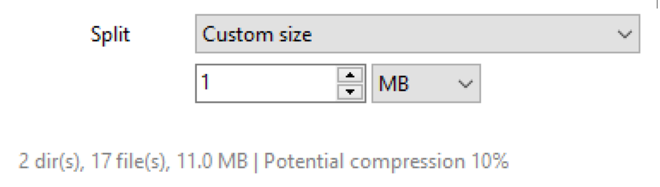
Single volume, do not split
Custom size
1.44 MB Floppy Disk
10 MB attachment
25 MB attachment
650 MB CD
700 MB CD
4 GB FAT32 file
4.7 GB DVD
8.5 GB DVD DL
25 GB Blu-Ray
50 GB Blu-Ray DL
100 GB Blu-Ray XL3
128 GB Blu-Ray XL4

ir(s), 3 file(s), 11.

Figure 11: Standard file splitting options

You can choose the units in the pull-down list – here we have selected Gigabytes.

The optimum choice for WeTransfer is probably 1 GB chunks.


Split: Custom size

1 MB

2 dir(s), 17 file(s), 11.0 MB | Potential compression 10%

Figure 12: Custom size

Note in this guide a much smaller custom chunk size of 1 Mbyte has been chosen purely to simplify the preparation of this document.

Also note the summary at the bottom which shows there are in this case:

- 2 dirs(s) or folders
- 17 files with 11.0 Mbytes of data
- Estimate of compression = 10%

4.3.8 On completion

When the operation has complete you should finish up with an archive that either:

- contains a single Zip file
- or multiple chunks, numbered using the file extension from . 001 to .nnn which are of equal size apart from potentially the final chunk which could be smaller.

Encryption can optionally have been applied using a password and/or a key-file.

This file, or these chunks, can now be transmitted by WeTransfer, or some other means, to the other party and, if they are encrypted, knowing that they are protected while in transit.

5 ON RECEIPT OF THE FILES

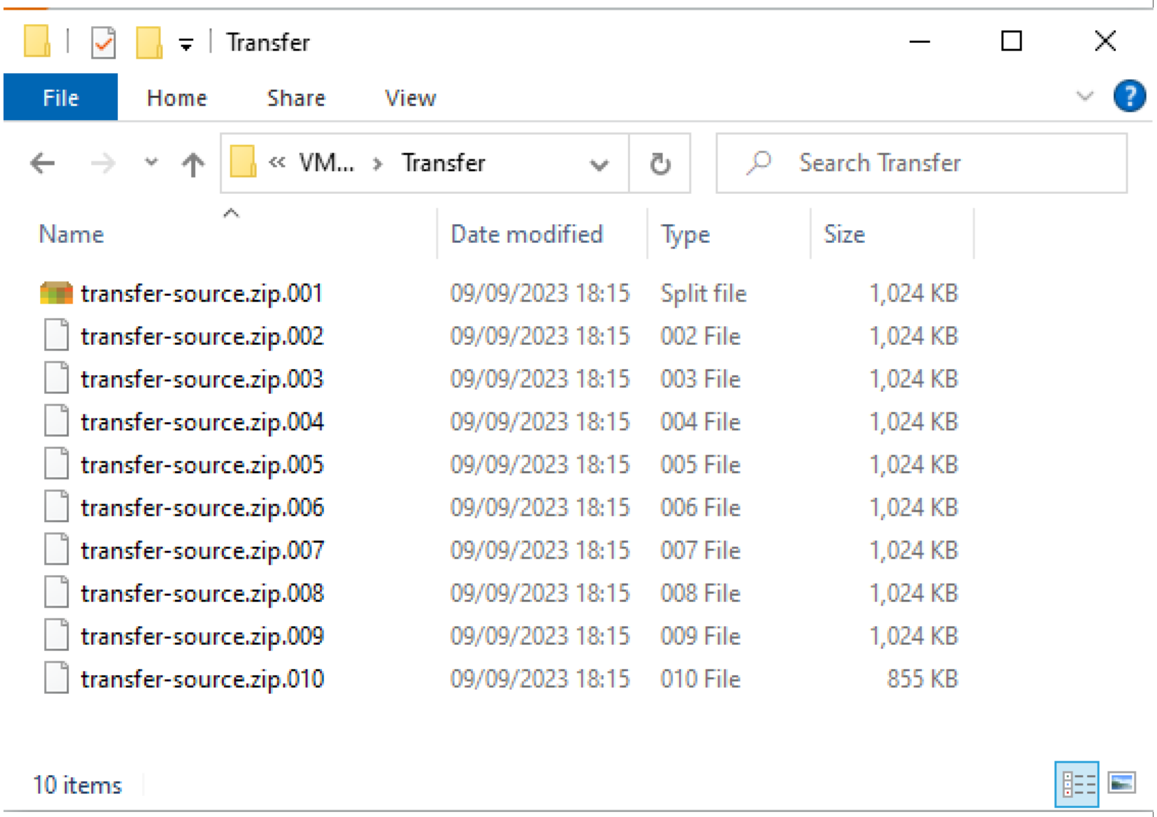
5.1 Archive files

It is assumed here that the data is all in a single archive file or that all the chunks have been received and are stored in a folder.

The location of the folder will have been selected when you receive the files via WeTransfer. The sample files that have been used for this document are in a folder and are listed in Figure 14. This diagram shows that the transfer has been send as 10 chunks.

If it were not divided into chunks then the single file would have a .zip file extension.

It is assumed that PeaZip has already been installed.



Name	Date modified	Type	Size
transfer-source.zip.001	09/09/2023 18:15	Split file	1,024 KB
transfer-source.zip.002	09/09/2023 18:15	002 File	1,024 KB
transfer-source.zip.003	09/09/2023 18:15	003 File	1,024 KB
transfer-source.zip.004	09/09/2023 18:15	004 File	1,024 KB
transfer-source.zip.005	09/09/2023 18:15	005 File	1,024 KB
transfer-source.zip.006	09/09/2023 18:15	006 File	1,024 KB
transfer-source.zip.007	09/09/2023 18:15	007 File	1,024 KB
transfer-source.zip.008	09/09/2023 18:15	008 File	1,024 KB
transfer-source.zip.009	09/09/2023 18:15	009 File	1,024 KB
transfer-source.zip.010	09/09/2023 18:15	010 File	855 KB

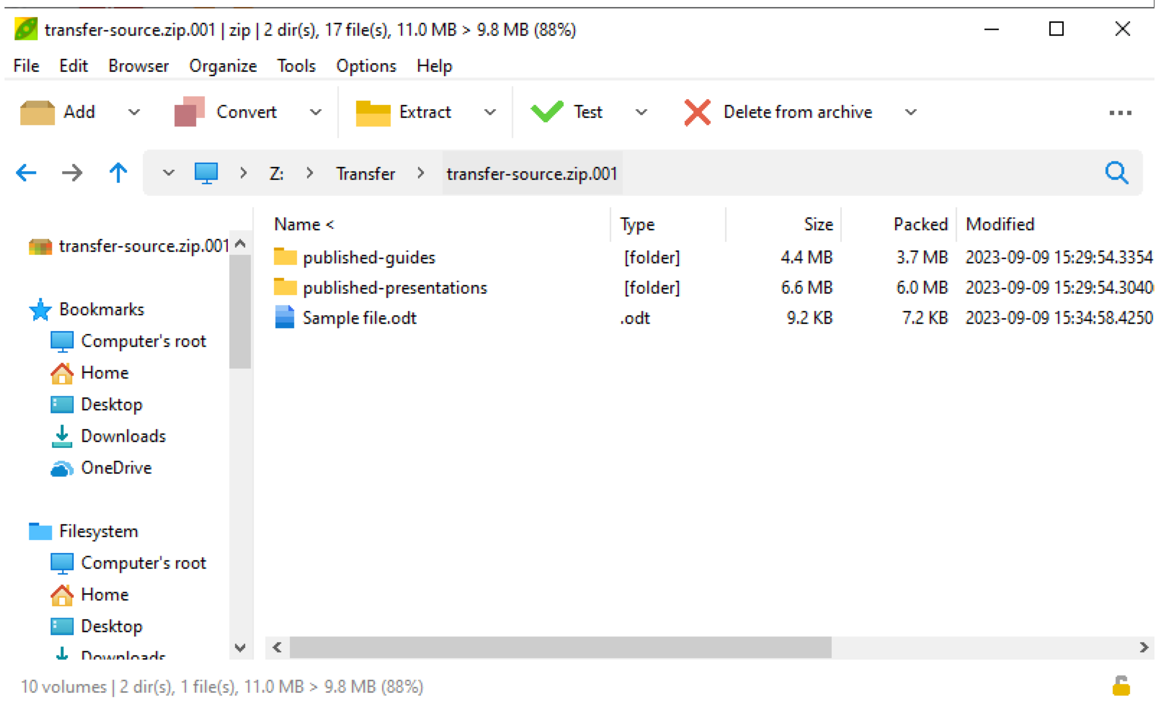
10 items

Figure 13: Received files as chunks

This shows that the files each have have a numbered file extension from .001 to .010 in this case.

You can also see that the first file in this list has a coloured icon. This shows that PeaZip has already been installed on this computer and has recognised this dile set as a series of chunks.

To extract these files i all that is needed is to double click on the coloured icon which will open PeaZip and it will display a view such as Figure 14. If there are no chunks then just clip the Zip file.



transfer-source.zip.001 | zip | 2 dir(s), 17 file(s), 11.0 MB > 9.8 MB (88%)

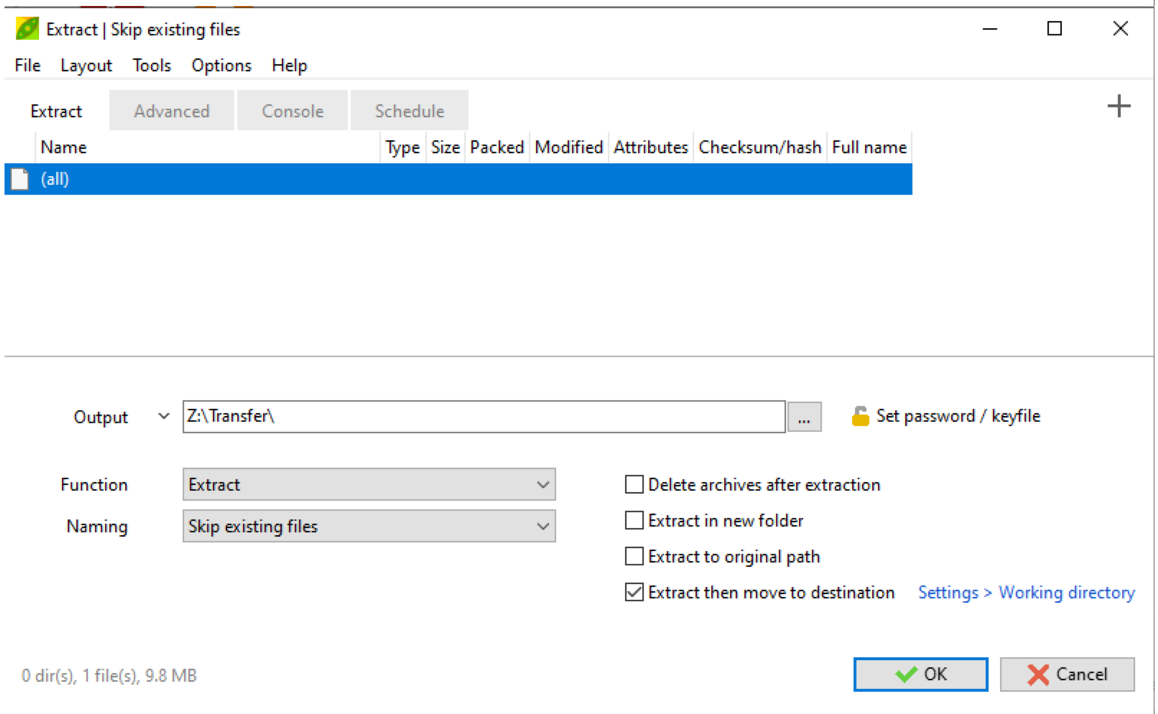
Name <	Type	Size	Packed	Modified
published-guides	[folder]	4.4 MB	3.7 MB	2023-09-09 15:29:54.3354
published-presentations	[folder]	6.6 MB	6.0 MB	2023-09-09 15:29:54.3040
Sample file.odt	.odt	9.2 KB	7.2 KB	2023-09-09 15:34:58.4250

10 volumes | 2 dir(s), 1 file(s), 11.0 MB > 9.8 MB (88%)

Figure 14: PeaZip has identified the Zip file and shows its content

All we now need to do is to extract the files.

Click the **`Extract`** button on the menu bar. This will show the extract screen. e.g. Figure 15



Extract | Skip existing files

Output: Z:\Transfer\

Function: Extract

Naming: Skip existing files

- [ ] Delete archives after extraction
- [ ] Extract in new folder
- [ ] Extract to original path
- [x] Extract then move to destination

0 dir(s), 1 file(s), 9.8 MB

Figure 15: Extract view

We now need to choose where to extract the archive to. Use the three dots to go to the option to select or create the folder as shown in Figure 16 below.

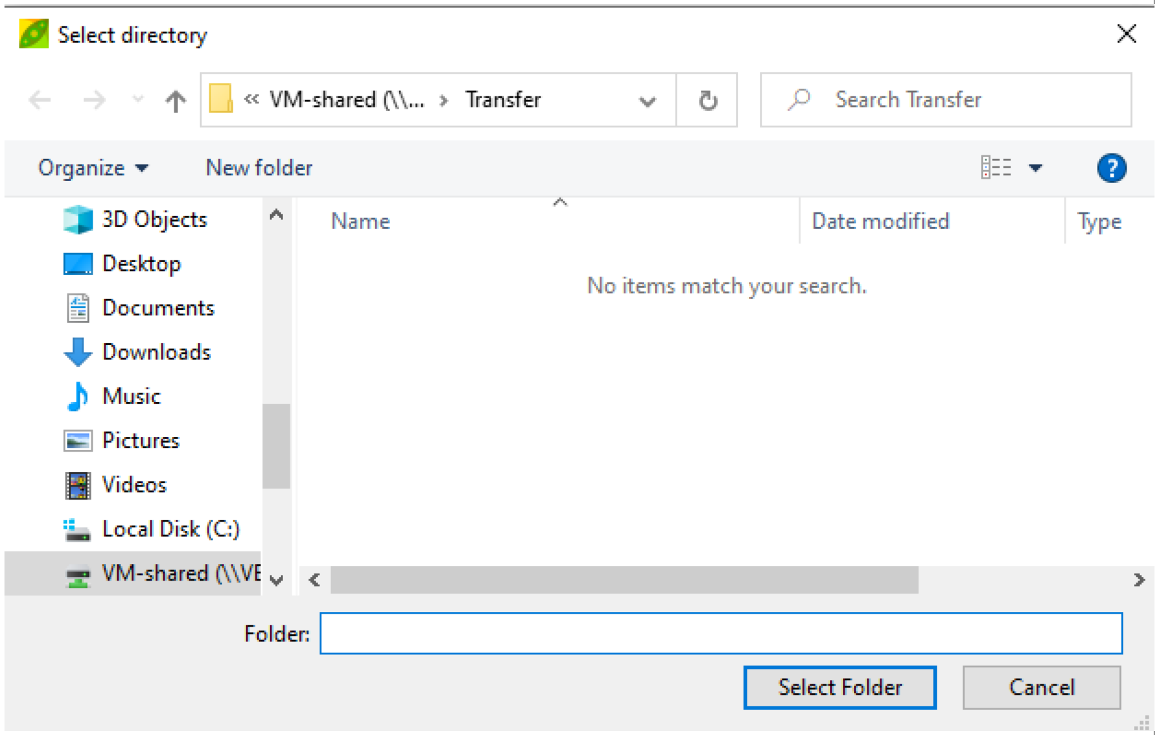
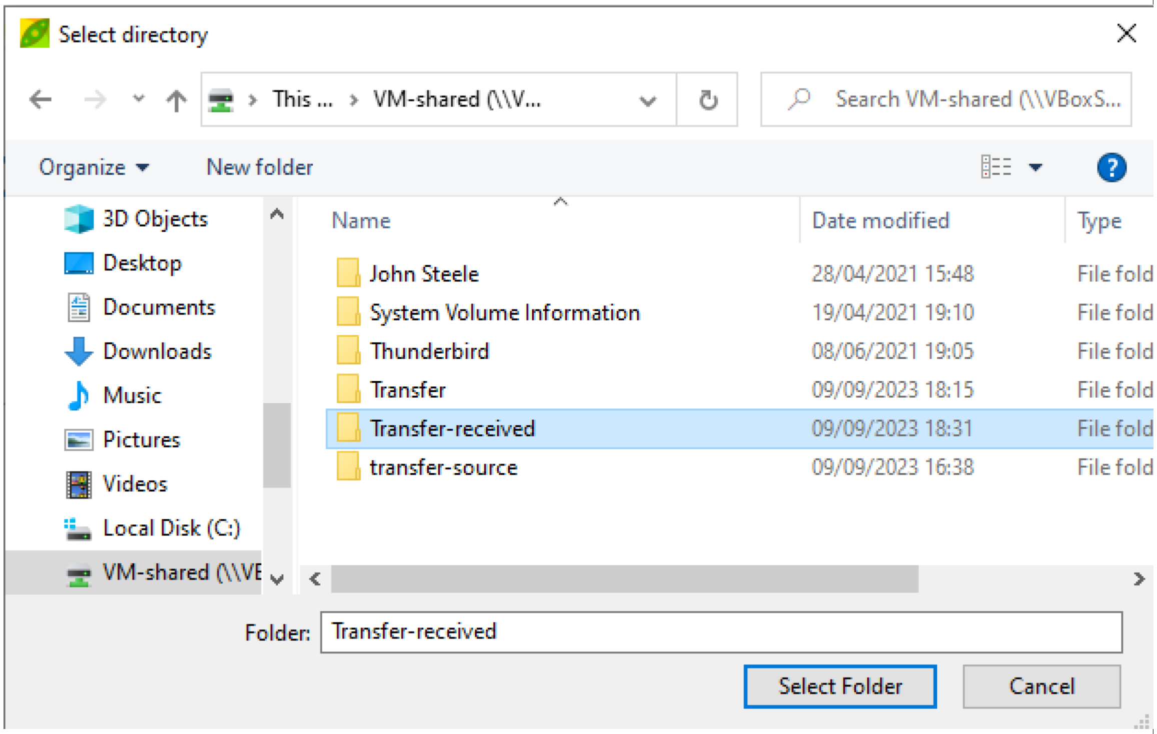


Figure 16: Choose folder to receive the extracted files

This includes a button **Select Folder** to enable you to choose there the files will be recombined, decrypted (if required) and decompressed to. See Figure 17 below,



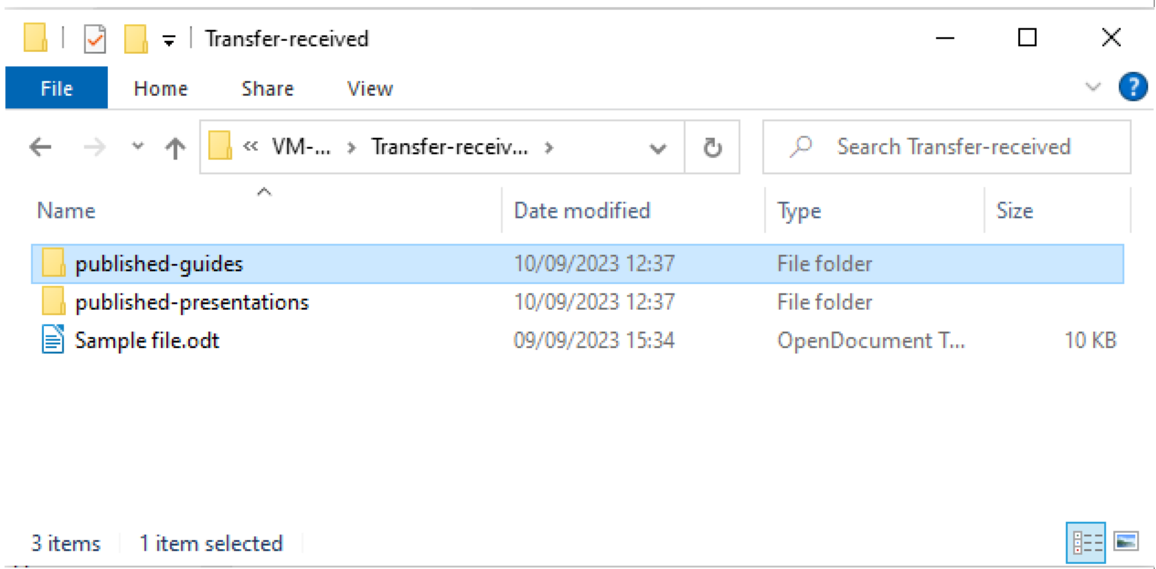
Name	Date modified	Type
John Steele	28/04/2021 15:48	File fold
System Volume Information	19/04/2021 19:10	File fold
Thunderbird	08/06/2021 19:05	File fold
Transfer	09/09/2023 18:15	File fold
Transfer-received	09/09/2023 18:31	File fold
transfer-source	09/09/2023 16:38	File fold

Folder: Transfer-received

Figure 17: Create folder to receive the files

This will revert to the view Figure 15 but with the selected folder shown and just click on the **OK** button.

You will then have an identical copy of the sender’s files! See Figure 18 below.



Name	Date modified	Type	Size
published-guides	10/09/2023 12:37	File folder	
published-presentations	10/09/2023 12:37	File folder	
Sample file.odt	09/09/2023 15:34	OpenDocument T...	10 KB

3 items 1 item selected

Figure 18: File/folder as received

The folders have the date created but the dates on the files extracted are those on the that ere in the source files.

If one of the chunks in the set received is missing you will get an error.