MoleBox Ultra

Creating Portable and Protected Applications With Ease

Table of Contents

Overview

MoleBox Ultra is a software virtualization and protection tool that allows delivering your application as a portable stand-alone secure EXE file which runs instantly with zero installation.

MoleBox Ultra packs all application files into a single efficient executable file that works without extracting packed files to the hard drive and creating temporary files. MoleBox Ultra also applies a number of protection techniques to packed files, including anti-crack protection for EXE and DLLs, resource protection, protection from modification for data files, and many more.

Despite its outstanding functionality, MoleBox Ultra is very easy to use: you don't need to deal with complicated settings, create installation programs or make system snapshots. All you need is to pick out the files required by your application and pack them.

MoleBox Ultra Use Cases

Creating Portable Versions of Your Applications

MoleBox Ultra lets you create portable applications from vast majority of Windows programs. In doing so, it processes all application files in order to create a single executable file or one executable and several data packages. Packed application keeps the functionality of the original program and works without extracting packed files to the hard drive.

Portable applications made with MoleBox Ultra have several benefits:

- **Running without installation from any source.** Moleboxed application runs without installation, extracting packed files or creating temporary files. It can be launched from any source – network drive, CD-ROM, USB device.
- **Avoiding DLL hell and version conflicts.** MoleBox embeds DLLs and ActiveX components into the EXE file, making sure that your application will always use the same DLL versions it was compiled with. No 3rd party program can harm your app's functionality, and vice versa.
- **No changes to the Windows registry.** Virtual registry technology used in MoleBox Ultra intercepts registry i/o operations. This allows using any embedded ActiveX components without registration on the client PC and also reduces compatibility risks.

Reliable Protection for Your Software

MoleBox Ultra protects your application from software piracy, reverse engineering, unauthorized use and modifications. The following are the main protection features:

- **A number of protection techniques for the main executable.** MoleBox Ultra offers you protection from static reverse engineering and classic cracker's attacks (anti-

debug, anti-dump, anti-trace), protection of the application entry point and import table.
- **Protecting DLLs from unauthorized use.** MoleBox hides embedded DLLs and protects them from being reused by 3rd party software.
- **Protecting media and data files from prying eyes.** Games and multimedia presentations often include a lot of exclusive content: 3D models, textures, 2D images, custom fonts, photos, sound effects and music files. Obviously, you wouldn't like if someone else will use your media files. After moleboxing your game or presentation, these files cannot be viewed, extracted, modified or otherwise used by another person.
- **Preventing antivirus 'false positives'.** Moleboxed application may be masked as a common just-compiled, non-wrapped program.

Releasing Patches and Updates

When you release an update to your software and the whole installation package would be huge, you can distribute a small package containing only new or/and modified files. For more information, see *Creating patches and updates* section.

Enhancing Your Application

Running a DLL on application startup. One of the embedded DLLs (an 'activator' DLL) can be executed before the original EXE entry point. For example, you can integrate a 3rd party activation or licensing component without any changes to your program code. For more information, see *Activator DLL* section.

System Requirements

MoleBox Ultra processes native 32bit and .NET applications and supports Windows 2000/ XP/Vista/7 operating systems.

Remarks
MoleBox replaces .NET launching procedure in packed .NET applications and makes them look native to the operating system. However, such applications still require a proper version of .NET framework installed on the end-user PC.

MoleBox Ultra Migration Guide

This section explains major changes in MoleBox Ultra since MoleBox 2.x (Standard and Pro editions).

Package Creation

MoleBox ultra project now allows creating one package only. To create additional packages, you need a new project. As an effect of this, data packages are now independent from EXE files.

Virtual Paths and Folders

MoleBox Ultra now allows packaging files from any locations rather than copying them to the original executable's folder. Package directory tree does not longer have to reproduce the one from your hard drive. Thus, you can create virtual folders inside the package and rename packed files.

Processing Appended Data

Some EXE files have additional data appended at the end, usually containing embedded media content or scripts; for example, Visual Fox Pro executables include their programs. MoleBox Ultra now processes such fragments appending the same data at the end of executable package; appended data can be compressed and encrypted along with the EXE file. Processed executable operates appended data just like the original one does. Now you don't need packing the EXE file twice as it was necessary in MoleBox 2.x.

Using MoleBox Ultra

How To Package And Protect Your Application

MoleBox Ultra processes software applications by packaging them into one or more files (*packages*). Each package is a virtual file system containing some files of your application suite. In the simplest case, you can put all application files into a single EXE file.

If a package contains the main EXE file of your application, it's called *executable package* and it should be compiled into an EXE file. Other packages are compiled into files with extensions of your choice.

When an executable package starts, it can *mount* other packages and use files they contain. Packages are mounted by file name pattern *(mask)*, in alphabetical order.

Projects are stored in MXB files. You need to create a separate project for each package.

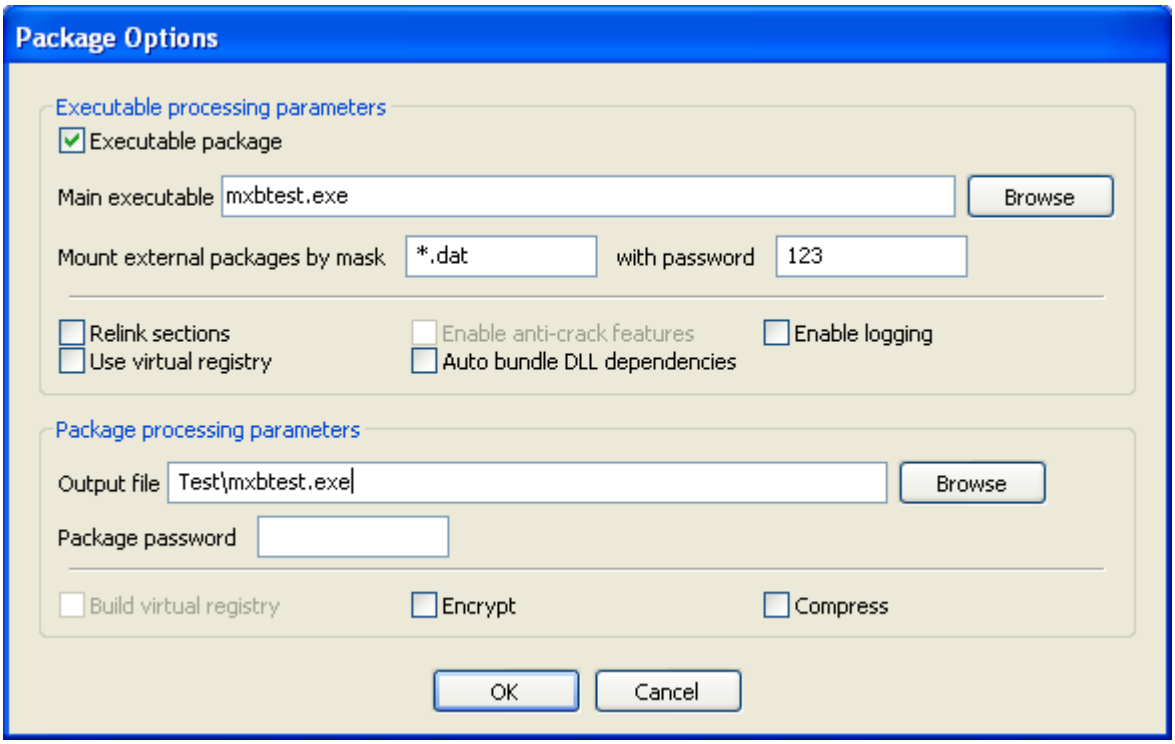
Creating Packages from GUI

To create a moleboxed package, follow these steps:

1. Set package options
2. Add files
3. Finalize package.

Setting Package Options

In the **Tools** menu, select **Options...**, or click **Package Options...** button in the toolbar. **Package Options** window opens.



Selecting the main executable

If the package contains main application executable, set **Executable package** option on and select your original executable with **Main executable**.

Identifying external packages

An executable package can locate and mount packages in its home directory, which allows packed program to use files from these packages. Packed files are extracted directly to the memory and never written to disk. MoleBox uses file name pattern **(mask)** and **password** to identify which packages to read. You may leave these options blank, then default values are used.

Setting wrapping mode

MoleBox offers 2 different wrapping modes: *compatibility mode* and *protection mode*.

In compatibility mode MoleBox makes minimal changes to the original EXE file, solely adding extra sections and changing entry point. Hence, protection and compression are not available. To select this mode, deselect **Relink sections** checkbox.

In protection mode application sections are rewritten from scratch, making it look like a common C++ application. Also, MoleBox packs application resources except the icon and version information. To select this mode, select **Relink sections** checkbox; you may also select **Enable anti-crack protection** option to enable protection features.

You may want to use compatibility mode in the following scenarios:
- after using MoleBox, your application should be wrapped with another 3rd party software, for example, by a publisher, and protection mode is not compatible to that software;
- your application needs to have its original structure intact.

In other scenarios, protection mode is recommended.

Virtual registry settings

Virtual registry allows running your application without any changes to Windows registry on the host computer. Virtual registry is used for temporary registration of packed components. Additionally, if your application creates a registry entry in runtime, it is stored in virtual registry if:
- its parent branch is stored in virtual registry,
- or it is located in components registration area.

When packed application requests for a registry key, MoleBox first looks for this key in the virtual registry, if it is enabled.

Build virtual registry option generates virtual registry for packed components, and **Use virtual registry** allows executable package to use built-in virtual registries.

Other executable processing parameters

MoleBox may check your application DLL dependencies and automatically embed all non-system DLLs. This procedure is recursive. System DLLs should be embedded explicitly. To do so, select **Auto bundle DLL dependencies** option.

Enable logging option turns on virtualization log in your packed application. It may be useful for debugging purposes.

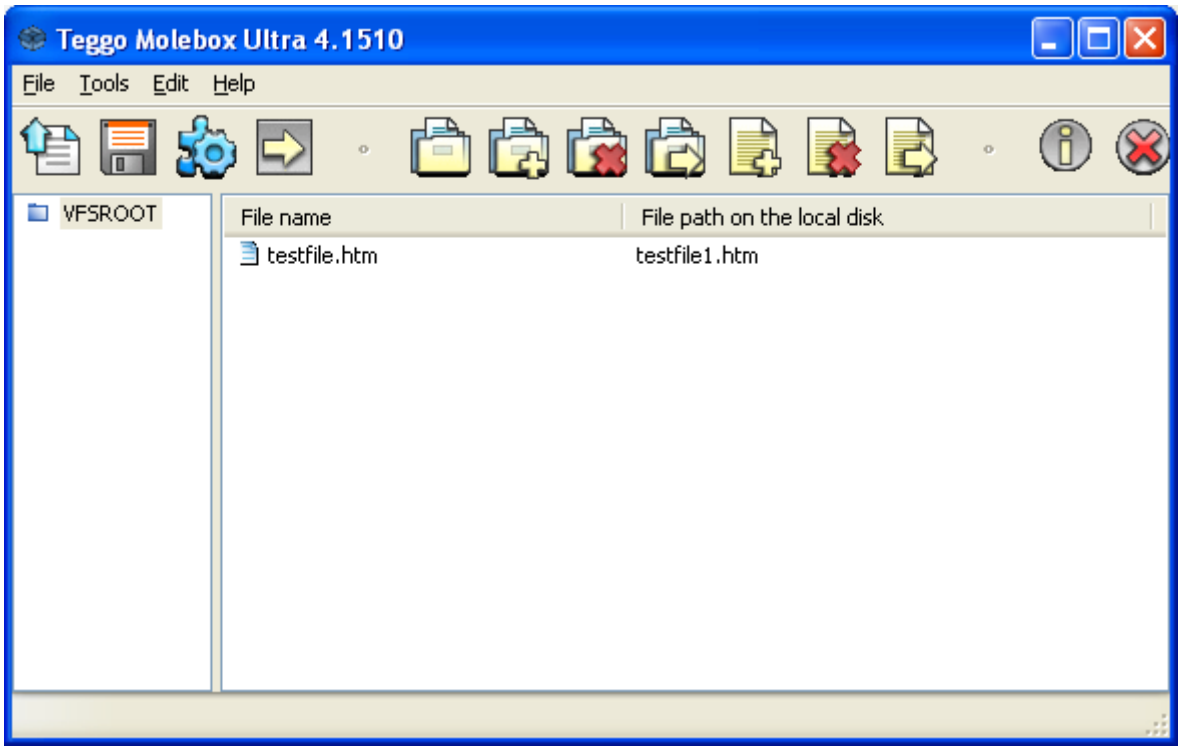
Package processing parameters

Select **Output file** and **Package password**. When compiling a data package, make sure they fit mounting options in the executable package.

You can also **Encrypt** and **Compress** packed data files; these options do not affect compression and encryption of the main executable.

Adding Files

The main window displays contents of the current package in a tree view and allows you to modify it.



File name column displays short name used to access the file from the packed application, **File path on the local disk** shows the name of the source file relative to the project file location.

Use toolbar buttons or **Edit** menu items to fill and modify the package contents. You can either add entire folders or separate files. If you add a folder, its subfolders are recursively added as well. Additionally, you can create virtual folders – folders inside the package, which allows building directory structure different from the one on the hard drive. You may also assign a different name (alias) to a packed file by renaming it.

Finalizing the Package

At this point your configuration is ready for packaging. Please note that you need to activate your copy of MoleBox Ultra before using this functionality.

To start packaging:

- in the **Tools** menu, select **Pack...** or click **Pack** button in the toolbar;
- **Packing** window opens and displays packing progress;

- when processing of an executable package is finished, you can click **Execute a test** button to check your application.

Creating Packages through the Command Line

You can also create packages using command line packaging tool, **mxbpack.exe**. This tool allows generating packages either using project files created earlier or without project files, specifying a list of files to pack directly in the command line.

Command line tool has the following syntax:

```
mxbpack.exe [options] <pefile.exe> [-o out.exe] @<mxbconf.mxb>
mxbpack.exe [options] <pefile.exe> [-o out.exe] [files_list]
```

Parameters:

`<pefile.exe>`
: Main executable for executable packages; '~~' for non-executable packages or when using an existing project file.

`-o <outputfile>`
: Output file name.

`@<mxbconf.mxb>`
: Existing project file.

`files_list`
: List of files to add to the package.

The following options are available:

Option	Meaning
`-e <dllname>`	Use DLL as an embedded activator
or	
`-a <dllname>`	Use DLL as an external activator
`-B`	**Auto bundle DLL dependencies**
`-l`	**Enable logging**
`-O`	**Build virtual registry** and **Use virtual registry**
`-z`	**Relink sections**, also compress and encrypt executable
or	
`-Z`	**Enable anti-crack features, Relink sections**, also compress and encrypt executable
`-x, -X`	Compress and encrypt packed content files.
`-p <password>`	**Package password**
`-P <password>`	Password used in external packages for mounting

`-m <filemask>` File name pattern (mask) for external packages

Option meanings in **bold** refer to option names in the **Package Options** window. Options set via command line override the same options from the configuration file.

Files list includes absolute or relative file names, and may also include file aliases.

Examples

Processing an existing configuration, either executable or non-executable package, without modifications:

```
mxbpack.exe ~~ @mxbconf.mxb
```

Integrating 2 DLLs into an EXE file with protection:

```
mxbpack.exe -Z program.exe -o packed.exe libs\lib1.dll;lib1.dll
libs\lib2.dll;lib2.dll
```

MXB File Syntax

MoleBox Ultra project is a text file that can be created and edited either with GUI or manually. The structure of project file is as follows:

```
;MOLEBOX CONFIG SCRIPT 1.0
<options>
option1=value1
option1=value1
...
<files>
[!]filename1[;alias1]
[!]filename2[;alias2]
...
<end>
```

Options section lists the same options that are available from **Package Options** window:

Option name	Value	Meaning
`target`	string	**Output file**
`source`	string	**Main executable**
`pkgpwd`	string	**Package password**
`extpwd`	string	Password used in external packages for mounting
`extmask`	string	File name pattern (mask) for external packages
`isexecutable`	yes|no	**Executable package**
`dologging`	yes|no	**Enable logging**
`usevreg`	yes|no	**Use virtual registry**
`buildvreg`	yes|no	**Build virtual registry**
`relink`	yes|no	**Relink sections**
`anticrack`	yes|no	**Enable anti-crack features**
`compress`	yes|no	**Compress**
`encrypt`	yes|no	**Encrypt**
`autobundle`	yes|no	**Auto bundle DLL dependencies**

Files section lists package contents. For each included file you need to specify its filename and you may also set an optional alias (file path relative to the package root, if different). **'!'** sign makes file hidden. You can use either absolute paths or file names relative to the project file location.

Example of a file record:

```
D:\Projects\Project1\Local\En\En.Strings.txt;Resource\Strings.txt
```

Activator DLL

MoleBox ultra allows executing a DLL code before application start. This DLL is called *activator* since it's commonly used for software licensing or activation purposes. Activator DLL may be either embedded into your executable package or kept externally.

The function to be called first should have the following prototype:

```
int Activate(void* p);
```

Parameters:

p – DLL address

Return values:

If the function succeeds and MoleBox should launch packed executable, return any non-zero value.

If the function fails, return 0.

Creating Patches and Updates

Moleboxed application can mount multiple packages in certain, alphabetical order. Files from packages loaded later replace files with same names loaded earlier. This ability can be used to create and distribute updates to your software. All you need is to create a new package and distribute it to your users.

To create a patch or update:

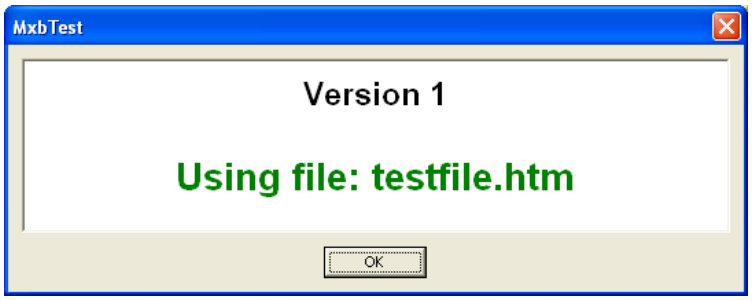
- Launch MoleBox Ultra and create a new project.
- Set the package name to a string 'greater' than already existing packages have. Make sure that package file name fits naming pattern specified in the executable package. For example, to replace a file in the package named "sounds1.dat", you can create a new package named "sounds2.dat" (entire file names, including name extensions, are compared).
- Set the appropriate package password and other options.
- Add the files which are new or being replaced into this package. If you're replacing files, their names in package should match the previously used names.
- Pack this package, and it's ready for distribution.

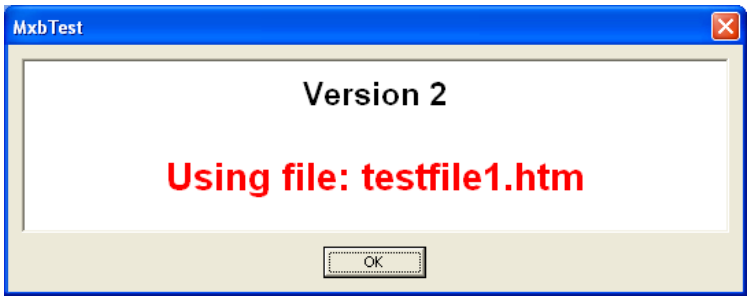
If you need to replace the main executable, you have to rebuild and re-distribute the executable package with the options used before.

Example Application

This section describes initial packaging of a small example application and creating an update package. You can find all necessary files in the **Examples** folder of your MoleBox installation.

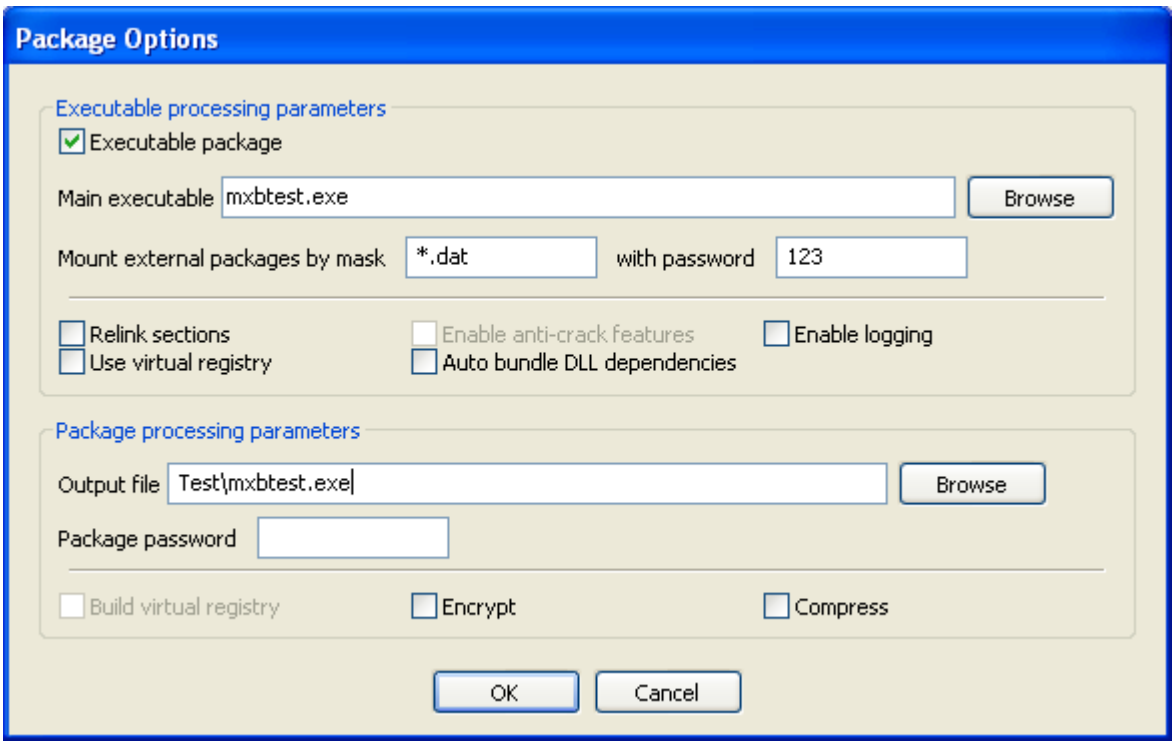
The application shows a form containing text from an HTML file. This application includes 1 executable file **mxbtest.exe** and 1 data file **testfile.htm**. In the 'updated' version the data file is modified and replaced with **testfile1.htm**.

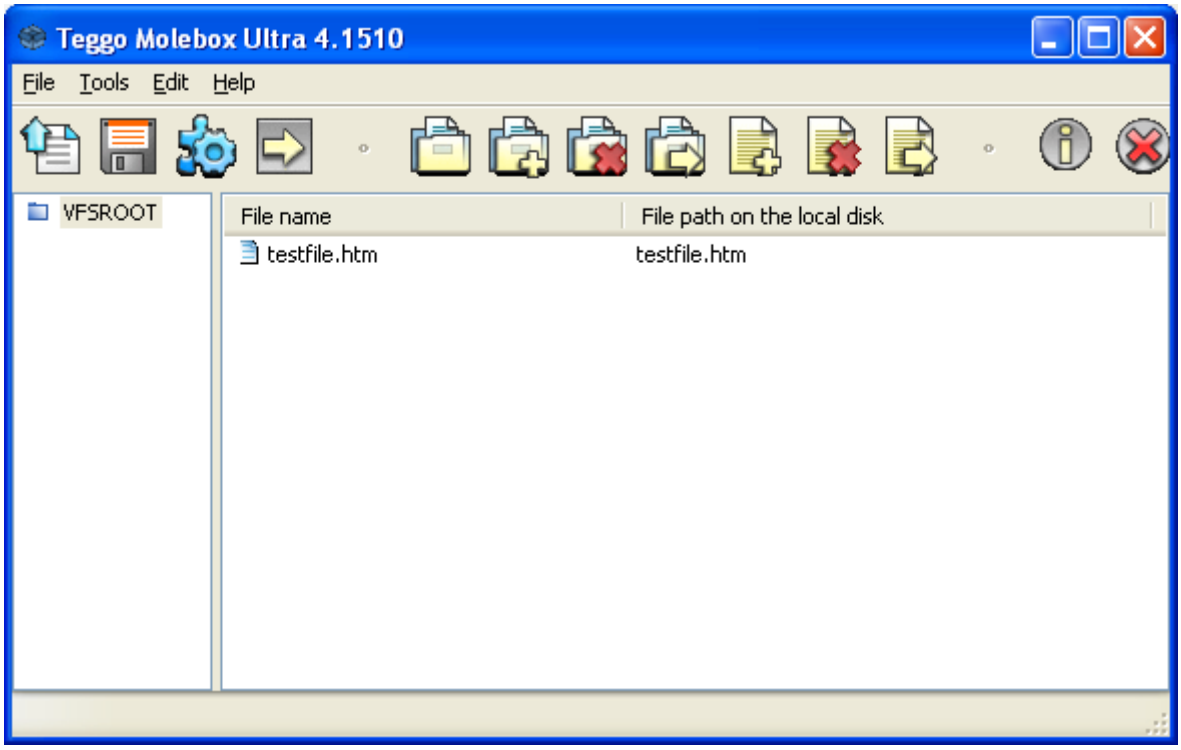




For initial packaging (see configuration **mxbtest.mxb**):

- create one executable package containing **mxbtest.exe**;
- set output file to **Test\mxbtest.exe**;
- set mounting options: mask ***.dat** and password **123**;
- add **testfile.htm** file to package contents;
- pack the package.



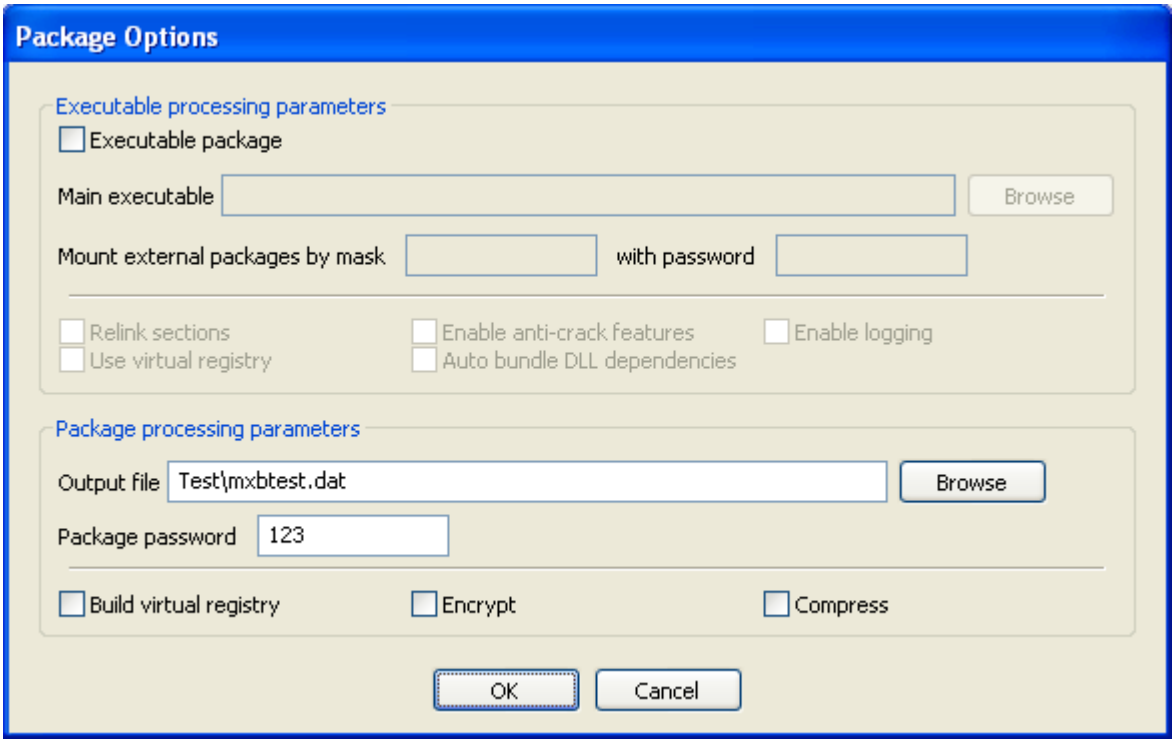


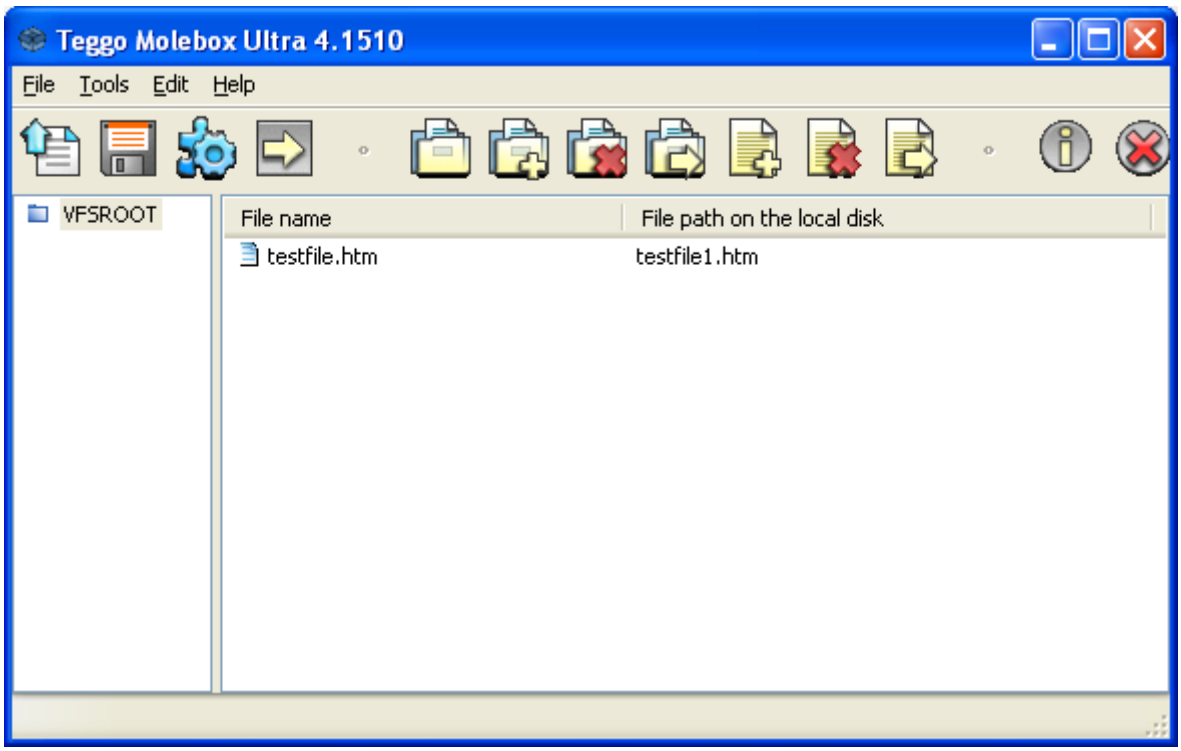
If you run **mxbtest.exe** now, you'll see the first version of the application.

To create the update package (see configuration **datapak.mxb**):

- create a new non-executable package;
- set output file **Test\mxbtest.dat** and password **123**;

- add **testfile1.htm** file to package contents;
- rename **testfile1.htm** to **testfile.htm**;
- pack the package.





If you run **mxbtest.exe** now, you'll see the second version of the application. If you remove **mxbtest.dat** or rename it into a file with another extension, you'll see the first version again.

For command-line version of the same project, see **pack.cmd** file.

MoleBox Ultra Activation

A fully functional MoleBox Ultra demo version is available for free download from the official web site (please find a link below). MoleBox Ultra requires internet activation in order to run, both for demo and registered version. If you use a firewall software, allow Internet access for MoleBox Ultra.

To access activation interface:
- in the **Help** menu, select **Activate...** or click the **ACTIVATE** button in the toolbar; **Activate** window opens.

Automatic Activation

To activate in demo mode:
- enter your e-mail address;
- click **Activate** button.

To unlock the full version:
- if you don't have ab activation code yet, click **Purchase** button to buy it;
- enter your activation code in the corresponding text field;
- click **Activate** button.

Manual Activation

If Internet activation fails for some reason, you can also activate software manually:
- click **Request for web activation** button; your default browser window opens; your hardware id is sent automatically;
- enter your e-mail address or activation code in the browser form;
- submit the web activation form; the next page opens;
- copy the activation response and paste it to the **Activation response** text area, then click **Activate** button.

Manual activation can be also done from another computer. To do so, you need to obtain the hardware id from the target computer. To obtain hardware id:
- access activation interface on the target computer; hardware id is displayed in the **Activate** window;
- or call 'mxbpack.exe -R'; hardware id is returned.

Once you have the hardware id:
- access the web activation form as described above;
- enter hardware id of the target computer;
- enter e-mail address or activation code;
- submit the form;
- write down or copy the activation response;
- access activation interface on the target computer;
- enter activation response and click **Activate** button.

Support and Contacts

Please send us your feedback at support@molebox.com or visit support page (see URL below).

Bug reports

If you have problems running MoleBox or your moleboxed app works differently then the original one, please make sure you are using the last released version of Molebox. If your problem repeats, please contact us via e-mail or via feedback from. Please include full description of your problem and provide the following information:
- your operating system and MoleBox version;
- sequence of your actions;
- expected behavior of packed application;
- wrong behavior;
- any other information that may help us solving the problem.

Sometimes we'll need a sample app which reproduces the error you encountered. Please send us a non-moleboxed test case, packed in zip or rar archive. *By providing us with your sample project, you grant us the rights to modify, analyze, decompile, disassemble, otherwise reverse engineer your application, and to include your application into our test base.*

Contact Information

Product page: http://www.molebox.com/molebox-ultra/
Download link: http://www.molebox.com/download/molebox-ultra/
Order page: http://www.molebox.com/order.shtml
Support page: http://www.molebox.com/support.shtml
E-mail: support@molebox.com
Company web-site: http://www.molebox.com/