RESTIX

User manual

Version: 0.9.6

Table of contents

Introduction

Restix extends open source backup software restic. It provides a GUI and a command line interface for a subset of restic's functionality. Restix uses the concept of backup targets consisting of target directory, scope and access credentials. This concept simplifies backup and restore tasks, especially when multiple backup media are used. A strict separation of backups by hostname, user and year reduces the possibility of inconsistencies in the data backed up.

Restix does not modify restic data in any way, hence restic can always be used directly to access backups made with restix.

Installation

System requirements

Restix runs on the operating systems Linux and Windows. It requires Python 3.10 and above to be installed. Backup software restic is required, version 0.17 and above is needed for full restix functionality.

Installation package integrity

For security reasons all installation packages are provided alongside with an SHA-512 hash. The file containing the hash values is signed with PGP. To verify the PGP signature you need key frank.sommer@sherpa-software.de.

On Linux, add line `keyserver hkps://keys.openpgp.org` to Gnu-PG configuration file `$HOME/.gnupg/gpg.conf`. Then import the key above with `gpg –search-keys frank.sommer@sherpa-software.de`.

On Windows, install GPG4Win and enter Key-Server `hkps://keys.openpgp.org` in the settings for app Kleopatra. Import the key above into Kleopatra.

Debian based Linux (e.g. Ubuntu, Linux Mint, Debian)

For these Linux derivates packages for automatic installation are provided. If any problems arise or installation shall be made for a specific user only, please follow the description for manual installation in section Other Linux derivates (e.g. Arch, Manjaro).

Install restix with full functionality

The full functionality package includes both restix GUI and command line interface.

After installation the restix GUI can be started from start menu, group Utilities resp. Accessoires. Command line interface is accessible from a terminal with command `restix`.

Installation steps:

Download restix-0.9.6.deb, SHA512SUMS and SHA512SUMS.sign from https://github.com/FrankSommer-64/restix/dist.
Open a terminal and change to download directory.
Verify signature with command `gpg --verify SHA512SUMS.sign SHA512SUMS`.
Determine package hash with command `sha512sum restix-0.9.6.deb`.
Make sure, hash matches the value from file `SHA512SUMS`.
Install package with command `sudo apt install ./restix-0.9.6.deb`.

Install minimal restix

The minimal functionaliy package contains restix command line interface only.

After installation command line interface is accessible from a terminal with command `restix`.

Installation steps:

Download restix_core-0.9.6.deb, SHA512SUMS and SHA512SUMS.sign from https://github.com/FrankSommer-64/restix/dist.
Open a terminal and change to download directory.
Verify signature with command `gpg --verify SHA512SUMS.sign SHA512SUMS`.
Determine package hash with command `sha512sum restix_core-0.9.6.deb`.
Make sure, hash matches the value from file `SHA512SUMS`.
Install package with command `sudo apt install ./restix_core-0.9.6.deb`.

Red-Hat based Linux (e.g. Fedora)

For these Linux derivates packages for automatic installation are provided. If any problems arise or installation shall be made for a specific user only, please follow the description for manual installation in section Other Linux derivates (e.g. Arch, Manjaro).

Install restix with full functionality

The full functionality package includes both restix GUI and command line interface.

After installation the restix GUI can be started from start menu, group Utilities resp. Accessoires . Command line interface is accessible from a terminal with command `restix`.

Installation steps:

Download restix-0.9.6.rpm, SHA512SUMS and SHA512SUMS.sign from https://github.com/FrankSommer-64/restix/dist.
Open a terminal and change to download directory.
Verify signature with command `gpg --verify SHA512SUMS.sign SHA512SUMS`.
Determine package hash with command `sha512sum restix-0.9.6.rpm`.
Make sure, hash matches the value from file `SHA512SUMS`.
Install package with command `sudo dnf localinstall ./restix-0.9.6.rpm`.

Install minimal restix

The minimal functionaliy package contains restix command line interface only.

After installation command line interface is accessible from a terminal with command `restix`.

Installation steps:

Download restix_core-0.9.6.rpm, SHA512SUMS and SHA512SUMS.sign from https://github.com/FrankSommer-64/restix/dist.
Open a terminal and change to download directory.
Verify signature with command `gpg --verify SHA512SUMS.sign SHA512SUMS`.
Determine package hash with command `sha512sum restix_core-0.9.6.rpm`.
Make sure, hash matches the value from file `SHA512SUMS`.
Install package with command `sudo dnf localinstall ./restix_core-0.9.6.rpm`.

Other Linux derivates (e.g. Arch, Manjaro)

There are no installation packages for other Linux derivates, restix must be installed manually here.

Install restix with full functionality

The full functionality package includes both restix GUI and command line interface.

After installation the restix GUI can be started from start menu, group Utilities resp. Accessoires.
Command line interface is accessible from a terminal with command `restix`.

Installation steps:

Download restix-0.9.6-custom.zip , SHA512SUMS and SHA512SUMS.sign from https://github.com/FrankSommer-64/restix/dist.
Open a terminal and change to download directory.
Verify signature with command `gpg --verify SHA512SUMS.sign SHA512SUMS`.
Determine hashes with command `sha512sum restix-0.9.6-custom.zip`.
Make sure, hash matches the value from file `SHA512SUMS`.
Extract archive `restix-0.9.6-custom.zip`.
Change to sub directory `restix-0.9.6`.
Update variables in file `install_full.sh`, see section Customise installation script.
To install restix for local user only run `./install_full.sh`.
For a system wide installation run `sudo ./install_full.sh`.

Install minimal restix

The minimal functionaliy package contains restix command line interface only.

After installation command line interface is accessible from a terminal with command `restix`.

Installation steps:

Download restix-0.9.6-custom.zip, SHA512SUMS and SHA512SUMS.sign from https://github.com/FrankSommer-64/restix/dist.
Open a terminal and change to download directory.
Verify signature with command `gpg --verify SHA512SUMS.sign SHA512SUMS`.
Determine hashes with command `sha512sum restix-0.9.6-custom.zip`.
Make sure, hash matches the value from file `SHA512SUMS`.
Extract archive `restix-0.9.6-custom.zip`.
Change to sub directory `restix-0.9.6`.
Update variables in file `install_core.sh`, see section Customise installation script.
To install restix for local user only, run `./install_core.sh`.
For a systemwide installation, run `sudo ./install_core.sh`.

Customise installation script

At the beginning of the script there are three variables controlling the installation process.

Variable **`INSTALL_PATH`** defines the directory, where restix code and Python virtual environment shall be created.

Variable **`LINK_PATH`** defines the directory, where symbolic links to restix executables shall be created. That directory must be in the user's PATH, so that the system can find the executables.

Variable **`SHORTCUT_PATH`** defines the directory, where desktop shortcuts to the restix GUI shall be created. Use directory `/usr/local/share/applications` for system wide installation, and `$HOME/.local/applications` for local user installation. In both cases restix is accessible through the start menu. Value `$HOME/Desktop` creates an icon on the user's desktop, the restix GUI can be accessed by double clicking that icon. When installing minimal restix, the variable is ignored.

Windows

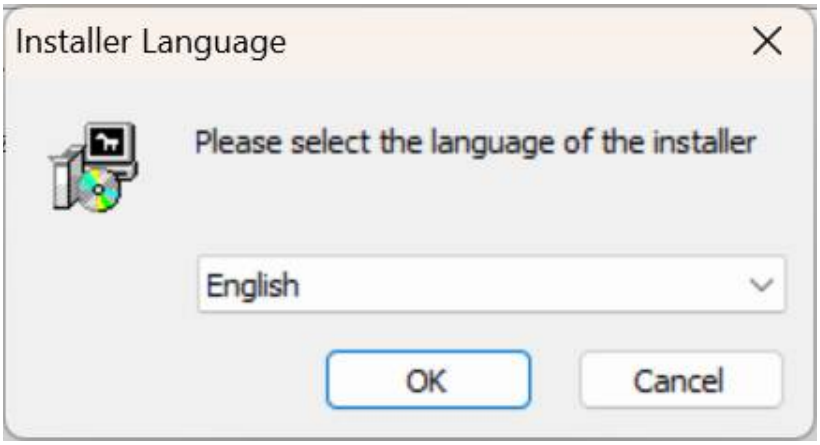
System wide installation

In a system wide installation all users can access restix, administrator rights are required in this case.

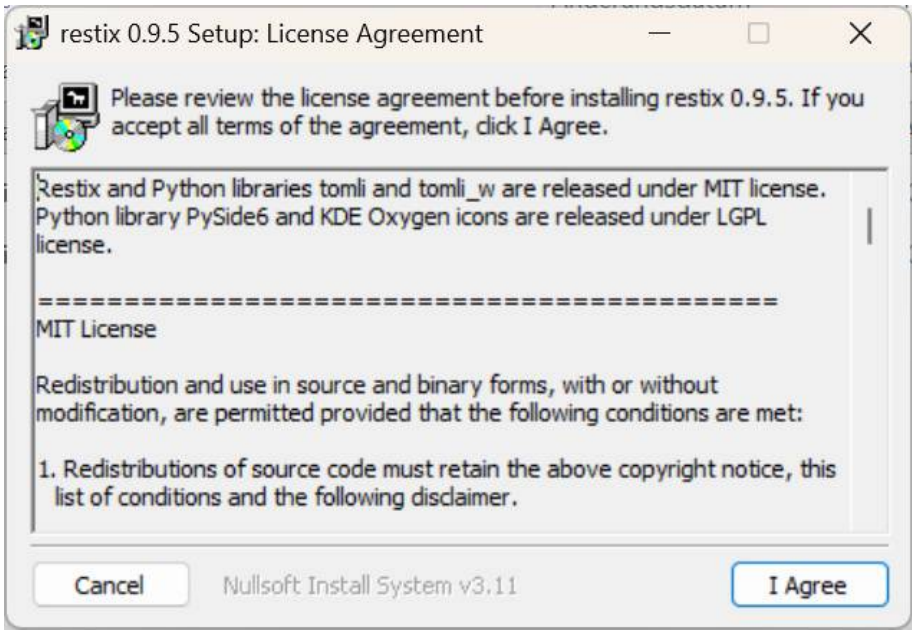
Installation steps:

Download install_restix_0.9.6.exe, SHA512SUMS and SHA512SUMS.sign from https://github.com/FrankSommer-64/restix/dist.
Open a command prompt and change to download directory.
Verify signature with command `gpg --verify SHA512SUMS.sign SHA512SUMS`.
Determine package hash with command `certutil -hashfile install_restix_0.9.6.exe sha512`.
Make sure, hash matches the value from file `SHA512SUMS`.
Start the installer `install_restix_0.9.6.exe` and enter administrator password.

Select desired language for the installation:



Accept license:



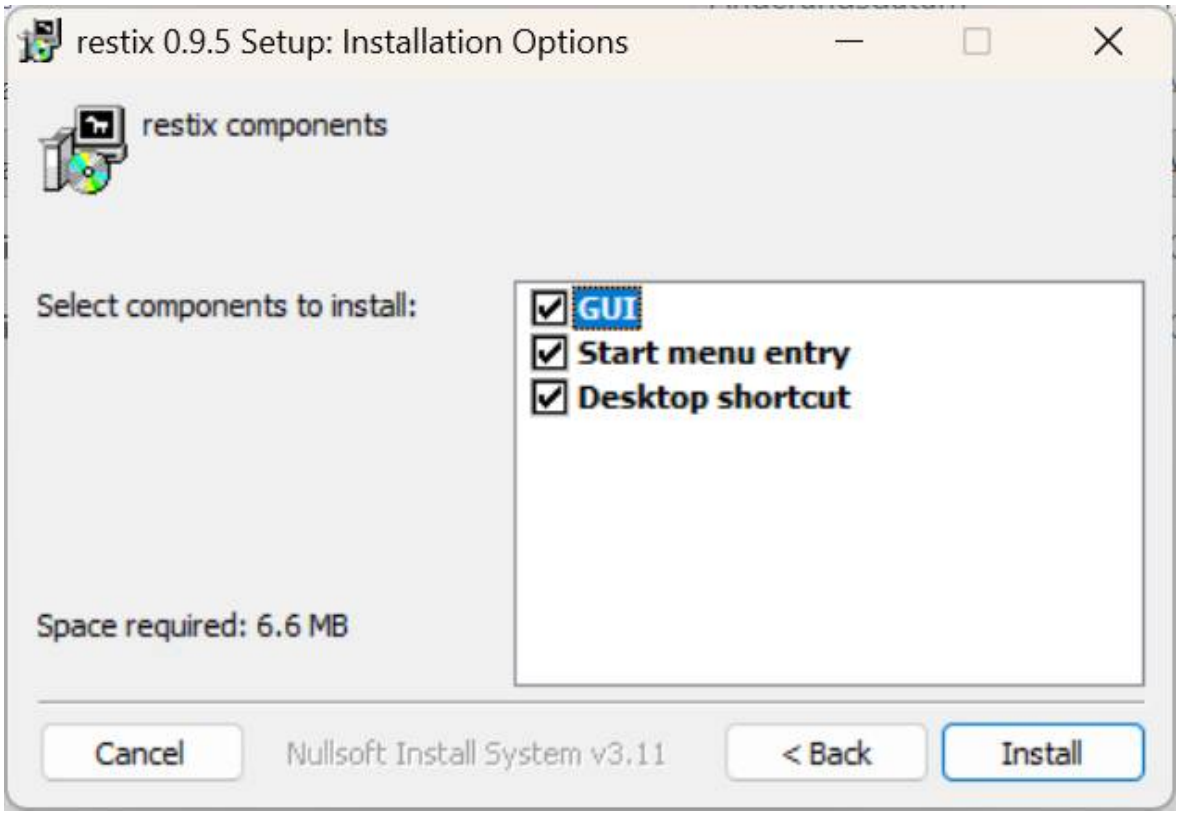
Select options on the last installer page.

If a GUI is not desired, uncheck the corresponding entry. Restix can be accessed through command line interface only in that case.

If an entry in Windows start menu is not desired, uncheck the corresponding entry.

If the installer shall not create a start icon on every user's desktop, uncheck the corresponding entry.

Finally click button **Install**.



Installation for a single user

If restix shall be installed for one user only, alle files are stored underneath the user's home directory. Administrator rights are **not** required in this case.

Installation steps:

Download install_restix_0.9.6_local.exe, SHA512SUMS and SHA512SUMS.sign from https://github.com/FrankSommer-64/restix/dist.
Open a command prompt and change to download directory.
Verify signature with command `gpg --verify SHA512SUMS.sign SHA512SUMS`.
Determine package hash with command `certutil -hashfile install_restix_0.9.6_local.exe sha512`.
Make sure, hash matches the value from file `SHA512SUMS`.
Start the installer `install_restix_0.9.6_local.exe`; language, license agreement and option selection are identical to system wide installation.

Deinstallation

Debian based Linux

Deinstallation is done with command `sudo apt remove restix`.

Configuration directory `$HOME/.config/restix` and file `$HOME/.restix` with GUI settings must be removed manually.

Red-Hat based Linux

Deinstallation is done with command `sudo dnf remove restix`.

Configuration directory `$HOME/.config/restix` and file `$HOME/.restix` with GUI settings must be removed manually.

Other Linux derivates

Remove symbolic links `restix` and `grestix` to restix executables in the directory specified in variable **`LINK_PATH`** of the installation script.

Remove file `/usr/local/share/application/grestix.desktop` resp. `$HOME/.local/applications/grestix.desktop` resp. `$HOME/Desktop/grestix.desktop`.

Delete directory containing restix code and Python virtual environment (the value of variable **`INSTALL_PATH`** from the installation script).

Windows

A system wide installation is removed via control panel, software/Deinstall software.

An installation for a single user create file `uninstall_restix.exe` in sub directory `AppData\Local\Programs\restix` under the user's home directory. Start the file and confirm the deletion.

Configuration

Configuration directory

Restix expects all local settings within a separate directory, default is **`.config/restix`** under the user's home directory. Optionally environment variable **`RESTIX_CONFIG_PATH`** can be set to use another location.

Configuration file

Main configuration is stored in file **`config.toml`** within the configuration directory. File format is TOML, which offers a good compromise between readability and easy machine processing. The file can be maintained either by text editor or restix' graphical user interface.

Variables

The variables below may be used in the configuration file:

`${HOME}` – replaced with the user's home directory (without trailing path separator)

`${USER}` – replaced with the username

File references

Certain settings require other files, like password or a list of elements to backup. If a settings uses a relative path to a file, it must reside within or underneath the restix configuration directory.

Path to restic executable

The restic executable to use by restix can be specified by parameter **`restic`**. Without the parameter, restix will use what's in the PATH.

```
restic = "${HOME}/bin/restic"
```

Credentials

All data backed up will be encrypted by restic. For encryption, restix offers different types shown in the examples below. Every definition must be placed in a **`credentials`** block with a unique aliasname. The aliasname is used to refer the credentials from a backup target definition.

Please observe, that automated backups are not possible if user interaction is required by the credentials setting. Use plain password or password file in that case. PGP may be an option, if you can make sure the pass phrase can be supplied by a running background agent.

Type “none” is supported, but not recommended. Data will not be encrypted in that case.

Plain text password:

```
[[credentials]]
alias = "pwd"
comment = "Plain password"
type = "password"
value = "topsecret"
```

Password file:

```
[[credentials]]
alias = "standard"
comment = "Password file"
type = "file"
value = "pwfile.txt"
```

Prompt:

```
[[credentials]]
alias = "prompt"
comment = "Request passwort"
type = "prompt"
```

PGP with pass phrase:

```
[[credentials]]
alias = "PGP passphrase"
comment = "PGP encrypted file protected with pass phrase"
type = "pgp"
value = "pgp_phrase.asc"
```

PGP with FIDO2 token:

```
[[credentials]]
alias = "PGP token"
comment = " PGP encrypted file protected with FIDO2 token"
type = "pgp"
value = "pgp_token.gpg"
```

Scopes

To specifiy the elements that shall be backed up, a block **`scope`** has to be defined. The block must use a unique aliasname. The aliasname is used to refer the scope from a backup target definition.

Parameter **`includes`** refers to a file containing all directories and files to include in a backup. This parameter is mandatory.

Optionally Parameter **`excludes`** refers to a file containg directories and files to **exclude** from a backup.

Optionally Parameter **`ignores`** takes a list of patterns. Elements matching any of the patterns are excluded from backup. If a directory matches, all elements within the directory are also excluded.

Example:

```
[[scope]]
alias = "minimal"
comment = "Minimal scope for upload to internet server"
includes = "minimal.list"
excludes = "minimal_excludes.list"
ignores = [
    ".git",
    ".idea",
    ".pytest_cache",
    ".venv",
    "__pycache__",
]
```

Backup targets

To specifiy backup targets, a block **`target`** has to be defined. The block must use a unique aliasname. The aliasname acts as a selector when using restix command line or graphical user interface.

Parameter **`location`** refers to a locally accessible directory or a URL to a server accessible by sftp protocol.

Parameter **`credentials`** holds the aliasname of the credentials to use.

Parameter **`scope`** holds the aliasname of the scope to use.

Example for a local directory:

```
[[target]]
alias = "extssd"
comment = "external SSD"
location = "/media/${USER}/58af5a30-36b5-a70683ae3e7e/restix"
scope = "full"
credentials = "standard"
```

Example for a server URL:

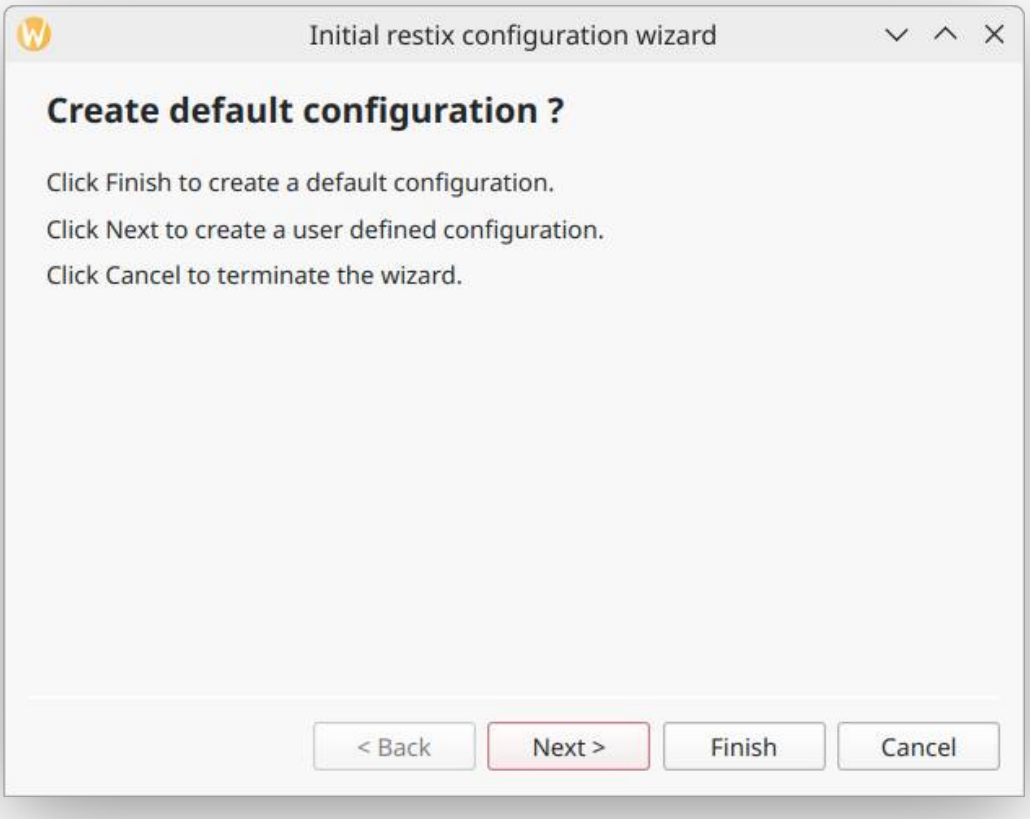
```
[[target]]
alias = "sirius"
comment = "Home Server"
location = "sftp:restic_sirius:data"
scope = "minimal"
credentials = "standard"
```

Graphical user interface

Restix' graphical user interface can be invoked from the start menu, on Linux you find it either in category Utilities or Accessoires, on Windows enter restix in the search field to find the app. Enter **`grestix`** to start the GUI from a console window.

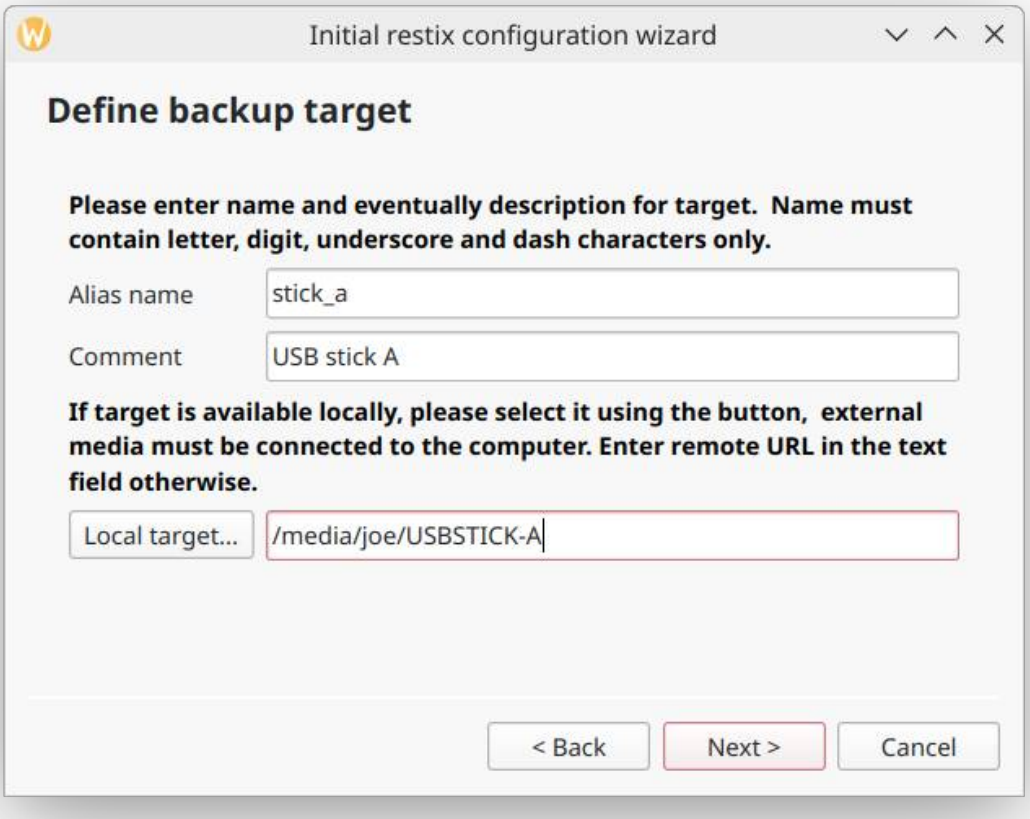
Initial configuration wizard

Immediately after installation restix is not yet configured. Restix offers a wizard, whenever the GUI is started and no configuration was found.



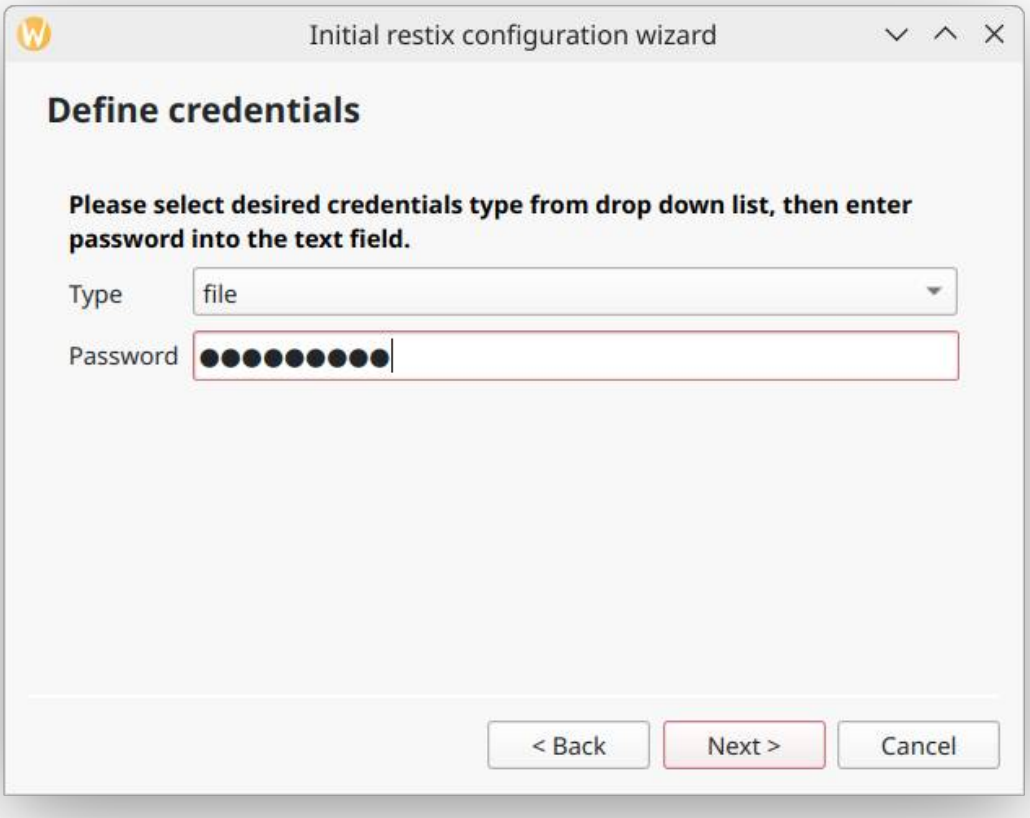
Click on the **Finish** button to create default configuration directory `$HOME/.config/restix` including a template of the restix main configuration file. After the wizard terminates the template has to be adapted according to the user's needs using the GUI's configuration button or a text editor.

Click on the **Next** button for a guided creation of a minimal restix configuration.



The wizard begins with the specification of a backup target. **Alias name** is used to select the backup target in GUI or command line and can be defined at your convenience within the allowed character range. A **comment** is optional, the GUI displays the value together with the alias name in the selection list. Specify **path** resp. **URL** in the bottom most field. If the backup target is locally accessible, its path can be selected using button **Local target**, direct input to the text field is also possible. For backup targets on a remote server the URL must be given in the text field, **sftp** is the only protocol allowed.

When you have entered all data, click on the **Next** button to proceed with the credentials.



All backups should be encrypted with a password or pass phrase. Restix offers the following alternatives:

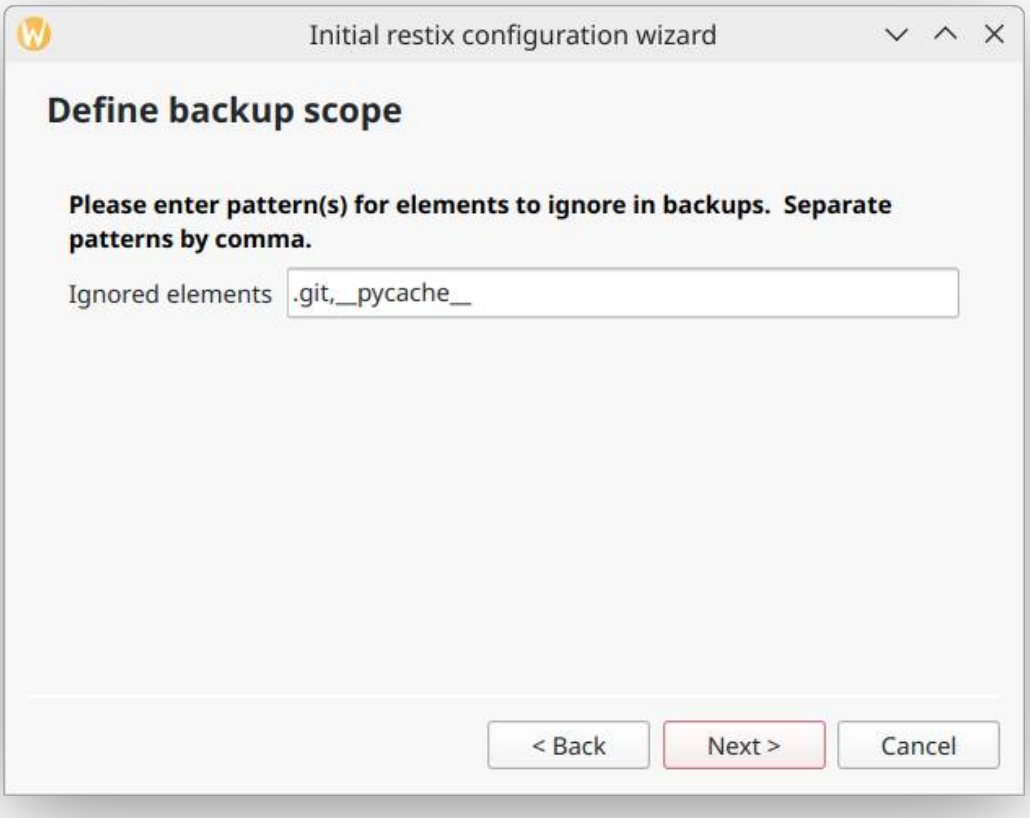
Type **file** stores the plain text password in a file, the wizard uses file `pw.txt` under the restix configuration directory.

Select type **none** to backup all data without encryption.

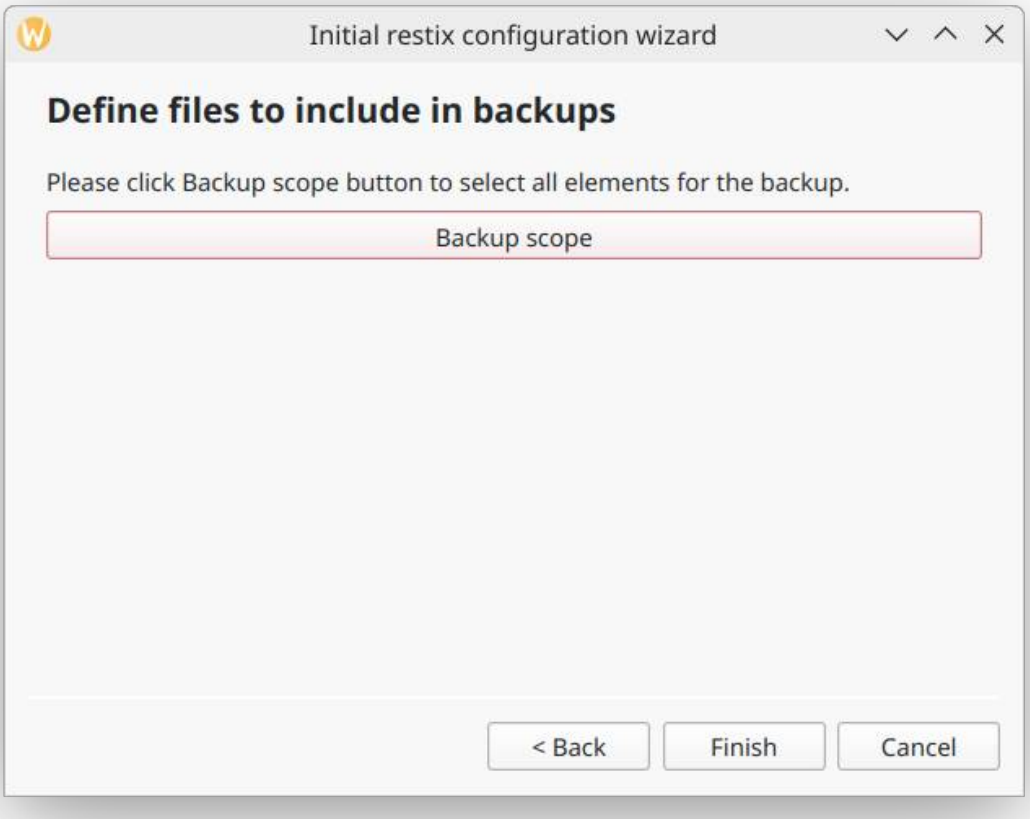
Type **pgp** encrypts the specified password with PGP and stores it in a file, the wizard uses `pw.asc` resp. `pw.gpg` under the restix configuration directory. The file is created at the end of the wizard, you have to specify E-Mail address and PGP format then. Whenever an action affected by encryption is triggered, you (or your background GPG-Agent) have to supply the PGP pass phrase.

Select type **prompt** if you want restix to ask you for the password each time it is needed.

Type **text** stores the plain text password in the restix configuration file.



The last two steps of the wizard are dedicated to the definition of the data that shall be backed up. In the first step you can define patterns for directories or files, which shall **not** be backed up, if one of the pattern matches their name. If a pattern matches a directory, all elements underneath are also excluded from the backup. The field can be left empty, of course.



Select all directories and files to back up in the last step of the wizard. Click button **Backup scope** to start a dialog, where all elements can be chosen from a file system tree viewer. Click button **Save** when you are through with the element selection. You find a detailed description of that dialog in section Scope editor.

Click button **Finish** to create a configuration from the data you've entered. Further settings can be made from the restix GUI start page, button **Configuration**.

Start screen

After invoking restix GUI the start screen is displayed. Enjoy the Gaul, who can relax since he's made backups right.

You find buttons for all restix functions offered on the top.

The button text for the currently active function is shown with yellow background.

Button **Backup** is used to transfer local data to a backup target, e.g. an external device or a server in the internet.

Button **Restore** allows data from a backup target to be restored on the local computer.

Button **Maintenance** offers functions to create and maintain backup targets.

Use button **Configuration** to edit your restix configuration.

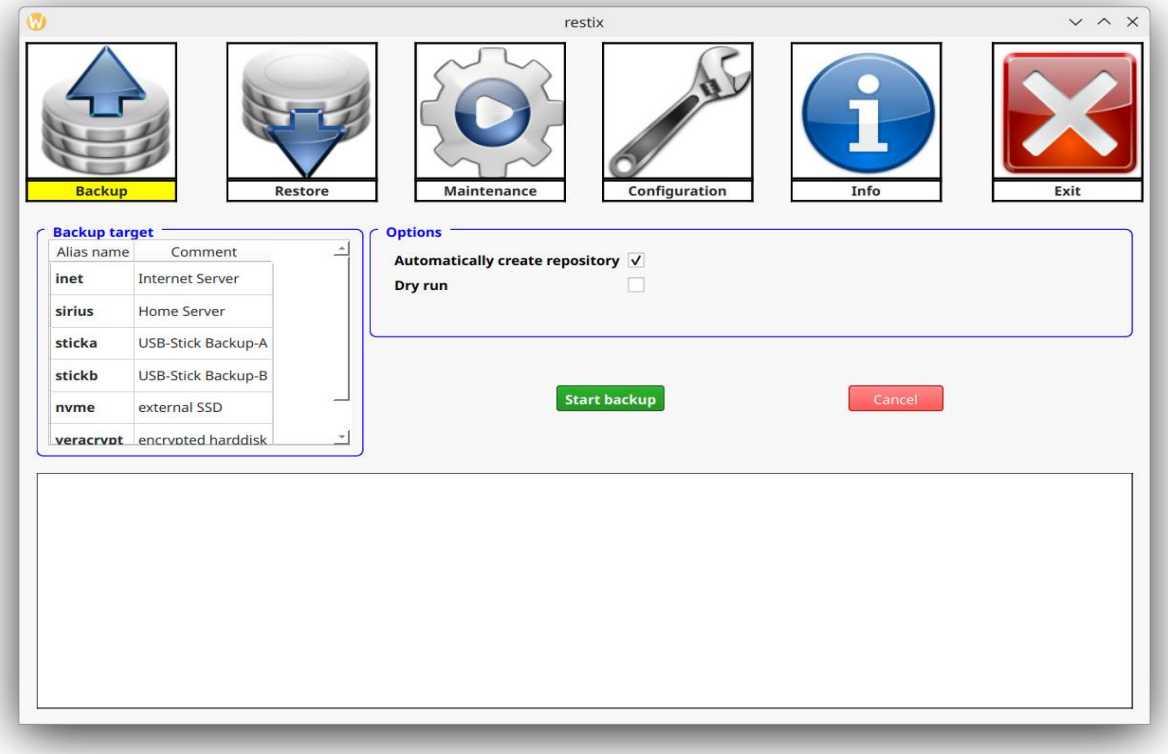
Button **Info** allows access to program details and the user manual.

Click button **Exit** to terminate restix.

Some data of the GUI like window position on the screen and size are stored in file `.restix` in the user's home directory and used for the next application start.

Backup

Click button **Backup** to display the graphical elements for a backup operation in the work area of the GUI.



Start by selecting the **Backup target** on the left with a mouse click.

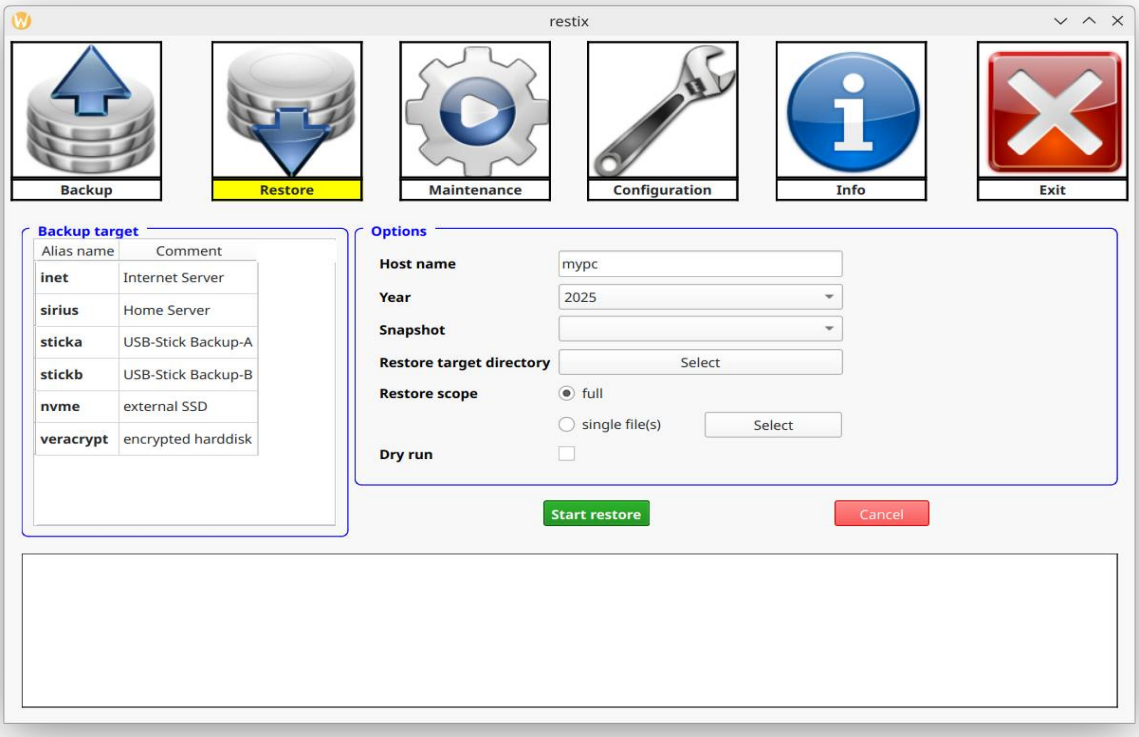
Check option **automatically create repository**, if the restic repository shall be created automatically if it doesn't exist. Please observe, that restic version 0.17 or higher is required in that case. A warning is issued otherwise.

Check option **Dry run**, if you just want to know how many elements would be backed up. There will be neither a repository created nor data transferred to the repository.

Click button **Start backup** to execute the backup. The action runs as a background task and may be aborted by clicking button **Cancel** prior to completion. The lower part of the window shows output from the restic command.

Restore

Click button **Restore** to display the graphical elements for a restore operation in the work area of the GUI.



Start by selecting the **Backup target** on the left with a mouse click.

Specify the desired restic repository with options **Host name** and **Year**. These options are preset with local hostname and current year.

Each time a backup operation is executed, restic creates a snapshot of the current contents of the repository. Select the desired **Snapshot**, the current one is displayed as **latest**.

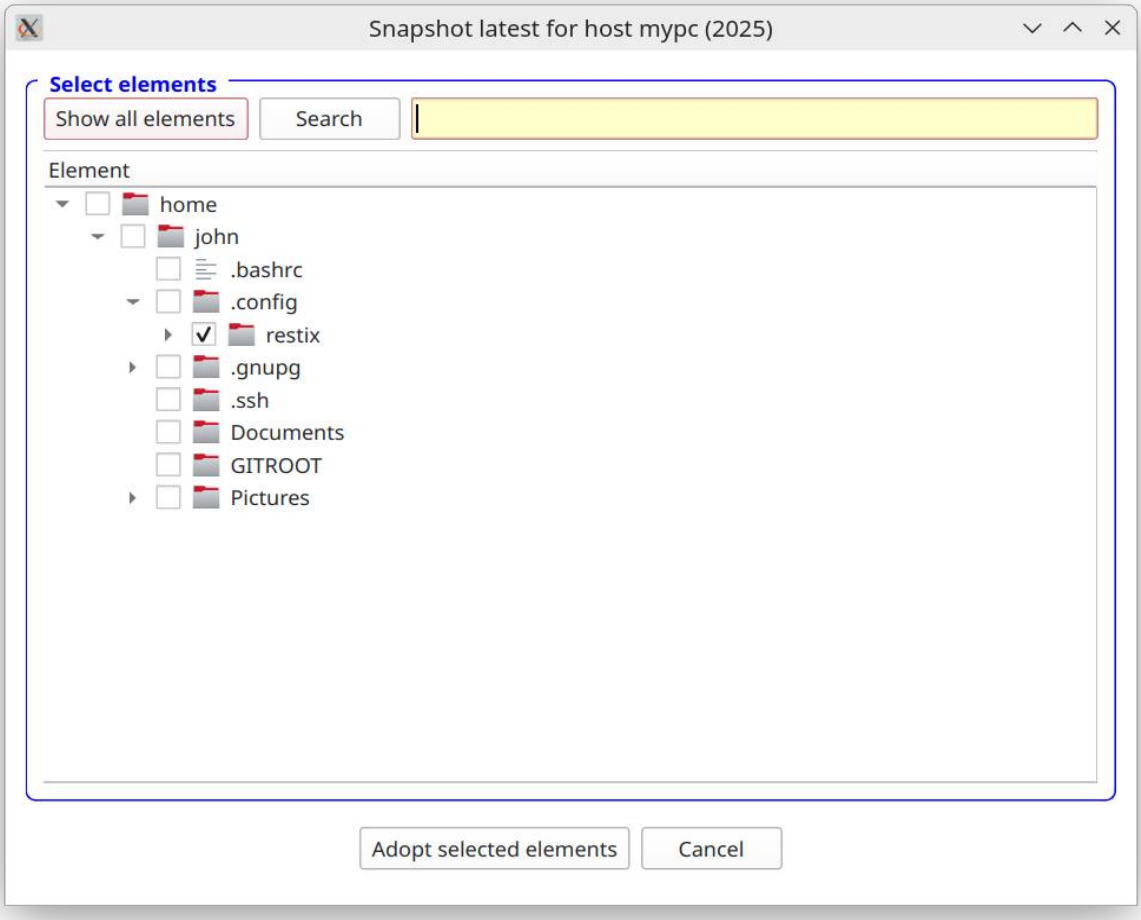
Option **Restore target directory** determines the location, where restored data shall reside. Default is file system root, i.e. all data is written to the same location as during backup. Existing data with same name is overwritten. If another directory is choosed through button **Select**, restored data will be copied underneath that directory. Example: directory `/home/user/docs` was backed up. If `/tmp` is chosen as target directory, the restored data will be copied to `/tmp/home/user/docs`.

Option **Restore scope** defines, which data shall be restored from the repository. If **full** is marked (the default), all data contained in the repository is copied to the local file system. Button **Select** next to radio button **single file(s)** allows to limit the data to be restored to a certain subset. See following section Restore single elements for details.

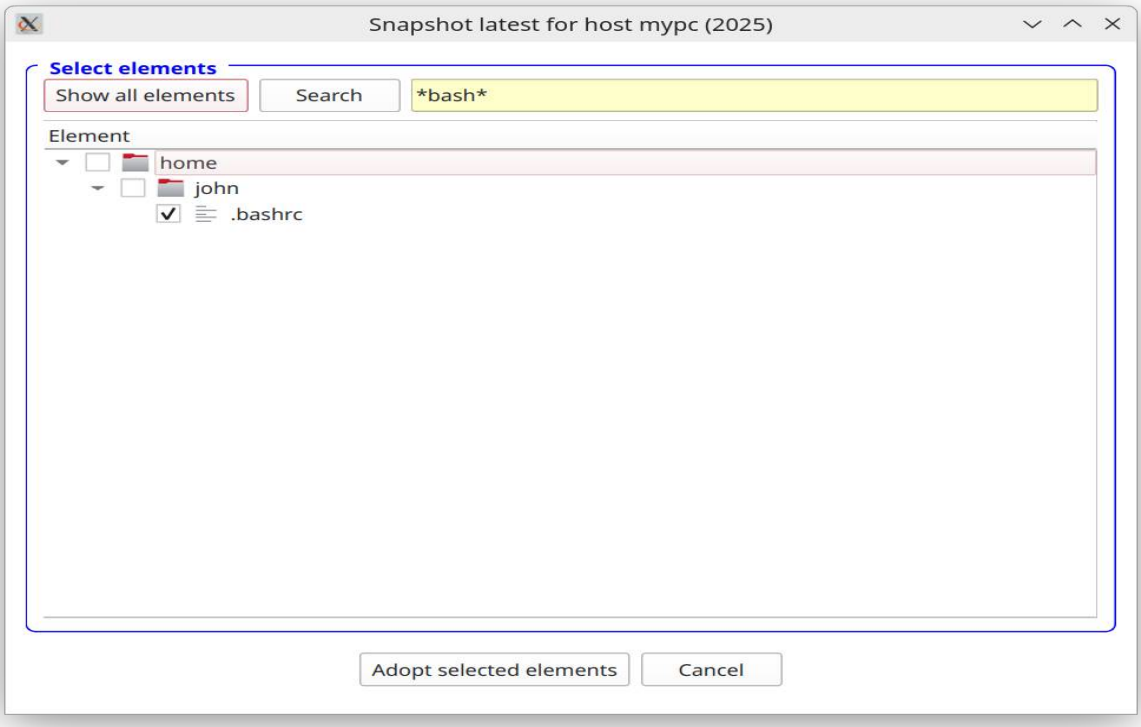
Check option **Dry run**, if you just want to see how many data would be restored. There will be no data copied from restic repository to local file system.

Click button **Start restore** to execute the restore operation. The action runs as a background task and may be aborted by clicking button **Cancel** prior to completion. The lower part of the window shows output from the restic command.

Restore single elements



Click button **Show all elements** to show all directories and files contained in the snapshot chosen. Check all elements you want to restore; for directories all elements underneath will automatically be checked, too.

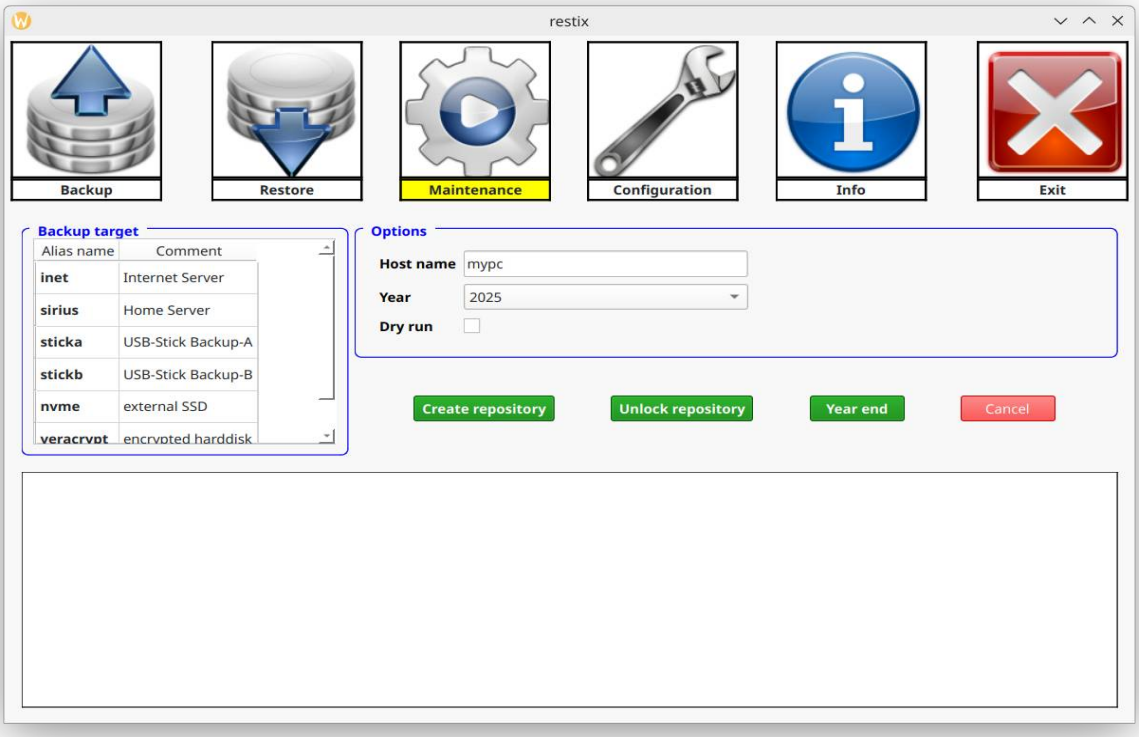


Enter a pattern into the text field on the top right and click the Search button to find elements in the snapshot. Only elements matching the search pattern will be displayed.

When all elements to be restored have been checked, click button **Adopt selected elements** to use them as a filter for the restore operation.

Maintenance

Click button **Maintenance** to display the graphical elements for restic repository maintenance in the work area of the GUI.



Start by selecting the **Backup target** on the left with a mouse click.

Option **Host name** is preset with the local hostname. The value can be changed, if the repository for another host shall be maintained.

Option **Year** is preset with the current year. The year can be changed, e.g. if a year end action shall be executed for the previous year.

Check option **Dry run**, if you just want to see, what would be changed in the repository. There will be no data changed in the repository.

Create restic repository

To create a new restic repository for a hostname and a year, click button **Create repository**. The lower part of the window shows output from the restic command.

Unlock restic repository

Sometimes it can happen that restic doesn't release a lock after an operation. To enforce the release of the lock, click button **Unlock repository**. The lower part of the window shows output from the restic command.

Archive restic repository at year end

It's a good idea to archive a repository around new year. Restic can reduce the size of a repository by deleting all snapshots but the latest of every month. Click button **Year end** to remove other snapshots from a repository. The lower part of the window shows output from the restic command.

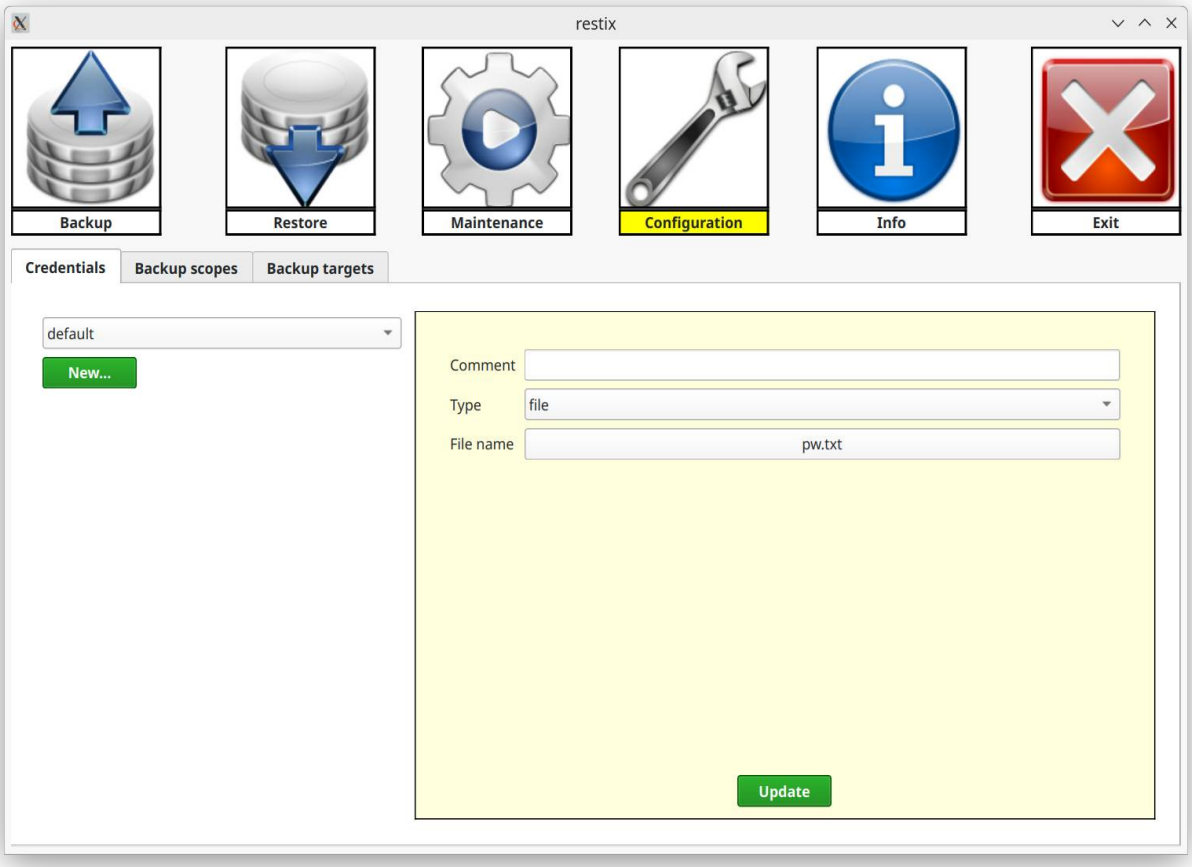
Later on, the repository can be archived on an external device or a remote server.

Configuration

Click button **Configuration** to display the graphical elements to configure restix in the work area of the GUI.

Please observe that changes are kept program internally until another action button is clicked or you want to terminate the application. Then you will be asked whether the changes shall be written to configuration file.

Edit credentials



Make sure tab **Credentials** on the left hand side is active, click on it if necessary. Select credentials' alias name in the drop down list on the left.

Comment for the credentials can directly be typed in the appropriate text field.

Edit credentials type

Select desired type from the drop down list.

Type **file** stores the plain text password in a file, its file name must be selected with the button next to caption File name.

Select type **none** to backup all data without encryption.

Type **pgp** encrypts the specified password with PGP and stores it in a file. Either select an existing file using the button to the right of caption File name, or create a new file using button **Create encrypted password file** (see section Create PGP encrypted password file).

Select type **prompt** if you want restix to ask you for the password each time it is needed.

Type **text** stores the plain text password in the restix configuration file, enter the password in the appropriate text field.

Change password

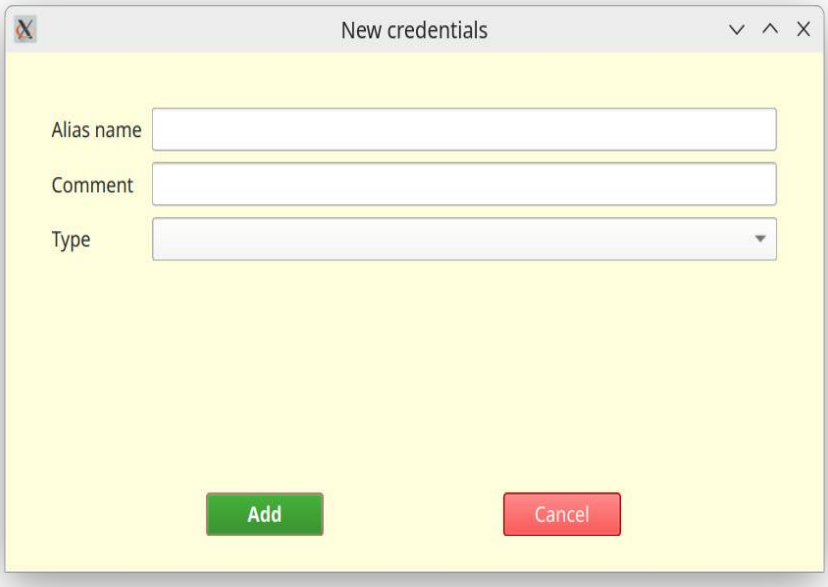
For type **file** either select another file with the button next to caption File name or change the contents of the file with an editor.

For type **text** enter the new password into the appropriate text field.

For type **pgp** either select another file using the button to the right of caption File name, or create a new file using button **Create encrypted password file** (see section Create PGP encrypted password file).

Create new credentials

Click button **New...** underneath the drop down list of the credentials alias names.



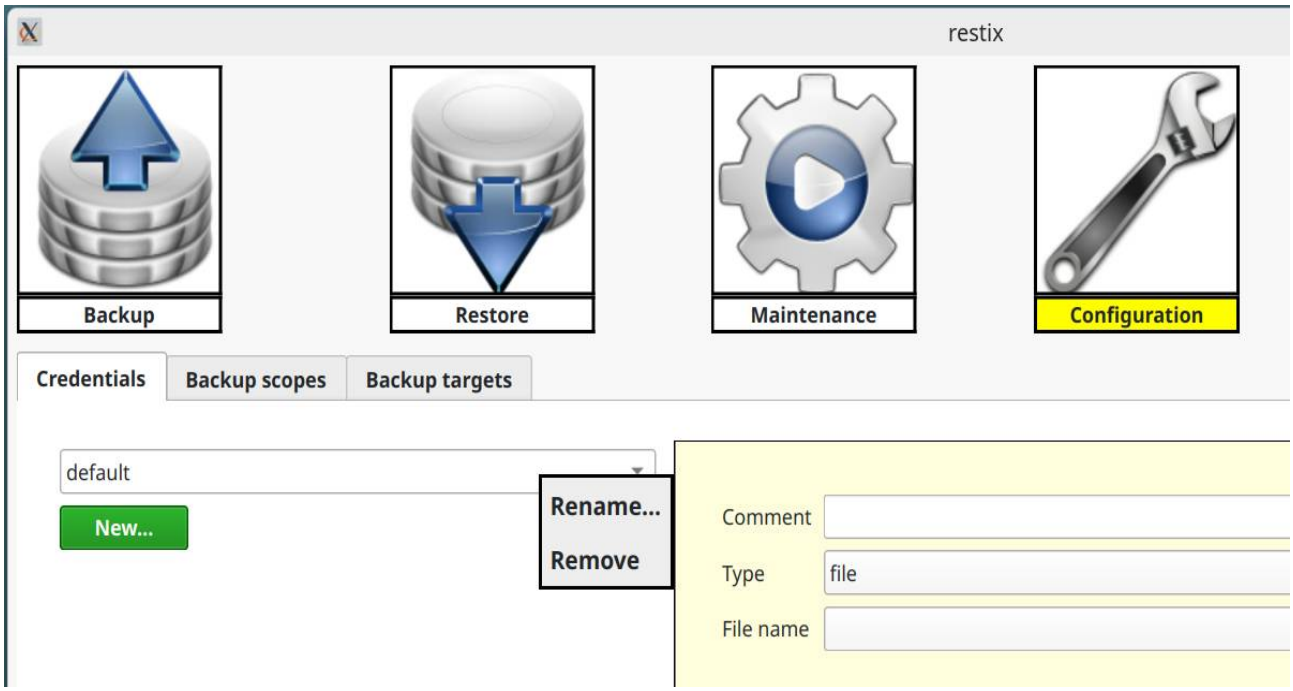
Specify a name for the credentials in text field Alias name, the name must not exist yet in your restix configuration. Only letters, digits, underscores and dashes are allowed for alias names.

Optional enter a short comment for the credentials.

See section Edit credentials for a description on the possible credentials types.

Change alias name of credentials

Select desired credentials from the drop down list on the left hand side, then right click the alias names to open a context menu. Select menu entry **Rename** and enter the new name.

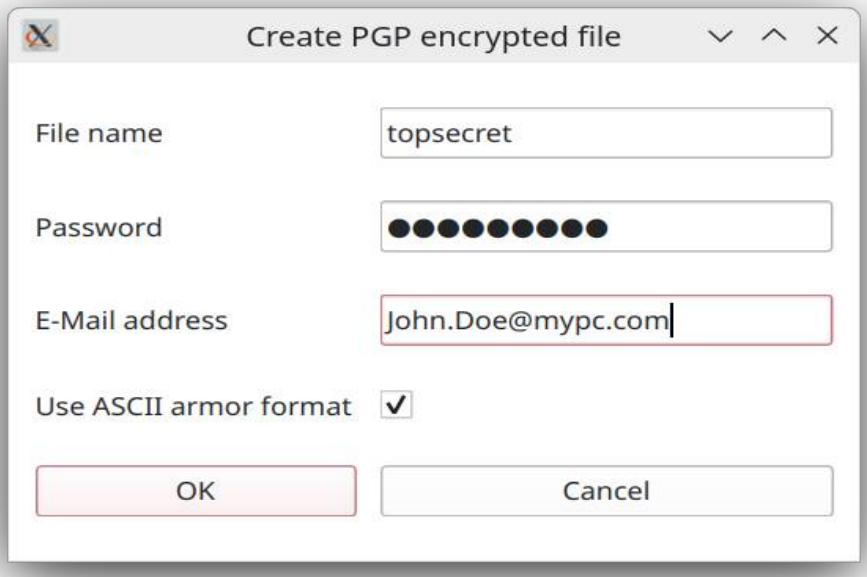


Delete credentials

Select desired credentials from the drop down list on the left hand side, then right click the alias names to open a context menu. Select menu entry **Remove** and confirm. A deletion is not possible, if the credentials are still in use by one or more backup targets.

Create PGP encrypted password file

To make use of this functionality, GnuPG must be installed on your system (GPG4Win on Windows). The PGP key to be used for the restix password must already exist. The PGP key is referenced by its Email address, hence you must specify it when creating the password file.



First enter desired file name **without suffix**. If a relative path is specified, the file will be created in restix configuration directory.

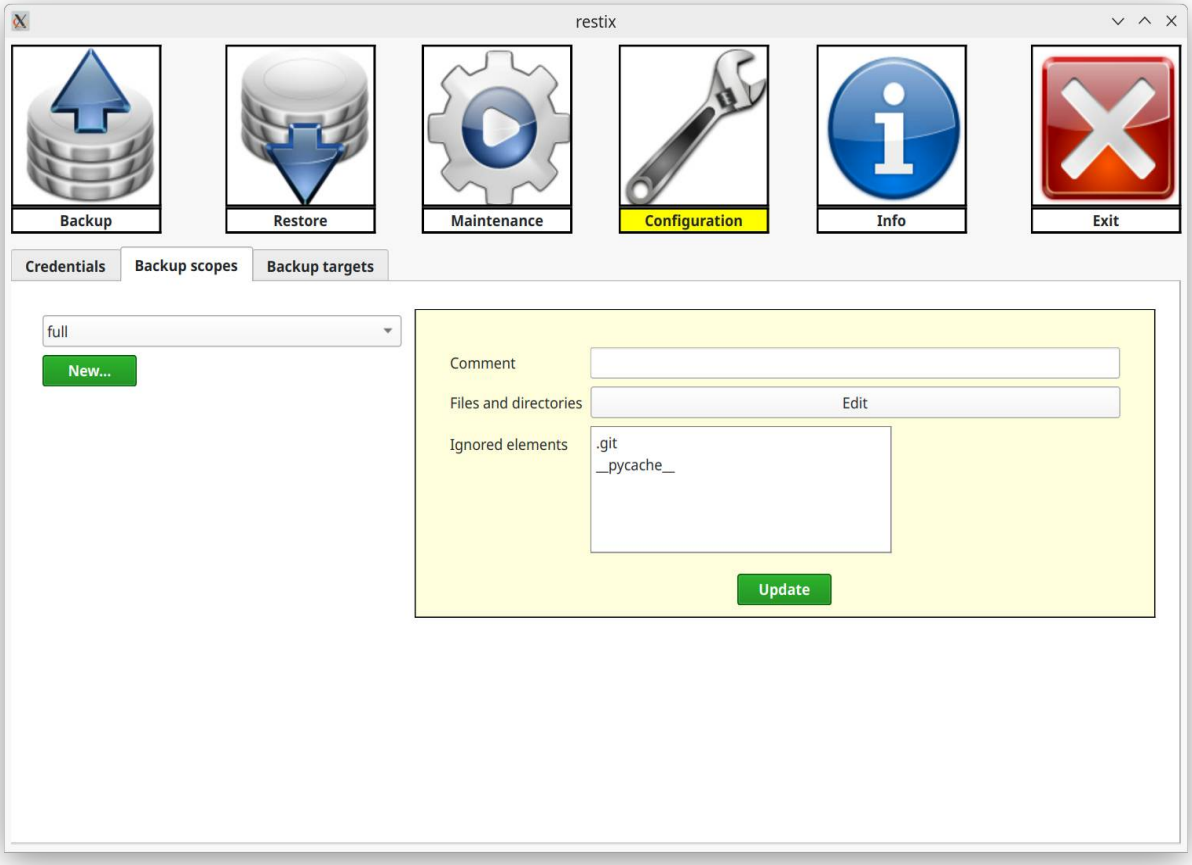
Enter desired password for your backed up data.

Enter E-Mail address for PGP key to use.

Check **Use ASCII armor format**, if you want the encrypted file in ASCII format, otherwise a binary file is created.

Click button **OK** to create a file containing the encrypted restix password; you don’t have to specify your PGP passphrase for this. ASCII armor formatted files receive suffix `.asc`, binary files `.gpg`. For the data shown above file `topsecret.asc` would be created inside restix configuration directory.

Edit backup scope



Make sure tab **Backup scopes** on the left hand side is active, click on it if necessary. Select scope alias name in the drop down list on the left.

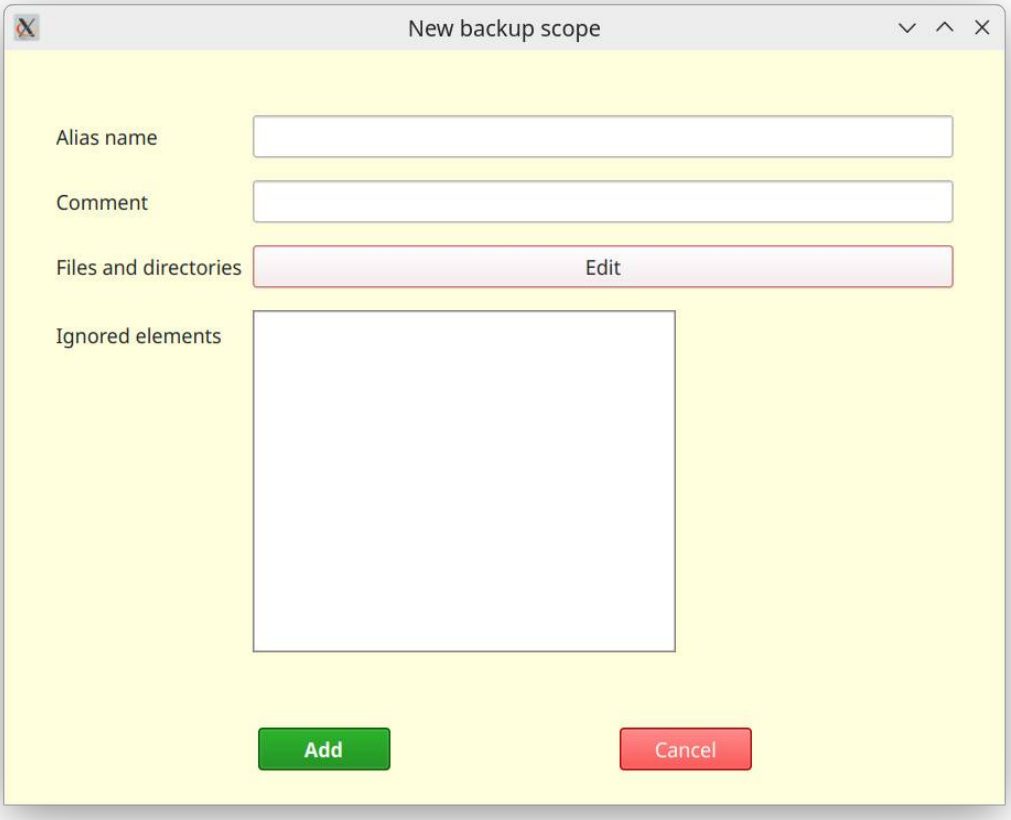
Comment for the backup scope can directly be typed in the appropriate text field.

Files and directories to backup can be specified with a file system tree viewer by clicking **Edit**, see section Scope editor.

Patterns for files and directories which shall always be excluded from backup can be entered into list Ignored elements. Pattern format follows restic specification (e.g. for pattern `*.so` all elements with suffix `.so` will be ignored). If a directory matches, all elements underneath the directory are also ignored.

Create new backup scope

Click button **New...** underneath the drop down list of the backup scope alias names.



Specify a name for the backup scope in text field Alias name, the name must not exist yet in your restix configuration. Only letters, digits, underscores and dashes are allowed for alias names.

Optional enter a short comment for the backup scope.

Refer to Edit backup scope on how to specify files and directories and ignored elements.

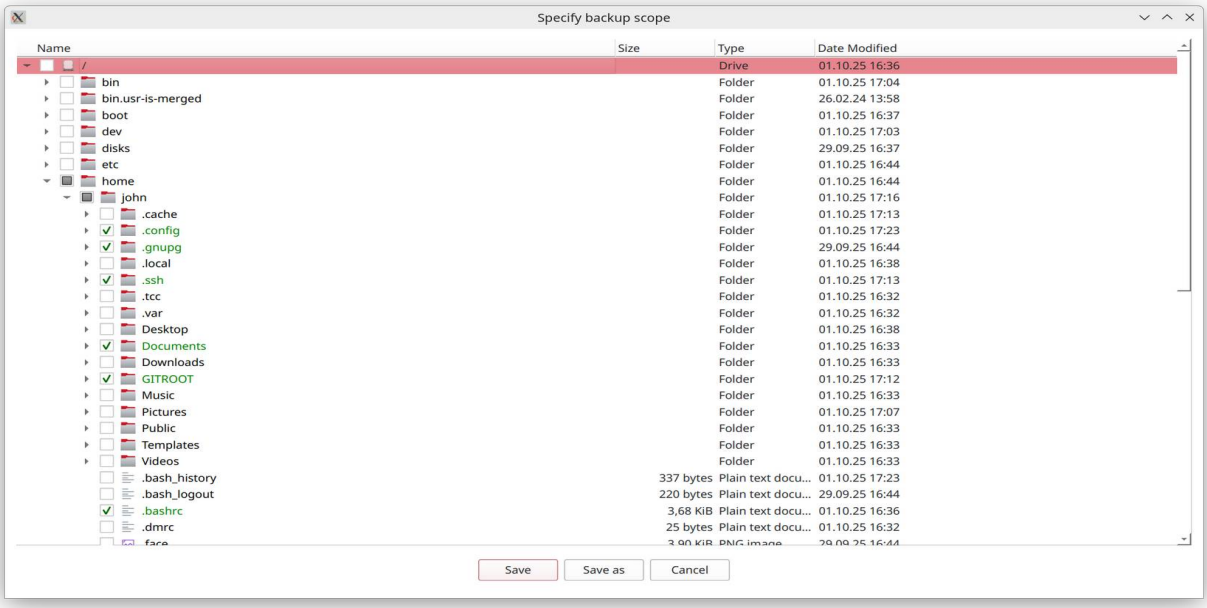
Change alias name of backup scope

Select desired backup scope from the drop down list on the left hand side, then right click the alias names to open a context menu. Select menu entry **Rename** and enter the new name.

Delete backup scope

Select desired backup scope from the drop down list on the left hand side, then right click the alias names to open a context menu. Select menu entry **Remove** and confirm. A deletion is not possible, if the scope is still in use by one or more backup targets.

Scope editor



A check mark to the left of an element indicates, that it is included in backups.

A square indicates, that there are elements being part of backups somewhere underneath that element.

The topmost element of a partial tree included in backups is shown in green color.

Elements explicitly excluded from backups are shown in red color.

Via pattern excluded elements (ignored elements) are grayed out.

Immutable elements

File system root and ignored elements cannot be checked or unchecked.

For elements explicitly excluded from backups (shown in red) only the topmost element of a subtree can be checked.

Add element to backup scope

Only elements, where the element itself and all parent elements are not included in backups, may be added to a scope. This can be easily recognized from the black color of the element name and the square to the left of the element being either empty or containing a square.

Check the element in the tree viewer.

The element itself will be shown in green color, all child elements will get a checkmark and be included in backups, too.

Remove element from backup scope

Only elements where all parent elements are not part of the backup can be removed, i.e. the element name must be displayed in green color.

Uncheck the element.

The element itself will then be shown with black name and the checkmark removed. The checkmark will also be removed for all child elements, if any.

Exclude element from backup scope

Only elements where the parent is currently part of the backup can be excluded, i.e. the element name must be displayed in black color and the checkbox to the left of the name must contain a checkmark.

Uncheck the element.

The element itself and all child elements will be displayed in red color with an empty checkbox to the left of their names.

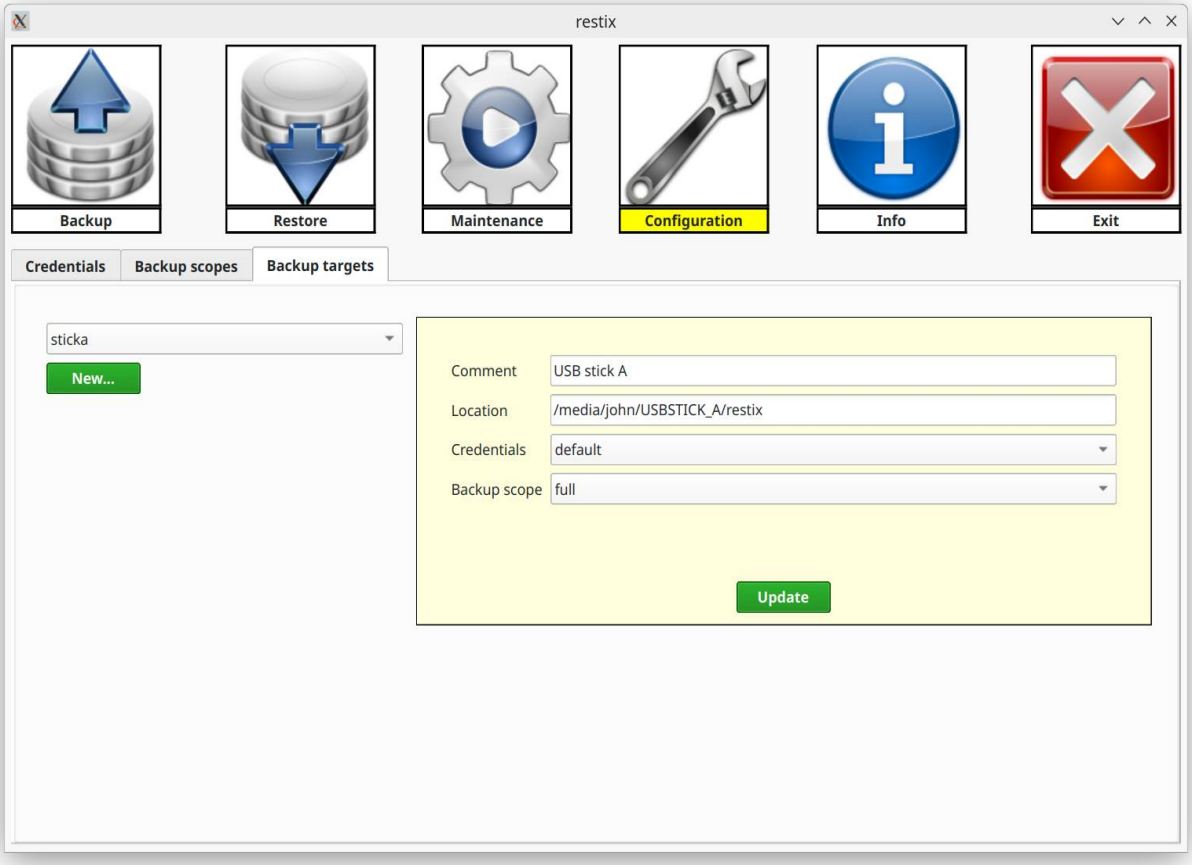
Add prematurely excluded element to backup scope

An “undo exclusion” can only be done for elements, where the direct parent is included in the backup. The element name is displayed red and the direct parent element name in black in that case.

Check the element.

The element itself and all child elements will be displayed in black color and the checkboxes to the left of their names will contain a checkmark.

Edit backup target



Make sure tab **Backup targets** on the left hand side is active, click on it if necessary. Select target alias name in the drop down list on the left.

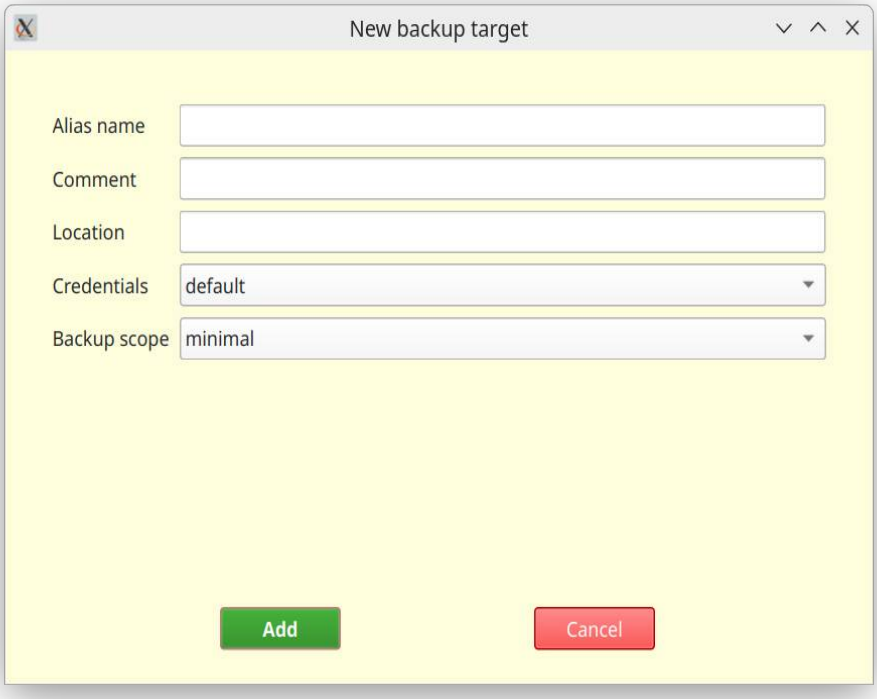
Comment and **location** of the backup target can directly be typed in the appropriate text field.

Select the alias name for the **credentials** to use from the drop down list.

Select the alias name for the **backup scope** to use from the drop down list.

Create new backup target

Hierzu den Button **Neu...** unterhalb der Auswahlliste mit den Sicherungszielen klicken.



Specify a name for the backup target in text field Alias name, the name must not exist yet in your restix configuration. Only letters, digits, underscores and dashes are allowed for alias names.

Optional enter a short **comment** in the appropriate text field.

Enter **location** of the backup target in the appropriate text field.

Select alias name of **credentials** to use for the backup target from the drop down list.

Select alias name of **backup scope** to use for the backup target from the drop down list.

Change alias name of backup target

Select desired backup target from the drop down list on the left hand side, then right click the alias names to open a context menu. Select menu entry **Rename** and enter the new name.

Delete backup target

Select desired backup target from the drop down list on the left hand side, then right click the alias names to open a context menu. Select menu entry **Remove** and confirm.

Program information

A click on button **Info** with either left or right mouse button opens a context menu, where you can select from the entries User manual and About restix.

Menu entry **User manual** displays the user manual in the language set for the local computer, at the moment English and German are available.

Meny entry **About restix** displays a dialog windows with informations about version and copyright for restix and all external libraries used.

Command line interface

Restix CLI is invoked with command **`restix`** in a terminal. This interface is primarily intended to be used for automated tasks.

Commands

Show restix version

`restix --version` displays the version installed: `restix-Version 0.9.6`

Show backup targets

`restix targets` displays alias name and description of all backup targets defined in restix configuration file. If no configuration file can be found, an error message is displayed.

Example:

```
restix targets

Backup targets:
homeserver – Home-Server
sticka – USB Stick Backup-A
stickb – USB Stick Backup-B
```

Create repository

**`restix init a`*`liasname`* creates a new restic repository under backup target with specified alias name. The repository is created in sub directory username/hostname/year. The repository is encrypted according to the backup targets credential settings in the configuration file.

Option **`--batch`** immediately executes the command; without the option the command must be confirmed by the user.

Example for user ***joe***, nost ***notebook***, year ***2025***, backup target ***sticka*** with URL ***`/media/joe/Backup-A/restix`***:

```
restix init --batch sticka
```

will create a restic repository without confirmation prompt in `/media/joe/Backup-A/restix/joe/notebook/2025`

Backup

`restix backup aliasname` transfers local data to the backup target with specified alias name. The data is encrypted according to the backup targets credential settings in the configuration file. Which data is transferred is determined from the backup target's scope settings in the configuration file.

Option **`--batch`** immediately executes the command; without the option the command must be confirmed by the user.

Option **`--dry-run`** displays information what would be backed up, but doesn't transfer any data to restic repository.

Option **`--auto-create`** automatically creates a new restic repository, if it doesn't exist. Option is applicable for restic version 0.17 and above.

Restore

`restix restore aliasname` copies data from backup target with specified aliasname to local file system. For data decryption the credential settings in the configuration file are used.

Option **`--batch`** immediately executes the command; without the option the command must be confirmed by the user.

Option **`--dry-run`** displays information what would be restored, but doesn't transfer any data from restic repository to local file system.

Option **`--snapshot`** specifies the snapshot ID in the restic repository acting as source for the restore operation. Snapshot ID **`latest`** is used as default.

Option **`--restore-path`** specified the root directory in local file system, where to store the restored data. Default setting is file system root, i.e. all data is restored to its original location.

Option **`--host`** specifies, that data shall be copied from the restic repository for that hostname. Default is name of local host.

Option **`--year`** specifies, that data shall be copied from the restic repository for that year. Default is current year.

Show snapshots in backup target

`restix snapshots aliasname` displays all snapshots in the backup target. For data decryption the credential settings in the configuration file are used.

Option **`--host`** specifies, that the restic repository for that hostname shall be used. Default is name of local host.

Option **`--year`** specifies, that the restic repository for that year shall be used. Default is current year.

List files in backup target

`restix ls` ***`aliasname`*** displays all files in the backup target with specified aliasname. For data decryption the credential settings in the configuration file are used.

Option **`--snapshot`** specifies the snapshot ID in the restic repository to be used. Snapshot ID ***`latest`*** is used as default.

Option **`--host`** specifies, that the restic repository for that hostname shall be used. Default is name of local host.

Option **`--year`** specifies, that the restic repository for that year shall be used. Default is current year.

Search for files in backup target

`restix find` ***`aliasname`*** displays all elements in the backup target with specified aliasname matching a given search pattern. For data decryption the credential settings in the configuration file are used.

Option **`--pattern`** specifies the search pattern for the elements to find. Option is mandatory. Pattern syntax is the same as used by restic.

Option **`--snapshot`** specifies the snapshot ID in the restic repository to be used. Snapshot ID ***`latest`*** is used as default.

Option **`--host`** specifies, that the restic repository for that hostname shall be used. Default is name of local host.

Option **`--year`** specifies, that the restic repository for that year shall be used. Default is current year.

Unlock repository

`restix unlock` ***`aliasname`*** unlocks a restic repository in the backup target with specified aliasname. For data decryption the credential settings in the configuration file are used. Sometimes restic doesn’t remove a lock after a command, then it’s time to issue this command.

Option **`--batch`** immediately executes the command; without the option the command must be confirmed by the user.

Option **`--host`** specifies, that the restic repository for that hostname shall be used. Default is name of local host.

Option **`--year`** specifies, that the restic repository for that year shall be used. Default is current year.

Cleanup repository

`restix cleanup` ***`aliasname`*** removes all snapshots in the backup target with specified name but the latest of each month. For data decryption the credential settings in the configuration file are used. The command is useful at the end of a year to reduce the size of a repository prior to archiving.

Option **`--batch`** immediately executes the command; without the option the command must be confirmed by the user.

Option **`--dry-run`** displays information which snapshots would be removed, but doesn't make any changes to the respository.

Option **`--host`** specifies, that the restic repository for that hostname shall be used. Default is name of local host.

Option **`--year`** specifies, that the restic repository for that year shall be used. Default is current year.

Setting up a server for restic backups

Finally some hints how to configure a server at home or in the internet for the use with restic. All commands below refer to a Debian Linux server, the commands may differ on other derivates.

All commands must be executed with root permissions.

Create user for restic

The user doesn't need shell and home directory, the user ID set below is optional.

```
adduser --uid 2000 --shell /bin/false restic
```

Create directory for restic backups

In the example below, `/var/restic` is used root directory. Below subdirectory `.ssh` receives the public ssh keys of all users being allowed to use the server for backups. Subdirectory `data` serves as backup target, this is where all restic repositories are created.

```
cd /var
mkdir restic
chmod 755 restic
mkdir .ssh data
chown restic:restic .ssh data
chmod 700 .ssh data
```

Transfer the **public** ssh keys (file ends with `.pub`) of all users to subdirectory `/var/restic/.ssh`. Execute the command below for all users, replace "user" with the actual username. Depending on the encryption algorithm `id_ed25519` must be changed, e.g. to `id_rsa`.

```
cat ~user/.ssh/id_ed25519.pub >> /var/restic/.ssh/authorized_keys
```

Finally update access rights as follows:

```
chmod 400 /var/restic/.ssh/authorized_keys
```

Update SSH server configuration

For enhanced security, access can optionally be limited to a certain group. If you decide to do so, execute the following commands. The usermod command must be executed for **all users** allowed to access the server with ssh, including the restic user. The group ID is optional.

```
groupadd -g 3000 sshusers
usermod -aG sshusers user
```

Update ssh daemon configuration `/etc/ssh/sshd_config` as shown below. The first three lines should already be present with their values, the latter lines must be inserted or updated. The line starting with `Subsystem sftp` … has a default value of `/usr/lib/openssh/sftp-server`, this **must** be changed to `internal-sftp`.

```
PermitRootLogin no
PasswordAuthentication no
PubkeyAuthentication yes
AllowGroups sshusers
Subsystem sftp internal-sftp
Match User restic
  AuthorizedKeysFile /var/restic/.ssh/authorized_keys
  ForceCommand internal-sftp
  ChrootDirectory /var/restic
```

Check whether all changes have been entered correctly, the following command must not show any errors:

```
sshd -t
```

Finally restart ssh daemon:

```
systemctl restart sshd
```

Update SSH client configuration

When accessing the server from a client, define a host alias in the user's **`$HOME/.ssh/config`** file. Use that alias to specify the server URL in the restix backup target definition. Example:

```
Host restic_sirius
 User restic
 Hostname sirius
 Port 22
 PreferredAuthentications publickey
 IdentityFile "/home/user/.ssh/id_ed25519"
```