

Contents

1	NAME	2
2	DESCRIPTION	2
3	PUBLIC FUNCTIONS	2

1 NAME

OpenSLX::Utils - provides utility functions for OpenSLX.

2 DESCRIPTION

This module exports utility functions, which are expected to be used all across OpenSLX.

3 PUBLIC FUNCTIONS

copyFile(\$fileName, \$targetDir, \$targetFileName)

Copies the file specified by *\$fileName* to the folder *\$targetDir*, preserving the permissions and optionally renaming it to *\$targetFileName* during the process.

If *\$targetDir* does not exist yet, it will be created.

fakeFile(\$fullPath)

Creates the (empty) file *\$fullPath* unless it already exists.

If the parent directory of *\$fullPath* does not exist yet, it will be created.

linkFile(\$linkTarget, \$linkName)

Creates the link *\$linkName* that points to *\$linkTarget*.

If the directory where the new link shall live does not exist yet, it will be created.

slurpFile(\$fileName, \$flags)

Reads the file specified by *<\$fileName>* and returns the contents.

The optional hash-ref *\$flags* supports the following entries:

failIfMissing

Specifies what shall happen if the file does not exist: die (*failIfMissing* == 1) or return an empty string (*failIfMissing* == 0). Defaults to 1.

io-layer

Specifies the Perl-IO-layer that shall be applied to the file (defaults to 'utf8').

spitFile(\$fileName, \$content, \$flags)

Writes the given *\$content* to the file specified by *<\$fileName>*, creating the file (and any missing directories) if it does not exist yet.

The optional hash-ref *\$flags* supports the following entries:

io-layer

Specifies the Perl-IO-layer that shall be applied to the file (defaults to 'utf8').

mode

Specifies the file mode that shall be applied to the file (via chmod).

appendFile(\$fileName, \$content, \$flags)

Appends the given *\$content* to the file specified by <\$fileName>, creating the file (and any missing directories) if it does not exist yet.

The optional hash-ref *\$flags* supports the following entries:

io-layer

Specifies the Perl-IO-layer that shall be applied to the file (defaults to 'utf8').

followLink(\$path, \$prefixedPath)

Deeply traverses the given *\$path* until it no longer contains a link and returns the resulting file or directory.

If you pass in a *\$prefixedPath*, each link will be resolved relatively to that path (useful for example with respect to chroot-environments).

copyBinaryWithRequiredLibs(\$params)

Copies a binary to a specified folder, taking along all the libraries that are required by this binary.

The hash-ref *\$params* supports the following entries:

binary

The full path to the binary that shall be copied.

targetFolder

The full path to the folder where the binary shall be copied to.

libTargetFolder

Defines a path relatively to which all required libs will be copied to.

An example: during execution of

```
copyBinaryWithRequiredLibs({
  binary      => '/bin/ls',
  targetFolder => '/tmp/slx-initramfs/bin',
  libTargetFolder => '/tmp/slx-initramfs',
});
```

the library `lib/libc-2.6.1.so` will be copied to `/tmp/slx-initramfs/lib/libc-2.6.1.so`.

targetName [optional]

If you'd like to rename the binary while copying, you can specify the new name in this entry.

unshiftHereDoc(\$content)

Returns the here-doc (or string) given in *\$content* such that the leading whitespace found on the first line will be removed from all lines.

As an example: if you pass in the string

```
#!/bin/sh
if [ -n "$be_nice" ]; then
    echo "bummer!" >/etc/passwd
fi
```

you will get this:

```
#!/bin/sh if [ -n "$be_nice" ]; then echo "bummer!" >/etc/passwd fi
```

string2Array(\$string)

Returns the given string split into an array, using newlines as separator.

In the resulting array, empty entries will have been removed and each entry will be trimmed of leading or trailing whitespace and comments (lines starting with a #).

chrootInto(\$osDir)

Does a `chroot()` into the given directory (which is supposed to contain at least the fragments of an operating system).

mergeHash(\$targetHash, \$sourceHash, \$fillOnly)

Deeply copies values from *\$sourceHash* into *\$targetHash*.

If you pass in 1 for *\$fillOnly*, only hash entries that do not exist in *\$targetHash* will be copied (**Merge-mode**), otherwise all values from *\$sourceHash* will be copied over (**Push-mode**).

Returns the resulting *\$targetHash* for convenience.

getFQDN()

Determines the fully-qualified-domain-name (FQDN) of the computer executing this function and returns it.

readPassword(\$prompt)

Outputs the given *\$prompt* and then reads a password from the terminal (trying to make sure that the characters are not echoed in a readable form).

hostIs64Bit()

Returns 1 if the host (the computer executing this function) is running a 64-bit OS, 0 if not (i.e. 32-bit).

getAvailableBusyboxApplets()

Returns the list of the applets that is provided by the given busybox binary.

grabLock()

pathOf()

Returns the path of a binary it is installed in.

isInpath()

Returns whether a binary is found.