

# D2Archive

## D2Archive utility

**D2Archive** compresses and decompresses the archives \*.d2a, which are used as the archives of [application modules](#).

### Usage:

```
d2archive <command> [<switches>...] <archive_name> [<directory>]
```

### <Commands>

c:	Compresses a content of the directory.
l:	Shows an archive content.
x:	Extracts the archive content into some directory.

### <Switches>

-b:	A password is binary (256-bits) and uses Base64 encoding.
-p{password}	Sets a password.

## Usage of D2Archive utility

D2archive compresses and encodes any files. It always works with the content of one directory - there is possible to compress only the whole directory including the subdirectories. It decompresses the files as well as and shows the content of archive. The created archive can be protected by some password.

The password modes:

- 256-bits binary password with Base64 encoding (e.g. for AM) - with switch "-b".
- Standard text - from the text is calculated 256-bit hash which is used on encoding.

If the archive is encoded you need the password to show the archive content - also an information on files are encoded (name, size, modify time, ...).

## Conditions for proper AM archive

- Directory containing the exported application module must contain the exported object of *Application module* type in its root. Other objects can be in any subdirectory.
- Name of final AM archive (without suffix .d2a) must be the same as the name of object of *Application module* type. Also XML file of this object must have the same name (without suffix .xml).
- The password for the application module is in licence program or the archive may be created without password.

### Creating of AM archive:

```
d2archive c -b -pPASSWORD_FOR_AM archive_name directory_of_module
```

**Note:** The switch -p is not followed by space! Archive name may be written without suffix - ".d2a" is added automatically.

### Decompressing of AM archive:

```
d2archive x -b -pPASSWORD_FOR_AM archive_name directory_of_module
```



#### Related pages:

[Application modules](#)