

# Adding a set of files

## The <fileset> element

The <fileset> tag is a nested element to the <pack> element and allows files to be added to that [pack](#) by selecting them similar like in an Apache Ant fileset.

### Attributes

Attribute	Description	Required	Values (Default)
dir	A base directory for the fileset (relative paths are treated against the compiler base directory)	yes	
file	A single file to copy, can be relative to the directory specified in <i>dir</i> . If not given, the whole directory <i>dir</i> is recursively included, filtered by the include and exclude expressions below.	no	
targetdir	the destination path, works like for <file>	no	("\${INSTALL_PATH}")
casesensitive	optionally lets you specify if the names are case-sensitive or not - takes yes or no	no	
<del>os</del>	<del>specifies the operating system, works like for &lt;file&gt;</del>	<del>no</del>	<del>"unix"   "windows"   "mac"</del>
override	Whether to overwrite existing files. Use <code>asktrue</code> or <code>askfalse</code> if the user should be interactively asked what to do and supply default value for non-interactive use. Another possible value is <code>update</code> . It means that the new file is only installed if its modification time is newer than the modification time of the already existing file (note that this is not a reliable mechanism for updates - you cannot detect whether a file was altered after installation this way.)	no	"true"   "false"   "asktrue"   "askfalse"   "update" ("update")
overrideRenameTo	Globmapper to rename a conflicting file to. This works similar like the <globmapper> in <a href="#">File Name Mappers</a> , whereby the mapper's from attribute is set to "" and the to attribute exactly to the value given here. Example ".bak" will rename the target file by appending the suffix .bak before overwriting it. The override attribute must be set "true" to activate this feature.  Since IzPack 5.0	no	String - valid globmapper target expression

<p>blockable</p>	<p>For Windows only, ignored on non-Windows systems:          Defines whether and how blocked target files on Windows should be handled. This might result in pending file operations which require a system reboot.</p> <p>Pending file operations are introduced during the installation in two phases:</p> <ol style="list-style-type: none"> <li>1. The <code>blockable</code> attribute marks files pending to be replaced after a system reboot if they are blocked. Blocked files are recognized during physically installing them.</li> <li>2. How and whether the system should be really rebooted directly from the installer in case there are such pending file operations at the end of the installation can be controlled by the <code>&lt;rebootaction&gt;</code> tag nested to <code>&lt;info&gt;</code>.</li> </ol> <p>Note:          Using <code>blockable</code> does not necessarily force you to limit such files on Windows systems. For multi-platform installations there is a compiler warning shown that <code>blockable</code> will be ignored on non-Windows systems.          The native library <code>WinSetupAPI</code> must be explicitly included using this feature.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li>• <b>none</b>              No recognition of blocked target files will be done at all, this is the default behavior of previous IzPack versions.</li> <li>• <b>auto</b>              Automatic recognition of a blocked target file by the operating system, resulting in leaving a pending file operation to be finished after system reboot. Using <code>auto</code> this applies only for files that are really blocked, the other files are copied normally, which can result in mixed, old and new target files at the end of the installation, unless the system won't be really rebooted.</li> <li>• <b>force</b>              Forces target file to be always assumed a blocked, resulting in leaving a pending file operation to be finished after system reboot. Using <code>force</code> this applies for each file, regardless whether it is really blocked during installation. This makes sense if you don't want to mix files old and new files at the end of the installation to not disturbing a running process, but having the complete set of target files installed after system reboot.</li> </ul> <p>Since IzPack 5.0</p>	<p>no</p>	<p>"none"   "auto"   "force"          ("none")</p>
<p>includes</p>	<p>comma- or space-separated list of patterns of files that must be included; all files are included when omitted. This is an alternative for multiple include tags.</p>		
<p>excludes</p>	<p>comma- or space-separated list of patterns of files that must be excluded; no files (except default excludes) are excluded when omitted. This is an alternative for multiple exclude tags.</p>		
<p>condition</p>	<p>an id of a condition which has to be fulfilled to install the files in this fileset</p>		
<p>casesensitive</p>	<p>Whether to treat the file name case-sensitive.</p>	<p>no</p>	<p>"true"   "false"          ("true")</p>

defaultExcludes	<p>Whether to use global default excludes. Implicit default exclude patterns are typically:</p> <pre> * /~{ } * /## * /.# * /%% * /._ ** /CVS */CVS/* ** /.cvsignore ** /SCCS */SCCS/* ** /vssver.scc ** /.svn */.svn/* ** /.DS_Store </pre> <p>Since IzPack 5.0</p>	no	"true"   "false" ("true")
followSymLinks	<p>Whether to follow symbolic links on target systems which support them. Since IzPack 5.0</p>	no	"true"   "false" ("true")

## Nested Elements

The following nested elements can be used in the `<fileset>` tag:

### <OS>

Limit the installation of this file to conditions depending on the target OS, see [OS Restrictions](#).

### <additionaldata>

This tag can also be specified in order to pass additional data related to a file tag for customizing.

Attribute	Description
<b>key</b>	key to identify the data
<b>value</b>	value which can be used by a custom action

`<additionaldata>` is an element which may provide additional information as key-value pairs to certain custom actions. The particular key-value pairs you might use depend on the particular custom action.

Currently, there are two built-in custom actions consuming such data, `ChmodCompilerListener` and `ChmodInstallerListener`, where relevant keys are

- `permission.dir`,
- `permission.file`

with integer values interpreted as permissions like in the Unix `chmod`:

If value begins with "0" -> octal number,

otherwise is is a decimal number representing some permission.

These permissions are applied to the appropriate files either during the compilation of the package or while installing them later, depending on whether the consumer implements a `CompilerListener` or `InstallerListener`.

### <include>

Explicitly include files by pattern, similar like Ant fileset patterns. For more information see the [FileSet](#) core type.

Note: There is currently no file name mapper support in pack filesets.

## **<exclude>**

Explicitly exclude files by pattern, similar like Ant fileset patterns. For more information see the [FileSet](#) core type.

**Note:** There is currently no file name mapper support in pack filesets.