Packs

The <packs> Element

Packs in IzPack are bundles of files grouped under a certain package ID. Each file or set of files in a pack can be given certain attributes, which tell the compiler

- from where to get files into the compiled setup and the installer
- where to put the files
- which files to directly execute (or just receive the executable permission)
- which files to parse and replace variables in
- whether to overwrite existing target files with the same path
- whether to rename existing target files with the same path before they are overwritten

Each pack can be selected or deselected by default or by the user except is marked hidden or mandatory.

Packs Reference

There are the following possible root elements to define packs:

- pack
- refpack
- refpackset

which might embed a couple of possible nested elements describing the pack.

<pack>

Attributes

Attribute	Usage
`id`	unique id for the pack to be used for internationalization via packsLang.xml file
`name`	the user-friendly name that will be displayed during the installation
`required`	specifies whether the pack must be installed (**`yes`**) or is optional (**`no`**)
`os`	**optional**. Lets you make the pack targeted to a specific operating system, see OS Restrictions.
`preselected`	**optional**. Lets you choose whether the pack is selected for installation by default or not. Possible values are yes and no. A pack which is not preselected needs to be explicitly selected by the user during installation to get installed
`loose`	**optional**. indicates that the files are not located in the installer Jar. The possible values are true or false, the default being false. For example, your application could be distributed on a CD so that the users could run it directly from the CD. In order to install the software locally without duplicating the files in a jar, you can set **`loose`** to "true" to make IzPack take the files on the CD rather than from the installer jar. Please make sure that your relative file paths are correct!
`packImgId`	**optional**. reference to a resource that defines the pack's image for the ImgPacksPanel. The resource should be defined in the <resources> element of the installation XML using the same value for the **`id`** attribute of the `<res>` element.

<table>
<tr><td><code>condition</code></td><td>optional. ID of a condition which has to be fulfilled to enable this package to be selected by the user. If the condition evaluates to false the package will be "greyed out" in the PacksPanel user view.

The condition interacts with required in the following way:

<table>
<tr><td></td><td><code>required</code>: false
(default if not defined)</td><td><code>required</code>: true</td></tr>
<tr><td><code>condition</code>: false</td><td>Pack is ...
preselected: no
user interaction: no
disabled: yes
("not enabled")</td><td>Pack is ...
preselected: no
user interaction: no
disabled: yes
("not enabled")</td></tr>
<tr><td><code>condition</code>: true
(default if not defined)</td><td>Pack is ...
preselected: yes
user interaction: yes
disabled: no
("optional")</td><td>Pack is ...
preselected: yes
user interaction: no
disabled: yes
("required")</td></tr>
</table>

Notice: if a package condition evaluates to false but it is marked required the pack remains unselected. Vice versa, a pack's required attribute forces the package to be selected just in case an optional pack condition isn't used or evaluates true.</td></tr>
<tr><td><code>hidden</code></td><td>optional. takes true or false and specifies whether the pack shall be shown in the packs panel. The size of a hidden pack will be used to calculate the required space, but the pack itself won't be shown. A hidden pack can be preselected or selected conditionally. For the latter, you have to specify a <code>condition</code>. The default for this attribute is false.</td></tr>
<tr><td>size</td><td>optional. Specifies the size of the pack, in bytes. If not specified, the size will default to the sum of all file lengths in the pack. Since 5.0</td></tr>
<tr><td><code>installGroups</code></td><td>optional. A comma separated list of groups, see InstallationGroupPanel for using them.</td></tr>
<tr><td><code>uninstall</code></td><td>optional. If set "<code>true</code>", the according pack gets into the uninstaller's list of files to be uninstalled. This attribute has effect just in case of <code><uninstaller write="true"/></code>. Default: <code>true</code></td></tr>
</table>

Nested Elements

<singlefile> - Adding/renaming a Single Pack File

See Adding or renaming a single file for more details.

<file> - Adding a Set Of Files

See Adding or unpacking a single file to a target directory for more details.

<fileset> - Adding a Set Of Files With Filtering

See Adding a set of files for more details.

<description> - Adding a Description

The contents of the <description> tag describe the pack contents. This description is displayed if the user highlights the pack during installation.

<depends> - Defining Dependencies

This can be used to make this pack selectable only to be installed only if some other is selected to be installed. The pack can depend on more than one by specifying more than one <depends> elements. Circular dependencies are not supported and the compiler reports an error if one occurs.

This tag takes the following attribute:

packname	The name of the pack that this one depends on

<onSelect> - Select/Deselect a pack when this pack is selected
<onDeselect> - Select/Deselect a pack when this pack is deselected

Gives you the ability to select or deselect other packs upon selection or deselection of this pack.
Allows for flexible pack management.
These tags take may take the following attributes:

name	**Required.** A comma separated list of pack names. Pack names per-appended with an exclamation mark '!', will be deselected when this pack is selected. Pack names that are not per-appended with an exclamation mark will be selected when this pack is selected.
condition	**Optional.** When the specified condition is true the onSelect action will work as normal. When the specified condition is false the onSelect action will have no effect.

`<parsable>` - mark text files for variable replacement

See Marking files for variable replacement for more details.

`<executable>` - mark files for execution

See Marking files for execution for more details.

`<updatecheck>` - Cleaning up on updates

See Cleaning up unwanted files after updates for details.

`<refpack>`

The `<refpack>` takes only one attribute `file`, which contains the relative path (from the installation compiler) to an externally defined packs-definition. This external packs definition is a regular IzPack installation XML. However the only elements that are used from that XML file are the <packs> and the <resources> elements.

This enables a model in which a single developer is responsible for maintaining the packs and resources (e.g. separate packsLang.xml_xyz files providing internationalization; see Internationalization of the PacksPanel) related to the development-package assigned to him. The main install XML references these xml-files to avoid synchronization efforts between the central installation XML and the developer-maintained installer XMLs.

Attributes

Attribute	Description
`file`	Relative path at compile-time to an externally defined packs-definition

`<refpackset>`

The `<refpackset>` tag can be used in situations were there is no predefined set of `<refpack>` files, but a given directory should be scanned for `<refpack>` files to be included instead. This element takes the following parameters:

Attribute	Description
`dir`	Relative base directory at compile-time for the refpackset
`includes`	Pattern of files in `<refpack>` format that will be included

Example:

```
<refpackset dir="" includes="**/refpack.xml" />
```

Internationalization of the PacksPanel

In order to provide internationalization for the PacksPanel, so that your users can be presented with a different name and description for each language you support, you have to create a file named packsLang.xml_xyz where xyz is the ISO3 code of the language in lowercase. Please be

aware that case is significant. This file has to be inserted in the resources section of **install.xml** with the **id** and **src** attributes set to the name of the file. The format of these files is identical to the distribution langpack files located at **$IZPACK_HOME/bin/langpacks/installer**. For the name of the panel you just use the pack **id** as the part of text **id**. For the description you use the pack id suffixed with .description.

Example:

```
<resources>
    <res id="packsLang.xml_eng" src="i18n/myPacksLang.xml_eng"/>
</resources>
```

The packsLang.xml_eng file:

```
<langpack>
    <str id="myApplication" txt="Main Application"/>
    <str id="myApplication.description" txt="A description of my main application"/>
</langpack>
```

Example

```
<pack name="Core files" required="yes" id="pack.core" condition="Install">
  <description>Core files</description>
  <fileset dir="@{staging.dir}" override="true">
    <exclude name="*.zip" />
    <exclude name="conf/*.properties" />
    <exclude name="conf/*.xml" />
  </fileset>
  <fileset dir="@{staging.dir}/config_files" targetdir="${INSTALL_PATH}/conf"
override="true" overrideRenameTo="*.configbak">
    <include name="*.properties" />
    <include name="*.xml" />
    <exclude name="special.xml" />
  </fileset>
  <parsable encoding="UTF-8">
    <fileset targetdir="${INSTALL_PATH}/conf">
      <include name="wrapper.conf" />
    </fileset>
  </parsable>
  <parsable>
    <fileset>
      <include name="**/*.bat" />
      <include name="**/*.cmd" />
    </fileset>
  </parsable>
  <parsable type="shell">
    <fileset>
      <include name="**/*.sh" />
    </fileset>
  </parsable>
  <executable>
    <fileset>
      <include name="**/*.sh" />
    </fileset>
  </executable>
</pack>
```

The dir attribute should no longer been parsed in <fileset> nested to <executable>, <parsable> at all.