

AntActionInstallerListener and AntActionUninstallerListener

IzPack provides support to execute Ant tasks during installation and uninstallation.

For example:

```
<resources>
    ...
    <res id="AntActionsSpec.xml"
src="myInstallSpecs/MyAntActionsSpec.xml" />
    ...
</resources>

<jar src="jar/ant.jar" stage="both"/>
<jar src="jar/customtasks.jar"
stage="both"/>

<listeners>
    <listener
classname="AntActionInstallerListener"
stage="install" />
    <listener
classname="AntActionUninstallerListener"
stage="uninstall" />
</listeners>
```

In the above:

- the **AntActionsSpec.xml** resource determines the Ant tasks to be executed. The base path of `src` is the installation project path.
- the **<jar/>** element specifies jar dependencies required to execute the Ant tasks. At a minimum, this must include the Ant jar.
- the **<listeners>** element is used to hook execution of the Ant tasks into the installer and uninstaller

Nor that an "extended" ant use needs more than one jar, for example often `xercesImpl.jar`. If an obscure "class not found" exception is raised during testing, check first for missing jar files.

The `stage` attribute of the `jar` element determines when a jar is required. If an ant uninstaller custom action is used, the uninstaller also needs the jar files. If `stage` is "both" or "uninstall", the contents of the referenced jar file will be packed into the uninstaller.

Note that not the jar file itself, but the contents of it are required. This implies, that the paths of the contained files are unique and the information in `meta-inf/Manifest.mf` will be lost.

The AntActionSpec XML Structure

An ant action is defined in the resource with the id "AntActionsSpec.xml". In IzPack 5.0 (as of 5.0.0-beta-11), an XML-Schema is provided, which may be declared as follows:

```
<izpack:antactions version="5.0"
xmlns:izpack="http://izpack.org/schema/antactions"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://izpack.org/schema/antactions
http://izpack.org/schema/5.0/izpack-antactions-5.0.xsd">
<!-- .... -->
</izpack:antactions>
```

Prior to processing the packs a substitution is performed using all defined `IzPack` variables on the spec file. This is a common way of loading spec files into custom actions.

For more information see method `com.izforge.izpack.util.SpecHelper.readSpec`. If you want to substitute a value in the file, simply add a variable via `installData.setVariable` in a custom panel before `InstallPanel1`. The given variable name (id) should be written into the xml file in the common variable notation.

The following outlines the sections available in the `AntActionsSpec.xml`:

antactions

The top level XML section is called `<antactions>`. The `antactions` element may contain zero or more `pack` elements described below:

pack

The `pack` element is used to tie the action(s) to be performed to the packs selected in the installation. The `pack` element may have the following attribute:

Name	Description
name	The name of the pack that this pack element maps to. This should correspond to the name of a pack defined in <code>install.xml</code>

The `pack` element must contain 1 or more `antcall` elements.

antcall

The `antcall` element has the following attributes:

Name	Required	Description	Allowed Values
order	yes	Determine at what point of installation the <code>antcalls</code> defined by element <code>target</code> should be performed. Be aware that with <code>beforepack(s)</code> there are no installed files and also no installed build file. With this order only preexistent build files are useable.	<code>beforepack</code> , <code>afterpack</code> , <code>beforepacks</code> or <code>afterpacks</code> .
uninstall_order	no	Determine at what point of uninstallation the <code>antcalls</code> defined by element <code>uninstall_target</code> should be performed. As opposed to the behaviour of <code>order</code> the referenced files are also accessible in the order <code>afterdeletion</code> . The uninstaller action copies the files into <code>tempfiles</code> before deletion which are marked as <code>deleteOnExit</code> .	<code>beforedeletion</code> , <code>afterdeletion</code>
quiet	no	To quit or not.	yes or no, default: no

verbose	no	To output verbose information or not.	yes or no, default: no
logfile	no	Path to the file where logging should be performed. Note: The logfile should be not marked for uninstallation otherwise it will be deleted too.	Any valid file path (TODO: should verify format)
buildfile	yes, if buildresource not specified	Path to the file which contains the antcall. This is the file you normally use as <code>-buildfile</code> during an ant call via the command line. Variables are not substituted in this file. Variables should be set using ANT properties. Note: Never write an <code>IzPack</code> variable in an ant buildfile.	Any valid file path (TODO: should verify format)
buildresource	yes, if buildfile not specified	The value is the id of the resource which contains the antcall. This resource will be extracted out into a temporary file and the path to this file will be passed as if <code>-buildfile</code> were specified during the ant call via the command line. The temporary file is removed after the ant call. In this file variables are not substituted. For substitution there are properties in ant which can be used. Note: Never write an <code>IzPack</code> variable in an ant buildfile.	A resource id.
messageid	no	If it is defined, the message will be displayed in the InstallPanel whilst performing the ant call.	A string ID which refers to <code>bin/langpacks/installer/<lang>.xml</code> .
condition	no	Execute this action only if the condition is fulfilled.	A valid condition string.

In addition to the possible attributes there are some elements may be defined within an antcall element. All elements can be defined zero or more times in an `<antcall>`. Although all elements are optional, no ANT tasks will be performed unless a `<target>` element is specified. Do not confuse the following: "required"s are related to the attributes of the elements, not to the elements themselves.

property

Defines a property to be used with all targets and `uninstall_targets` which are defined for this antcall. The following attributes may be defined:

Name	Required	Description	Allowed Values
name	yes	The name of the property to set.	Any string value
value	yes	The value to set for the property.	Any string value

propertyfile

Defines properties to be used, read from a property file which are defined for this antcall. The following attributes may be defined:

Name	Required	Description	Allowed Values
path	yes	The path to the properties file.	A valid file path.

One way to fill specific data into it is to create a new file in a custom panel and fill it with values given by input fields. The file path can be set at installation time, if there is a variable in `AntActionSpec.xml` and an `IzPack` variable was defined before `InstallPanel`. That file can be only created with `deleteOnExit`, if no `<uninstall_target>` was defined in this `<antcall>`. This implies, that other `<antcall>`'s can have a `{{<uninstall_target>. <target>: target to call at installation h3. Targets to perform with this antcall at installation time. The targets should be defined in the given buildfile or else an ant exception will be raised. This is that what you use, if you don't want to perform the default target. e.g. cleaning the IzPack project with ant clean.`

target

The target to execute in the ANT build file during action specified in the order attribute of the antcall (install time). **Note:** Multiple targets to execute for the action may be specified by using multiple target elements. The following attribute must be defined:

Name	Required	Description	Allowed Values
name	yes	Name of the target to execute in the ANT buildfile	Any valid ANT target defined in the build file.

uninstall_target

Targets to perform with this antcall at uninstallation time. **Note:** Multiple targets to execute for the action may be specified by using multiple target elements.

Name	Required	Description	Allowed Values
name	yes	Name of the target to execute in the ANT buildfile	Any valid ANT target defined in the build file.