

Best practices

Best practices

This is a set of best practices for IzPack developers and contributors.

It tries to be normative without putting too much bureaucracy in the processes.

General recommendations

1. Read the [Codehaus Manifesto](#)
2. Be friendly. The IzPack community has always been like that and we don't want that to change!
3. A new feature must always be documented, else there is no point in adding it!
4. Any fix or enhancement must be attached to an issue from our JIRA instance at <http://jira.codehaus.org/browse/IZPACK> (this allows proper planning of the releases as well as generating complete and meaningful release notes).

Please keep in mind that most IzPack developers are volunteers that are not paid for developing it. We do it for fun on our free time, so please don't expect the same level of support as if we were a company (but we're doing our best to be as close to that as possible 😊).

Recommendations for developers

General recommendations

1. Always **create** a [JIRA issue](#).
2. Use **meaningful comments** when doing a commit/push/pull request.
3. Always **mention the JIRA issue** that is related to your commit (e.g., "*Fix for JIRA-123*").
4. When doing a commit based on a patch from a contributor, make sure to mention the contributor name in the comment (e.g. "*Fix for JIRA-123. (Contributor name)*").
5. Never commit incomplete changes (GIT is not a backup system).
6. Working on experimental changes is encouraged, but in this case use a branch.
7. Don't maintain *Versions.txt* anymore, we will extract release notes from JIRA starting from IzPack 4.0 and beyond.
8. **Subscribe** to the [SCM mailing list](#) to get **notified** of SVN commits and JIRA activity.
9. Subscribe to the [developer mailing list](#) to follow IzPack development conversations.
10. Do not hesitate to **pick new JIRA issues**: both the other developers and the issue submitter will thank you for the work!
11. When you start working on an issue, don't forget to mark it as *in progress* so that we know that you're solving it!
 - For developing workflow will be: *Open -> In progress -> Resolution -> Resolved -> Closed*
 - For language file update: *Open -> Resolved -> Closed*
12. For more information about an issue please visit: [What is an Issue?](#)

Code reviews

Before merging a patch submitted by an external contributor, developers are expected to make a simple code review.

1. Look at the code carefully to ensure that it is of good quality
2. Obviously... test it 😊
3. Attach your review to the related JIRA issue
4. Send your review to the dev mailing-list if needed.

In case no other developer makes a comment after a few days, you can act as you decided: accept or reject it!

Recommendations for contributors and issues reporters

1. Always **create** a [JIRA issue](#) in case of bug, improvement or new feature. You can follow progress of your issue there.
2. If you have a patch, then **attach it** to the JIRA issue.
3. Do not hesitate to attach **screenshots** and **stack traces**.
4. Make sure you **send a real diff-generated patch**, not whole files or ZIPs with the modified files... (Several programs will generate this, for example: IntelliJ, Eclipse, Netbeans, TortoiseSVN,...)
5. Should you propose a new feature, make sure to **attach documentation**. Ideally you will have a patch for the documentation, but plain text will also do.
6. **Subscribe** to the [developer mailing list](#) if you want to follow IzPack development conversations.
7. **Respect code format styles!** Otherwise we have a hard time trying to merge your code.

Code formatting

The **src** folder in the IzPack source code contains code style settings for Eclipse and IntelliJ IDEA.

Everyone has its own preferences, so to make everyone comfortable, here are a few things to respect.

The preferred style is ANSI-style, i.e., we prefer:

```
if (i == 0)
{
    doThis();
}
else
{
    doThat();
    andThat();
}
```

over:

```
if (i == 0) {
    doThis();
} else {
    doThat();
    andThat();
}
```

of even this:

```
if (i == 0) doThis();
else {
    doThat();
    andThat();
}
```

You may prefer putting braces on end of lines rather than on new lines, but in this case please don't be feel offended when we run code formatting from times to times 😊

Taking the project from version repository you will have Eclipse settings for the IDE and code formatting.

You should avoid long width lines. 80 characters is probably short nowadays. 100-120 characters looks like to be a good limit.

After IzPack 3.11.0 we will use **Java 5** as the language source and target version.