Confirm that the Contribution:
1. Contains No Cryptography
2. Is Developed from Scratch (without incorporating content from elsewhere) – Contact the EMO if you aren’t sure.
3. Is 100% Project License Code

Yes to All

No to Any

Did it come from Bugzilla/Gerrit?

yes

Contributor must create a Bugzilla/Gerrit entry and attach code.

Figure 10

Figure 11

Figure 11b

Is it less than 1,000 lines of code and configuration files and other source?

no

Go To: "Everything Else" (Figure 6)

Figure 13

yes

Create about.html (if required) & commit to the repository. Don’t forget to record the contribution in your Project IP Log!

Figure 8

The activity described in Figures 1 through 5 represent the majority of development activity at Eclipse.

Definitions, Dependencies, Moving Code to Eclipse and Distributing other Eclipse Projects, Plug-Ins or Bundles – See Page Three
Do NOT submit to the repository yet. Fill out a Contribution Questionnaire on Committer Tools. This creates an IP2Ela Bug that will enable you to monitor the Contribution and discuss it with the EMO and Community.

Figure 20

Attach related source code to your IPBug

Figure 21

A Member of your PMC must vote +1 on the IPBug Related to the Contribution

Figure 22

Foundation confirms that it has received all necessary information and code & if the package uses a license not previously used at Eclipse, informs the Board.

Figure 23

Is your contribution eligible for parallel IP? A contribution is eligible for parallel IP if:
(a) your project is incubating and in compliance with the Incubating project guidelines.
(b) In the case of mature projects, the contribution represents a minor change to a previously approved package.

Figure 24

Is the License approved for use within Eclipse (based on the information provided by the Committer) and there are no other difficulties apparent from the code based on a high level review. See comment on Figure 19.

Figure 25

IPBug is marked with the keyword “check”

Figure 26

Create about.html if required & commit to the repository.

Figure 27

Create about.html if required & commit to the repository.

Figure 28

EMO completes Due Diligence

Figure 29

Approved

If applicable, Foundation advises Committer and Board of Approval (ref Figure 23)

Figure 30

Foundation advises Committer of Non-Approval

Figure 31

If Non-Approval relates to incubating code in the repository code must be removed.

Figure 32

Work with PMC to Examine Alternatives

Figure 33

Every Everything Else

This includes:
1) Initial contributions to a project
2) Code containing cryptography
3) Non-Eclipse Content (shows page 3)
4) Code developed by Non-Committee, where workflow from Figure 2 or Figure 3 leads to Figure 12
5) Re-use of Eclipse components

Figure 6

Is the license approved for use within Eclipse? If uncertain, contact license@eclipse.org.

Figure 17

Is the license EPL?

Figure 18

For an unmodified binary distribution, the following licenses are considered approved as of the date of this poster’s release:

- Apache Software License 1.1
- Apache Software License 2.0
- W3C Software License
- Common Public License Version 1.0
- IBM Public License 1.0
- Mozilla Public License Version 1.1
- Common Development and Distribution License (CDDL) Version 1.0
- GNU Free Documentation License Version 1.3 (BSD)
- MIT

Licenses that are not approved as of the date of this poster’s release include:

- GNU GPL 2.0
- GNU LGPL
- Sun Binary Code License Agreement

Want to use a source distribution instead of a binary distribution? Want to distribute modified code? Want to use a license not contained in this list? Not certain of the answers? Contact license@eclipse.org.
**Definitions:**

“Project License” – your default Project license will be the EPL. Any other licensing strategy requires a unanimous vote of the Eclipse Board of Directors.

“Non-Eclipse Content” – any code maintained on servers other than those of the Eclipse Foundation.

“Under Supervision of the PMC” – refers to general supervision; sufficient to ensure the code being submitted is in line with the goals of the project from a technical standpoint. This level of supervision may vary by project. Determination is to be made by the relevant PMC of the project.

“Submitting Committer” – An Eclipse committer on the project at the time of development. Code developed prior to becoming an Eclipse committer requires due diligence review.

**Third Party Dependencies:**


And our Test and Build Resources: [http://wiki.eclipse.org/Development_Resources/IP/Test_and_Build_Dependencies](http://wiki.eclipse.org/Development_Resources/IP/Test_and_Build_Dependencies)

**Moving Code to Eclipse:**

Interested in moving code from somewhere else to Eclipse and maintaining it at Eclipse? Contact emo@eclipse.org

**Distributing Eclipse Projects, Plug-Ins & Bundles – Guidelines:**

Release Candidate Distributions **must not contain** Non-Release Candidate (e.g. not “RC1” or final release “1.0”) distributions from other Eclipse Projects as such releases may contain non-reviewed and approved content.

Release Candidate Distributions **may pre-req** Non-Release Candidate (e.g. not “RC1” or final release “1.0”) distributions from other Eclipse Projects provided the downstream consumer is made aware that the content that is being pre-req’d may contain non-reviewed and approved content.

Non-Release Candidate Distributions **may contain** Non-Release Candidate (e.g. not “RC1” or final release “1.0”) distributions from other Eclipse Projects.

Non-Release Candidate Distributions **may pre-req** Non-Release Candidate (e.g. not “RC1” or final release “1.0”) distributions from other Eclipse Projects.

**Simultaneous Release:** All Projects participating in the Release Candidate for the simultaneous release must be Release Candidate themselves. The above guidelines apply to any Project wishing to pre-req or incorporate other Eclipse Projects.