Release Nomenclature

Clarifying release numbering and error correction processes

Version: 1.1.0

Connect. Collaborate. Accelerate.

OPEN Compute Project®

Goal

The OCP Foundation’s intent is to provide guidance, socialize a proposed process and introduce the operational steps, initially, in terms of a best practice to make version control more uniform for The OCP Community.

Versions

In managed releases, each result of development that is to be distributed is carefully controlled. A unique name is assigned to each result and history tracked. The name includes a **version**, which is the main component of the identifier of the release assets.

Key Points:

- Indicates either something all new, or changes that result in new scope, or something updated that is no longer ‘compatible’ or with greatly different development/production/implementation risk.
- Rule of thumb >=25% total change to original
- Assets are made available to the community following the OCP Release process.

Revisions

Revision is a sub-component of the identifier of the release controlled assets. Revision should indicate modifications, optimizations or incremental changes that are more incremental than revolutionary.

Key Points:

- Minimal scope change (see version key points)
- Rule of thumb <=25% overall difference to original
- May or may not be from original contributor(s)
- Follows OCP Release process, revisions require Steering Committee review and approval

Errata

It is recommended that each project/sub-project/workstream systematically collect and document errors. These errors and corrections collectively are termed **Errata**.

Key Points:

- Each accepted error should be serialized (numbered or unique identifier) with exact original page number and line number identified to avoid any ambiguity.
- Errata (corrections) is into a separate document (see Errata template)
- Note, Errata in a separate document is labeled with the released identifier plus errata identifier.

Labeling: Semantic Versioning

Semantic labeling assigns a compound identifier designed to indicate scope and history at a glance. The format is MAJOR.MINOR.PATCH

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH/ERRATA format.

Pre-release, use Release Candidate or RC and a version. There is no preference as to using the pre-release tag before or after the label as it will disappear for a formal release.

- Example: Release Candidate 2 of Version 1 Revision 1 would be: RC2-1.1.0

Released versions and revisions begin at 1.

Errata tag only really applies to subsequent releases. Thus if errata is identified after release of Version 1.Revision 0, it is expected that Revision 1 incorporates errata thus errata is reset to 0.

When labeling a release, use MAJOR.MINOR.PATCH format and increment the:

1. MAJOR **{version}** when you make incompatible changes
2. MINOR **{revision}** when you add functionality in a backward compatible manner
3. PATCH **{errata}** when you make backward compatible bug fixes

Reference: https://semver.org

OCP Labeling Example with Semantic Versioning

Example of **current** versioning: *Open Accelerator Infrastructure (OAI) OCP Accelerator Module OAM Base Specification* **r2.0 v1.0**

Example of using **new** versioning: *Open Accelerator Infrastructure (OAI) OCP Accelerator Module OAM Base Specification* **1.2.0**

Contribution templates have Semantic versioning.

Release Process

Release of a version or revision of a contribution occurs when the Steering Committee approves a proposed specification. Pre-release is internal releases or release candidates and have not presented to the Steering Committee for approval. All packages of any type are release controlled with identifiers in the semantic labeling format.

It's best practice to label **pre-releases** with a label, such as **RC** for Release Candidate. This can be before or after the release identifier. {version.revision.errata}

- Examples:
  - *{LABEL}-VERSION.REVISION.ERRATA*: **RC**3-1.2.0
  - *VERSION.REVISION.ERRATA-{LABEL}*: 1.2.0-**RC**3

Errata collected after release is reviewed by the Project Leaders and referred to their SC representative. Once the SC representative approves the errata is published.

Close

Following the crawl, walk, run concept, a best practice for version/revision numbering and errata is defined for the OCP contribution process.

Please connect with The OCP Foundation team with questions or before using a different approach than what is outlined here.

Thank You

OPEN
Compute
Project®