

01. Requirements for Participation in Release

- 1. Planning (including Service Release participation)
- 2. Meeting Deadlines
- 3. Leadership & Communication
- 4. Quality
- 5. Documentation
- 6. Distribution

In order to participate in the simultaneous release, a project must exist (see the [list of existing projects](#) or create a [new project proposal](#)) and do the following things.

1. Planning (including Service Release participation)

- Projects must declare their intent to participate in the Simultaneous Release by M0. This is accomplished by:
 1. providing the [Increment blueprint](#)
 2. adding the project to the [Release Scope Table](#)
 3. sending the first milestone readout mail to [TSC email list](#).
- Participating, approved by TSC projects, must publish a candidate Release Plan shortly after M0, and declare their final Release Plan by M1 - the [Increment blueprint](#) and the [Release Scope Table](#) must be updated accordingly
 - Projects are required to negotiate cross-project dependencies for any new or modified APIs.
 - Projects are encouraged to think about any cross-project incompatibilities and how to resolve them, if possible, as part of their release plans.
 - Participating project Release Plans must contain Milestones that minimally line up with the Simultaneous Release Plan Milestones
 - Release plans should contain a complete list of the exposed APIs including the defined properties, e.g., the name of the interface, a short name, and the list of supporting bundles.
 - Per-project release plans now include sections for cross-project negotiation of provided APIs and for noting cross-project incompatibilities.
- All projects participating in the release are also required to participate in the stability releases described in the schedule after the formal release.

2. Meeting Deadlines

- To indicate this, the project lead/contact is expected to provide and send a milestone readout to the release mailing list by 23:59:59 UTC on the date listed for the the appropriate offset at each milestone.
- Most information will be communicated by filling out appropriate information in the [Tracking Spreadsheet](#), but a mail should still be sent indicating that the information has been filled in. Any other information or questions can be included in that mail.
- For **offset two** project this is mainly intended to be reflective and to help inform the release process.
- For **offset zero** and **offset one** projects, this should be completed within 24 hours of missing the deadline and must be presented to the TSC at the first TSC meeting after the deadline.
- *NOTE: For deliverables defined only in the project's release plan—and not as a requirement in this document—the release management staff and /or TSC will verify that the status of the deliverables has been reported. release management staff and/or the TSC may also, but are not required to, verify the delivered functionality.*
 - All projects are expected to meet the deadlines laid out in the schedule below.
 - If a project cannot make a deadline, the project lead/contact must write a summary of what was missed, why, the course correction that will be taken, and its impact on other projects.
 - All Milestone deliverables will be verified by the release management staff and/or the TSC.

3. Leadership & Communication

- Project Team Leader must be elected and confirmed as described in the TSC charter. [Casey Cain](#) will help projects with this process and it must be completed by M0.
- The results of the election, and other changes to the project lead during this release, should be reported by
 1. Updating the project facts for the project on its [main wiki page](#)
 2. Updating the [Release Scope Table](#) of this release
 3. Sending an e-mail to the [TSC email list](#)
- The TSC collects response time metrics for projects both to inform our planning and to measure project maturity going forward.
- The Project Leader is expected to be responsible for the project meeting deadlines, interacting with other projects, and interacting with the TSC
- The Project Leader will be subscribed to the *release mailing list* and must respond to requests sent to the project in a timely fashion—defined as **two business days**.
- The project Leader is expected to, at a minimum, read the *release mailing list*, read the [TSC meeting minutes](#). The Project Leader is strongly encouraged to attend these meetings if at all possible and some representative from the project is expected to attend each IRC meeting if at all possible.
- In addition to the Project Leader, each project must designate a Test Contact and Documentation Contact to handle test-related communication.

- All release-critical correspondence that requires a response will have a subject line containing "PLEASE RESPOND BY <DATE> <UTC-TIME>" or "URGENT RESPONSE REQUIRED/NEEDED"
- If Project Leaders are not able to do so, they should (i) have somebody else stand in and do this on their behalf, (ii) send a mail to the release mailing list indicating this and the time period, and (iii) note the same information in the participating projects section of the [Release Scope Table](#).
- Please limit traffic to correspondence directly relating to the release

4. Quality

- Projects, especially ones that form key infrastructure for other projects, are strongly encouraged to set goals for code coverage and reported bugs. Doing so will be seen favorably when evaluating projects for advancement in the Project Lifecycle.
- Stable Features must have an appropriate unit and/or integration test coverage of at least 75% prior to M4.
 - No later than M1, each project must verify that the project builds and passes tests for each new patch pushed to GitHub.
 - No later than M1 as part of the participating projects must push their binary artifacts to the repository
 - No later than M1, each project must have a Jenkins Job which rebuilds and retests to an appropriate level when a project it depends on publishes new artifacts, i.e., a Jenkins "-integration" job.
 - No later than M1, each project primarily written in Java must be reporting unit and/or integration test coverage via sonar. (See instructions on reporting test code coverage)
- Testing
 - Stable Features must have established integration and system tests as required for Mature project Stable Features.
 - Note: The system test requirements can be waived by the TSC for a given feature if for example the top-level feature is tested through another top-level feature.
 - Note: Projects running system test outside TF (external Lab) are required to report system test results in a timely fashion after release creations, e.g., weekly, RC, and formal releases.
 - In addition to setting up appropriate -verify, -merge, and -integration jobs by M1, projects are expected to provide adequate unit, integration and system tests.
 - The coverage provided by unit tests and integration tests should be reported by M1. (See instructions on reporting test code coverage)
 - Participating projects must describe a basic system test per top-level feature and a comprehensive system test including functionality, cluster, scalability, performance, longevity/stability per stable feature prior on M2.
 - Participating projects must run at least one basic automated system test for each top-level feature and several automated system tests including functionality, cluster, scalability, performance, longevity/stability for each stable feature by M4.
 - System tests are expected to reliably pass. If a system test turns out to be unstable and intermittently fails, it must be fixed or disabled. If intermittent system tests are seen as having value to the project, they can be written and run on-demand by the project, but won't be run as part of the automated CSIT suite.
 - Further details and requirements can be found in the schedule and project integration and test requirements below.
- Code Hygiene
 - No uses of System.out.println in non-test code.
 - No dependencies on 3rd party (non-TF) snapshot versions
 - Willing to use agreed-upon versions for dependencies (both 3rd-party and TF), i.e., to eliminate version skew
 - Willing to find source code for 3rd-party dependencies and/or move to versions or alternatives for which source code can be found.

5. Documentation

- Each participating project is expected to identify the kinds of documentation that would be useful (e.g., installation guide, user guide, developer guide) and provide them as part of the release.
- More details on the expectations can be found in the schedule and project documentation requirements below.

6. Distribution

- All projects must support a distribution model including defining features and adding them to integration repository no later than M2.
- No later than M2, each project must have a "distribution-check" Jenkins Job to verify changes in the code do not break integration distribution.