

Zowe Charter

v1.0 – 8.08.18

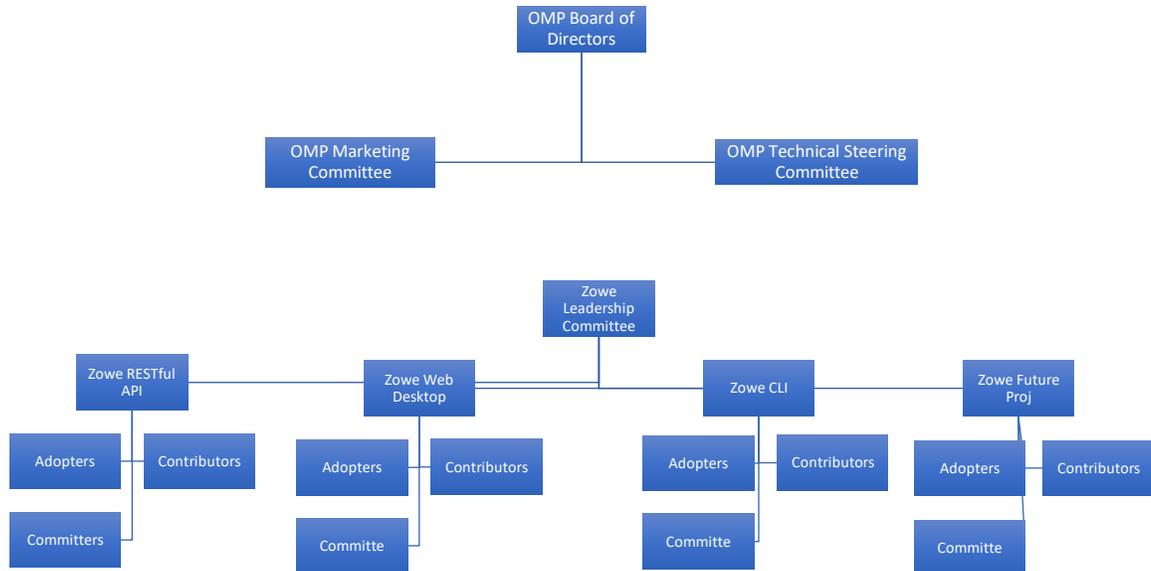
Meritocracy

A foundational principle of this new project is meritocracy. The more that somebody contributes, the more responsibility they will earn. A pattern of quality contribution to a project may lead to an invitation to join the project as a committer. Leadership roles in the Project are also merit-based and earned by peer acclaim. Merit must be demonstrated in publicly-accessible forums. Committers and project leads are added to a project via an election.

Those that are interested in participating in the community and delivering support in the form of code, documentation, test and other tangible contributions are welcome to be a part of the community.

Project Management - Governance and Sub-projects

Zowe has members, a leadership committee and other constituents.



Zowe Leadership Committee (ZLC):

Sub-projects (i.e. Zowe RESTful API) have a lead and the leads of all sub-projects make up the Zowe Leadership Committee (ZLC). In addition, a few members will be appointed to the initial Zowe ZLC. The Zowe ZLC will:

- Receive proposals for sub-projects leveraging Zowe technology and will decide, via a vote, on whether the sub-project is accepted.
- Be responsible for working with the OMP Marketing Committee to ensure messaging and content is consistent with the vision and mission of Zowe.
- Create and maintain development iteration criteria and release train to drive framework development and deployment
- When required, vote on revoking commit rights

Zowe ZLC decisions (including elections of new members) shall be based on a majority vote, provided that a quorum, at least sixty percent (60%), of the Zowe ZLC is either present or participating electronically or by written action in order to conduct a valid vote.

The initial Zowe ZLC will consist of equal representation of the founding parties: IBM, Rocket and CA. New Zowe ZLC members will be accepted onto the committee via a written proposal and agreed to by majority vote. A yearly election process will be developed to ensure leadership is based on meritocracy and will begin one (1) year after the project is initially launched so that a true community election process can take place.

Any Zowe ZLC member(s) may resign at any given time by giving written notice to the Committee. The resignation should take effect, and specifically noted with a date listed in the written notice. Once a notice is given, it is irrevocable.

Any Zowe ZLC member may be removed with cause if they are in breach of the [Code of Conduct](#).

Together the Zowe ZLC provides oversight and overall leadership for the projects that fall under the Zowe top-level project. The Zowe ZLC as a whole, and the Zowe ZLC leads in particular, are ultimately responsible for ensuring, in accordance with the principles of the project described below, that the Development Process is understood and followed by their projects. The Zowe ZLC is additionally responsible for maintaining the top-level project's charter/scope.

Zowe Project Lead(s):

Each sub-project is managed by a project lead. Collectively, Zowe Project leads are responsible for ensuring that their project's committers are following the Development Process, and that the project is engaging in the right sorts of activities to develop vibrant communities of users, adopters, and contributors. Each project lead is equal in responsibility. Subsequently, additional project leads must be elected by the project's committers.

Contributors and Committers:

Each project has a development team, led by the project lead(s). The development team is composed of committers and contributors. Contributors are individuals who contribute code, fixes, tests, documentation, or other work that is part of the project. Committers have write access to the project's resources (source code repository, bug tracking system, website, build server, downloads, etc.) and are expected to influence the project's development.

Contributors who have the trust of the project's committers can, through election, be promoted to committer for that project. The breadth of a committer's influence corresponds to the breadth of their contribution. A development team's contributors and committers may (and should) come from a diverse set of organizations. A committer gains voting rights allowing them to affect the future of the project. Becoming a committer is a privilege that is earned by contributing and showing discipline and good judgment. It is a responsibility that should be neither given nor taken lightly, nor is it a right based on employment by a Zowe member company or any company employing existing committers. The initial set of committers for the project(s) will be voted in, by majority vote, by the founding parties (IBM, Rocket and CA).

The election process begins with an existing committer on the same project nominating the contributor. The project's committers will vote for a period of no less than one week of standard business days. If there are at least three (3) positive votes and no negative votes within the voting period, the contributor is recommended to the project's ZLC for commit privileges. If there are three (3) or fewer committers on the project, a unanimous positive vote of all committers is substituted. If the ZLC approves, and the contributor signs the appropriate committer legal agreements managed by the OMP (wherein, at the very least, the developer agrees to abide by the Intellectual Property Policy), the contributor becomes a committer and is given write access to the source code for that project.

At times, committers may become inactive for a variety of reasons. The decision making process of the project relies on active committers who respond to discussions and vote in a constructive and timely manner. The project leads are responsible for ensuring the smooth operation of the project. A committer who is disruptive, does not participate actively, or has been inactive for an extended period may have his or her commit status revoked by the project leads. Unless otherwise specified, "an extended period" is defined as "no activity for more than six months".

Active participation in the user communication channels and the appropriate developer mailing lists is a responsibility of all committers, and is critical to the success of the project. Committers are required to monitor and contribute to the user communication channels.

Committers are required to monitor the written communication associated with the project. This is a condition of being granted commit rights to the project. It is mandatory because committers must participate in votes (which in some cases require a certain minimum number of votes) and must respond to the mailing list in a timely fashion in order to facilitate the smooth operation of the project. When a committer is granted commit rights they will be added to the appropriate mailing lists. A committer must not be unsubscribed from a developer mailing list unless their associated commit privileges are also revoked.

Committers are required to track, participate in, and vote on, relevant discussions in their associated projects. There are three voting responses: +1 (yes), -1 (no, or veto), and 0 (abstain).

Committers are responsible for proactively reporting problems in the bug tracking system, and annotating problem reports with status information, explanations, clarifications, or requests for more information from the submitter. Committers are responsible for updating problem reports when they have done work related to the problem.

Committer, Project Lead or Committee member are roles; an individual may take on more than one of these roles simultaneously.

Adopters:

An adopter uses the frameworks and technology provided by Zowe to extend and craft their own offerings. These offerings can be internal or external. If external, they can be made available either commercially or free of charge.

An active and engaged adopter community is the only way to prove that a Zowe project is providing extensible frameworks and applications. Reuse of the frameworks within the companies that are contributing to the project is necessary, but not sufficient to demonstrate an adopter community. Again, creating, encouraging, and nurturing an adopter community outside of the project's developers takes time, energy, and creativity by the project leadership, but is essential to the project's long-term open source success.

The Zowe community considers the absence of any one or more of these communities as proof that the project is not sufficiently open, transparent, and inviting, and/or that it has emphasized applications at the expense of extensible frameworks, unless the charter specifies an application.

Process

The development process must be clear and concise to enable project teams to navigate any complexities of development. The frameworks, projects, and community are an evolving landscape and rules or processes that set a static snapshot of any of these are detrimental to the growth and success of Zowe. Notwithstanding this, projects must use resources and services approved by the ZLC. This ensures consistency of source code management tools, build tools, code repositories, testing tools, and enables members to switch between projects more easily, as well as high level management to be enabled through consistent individual project tools.

Project Assets

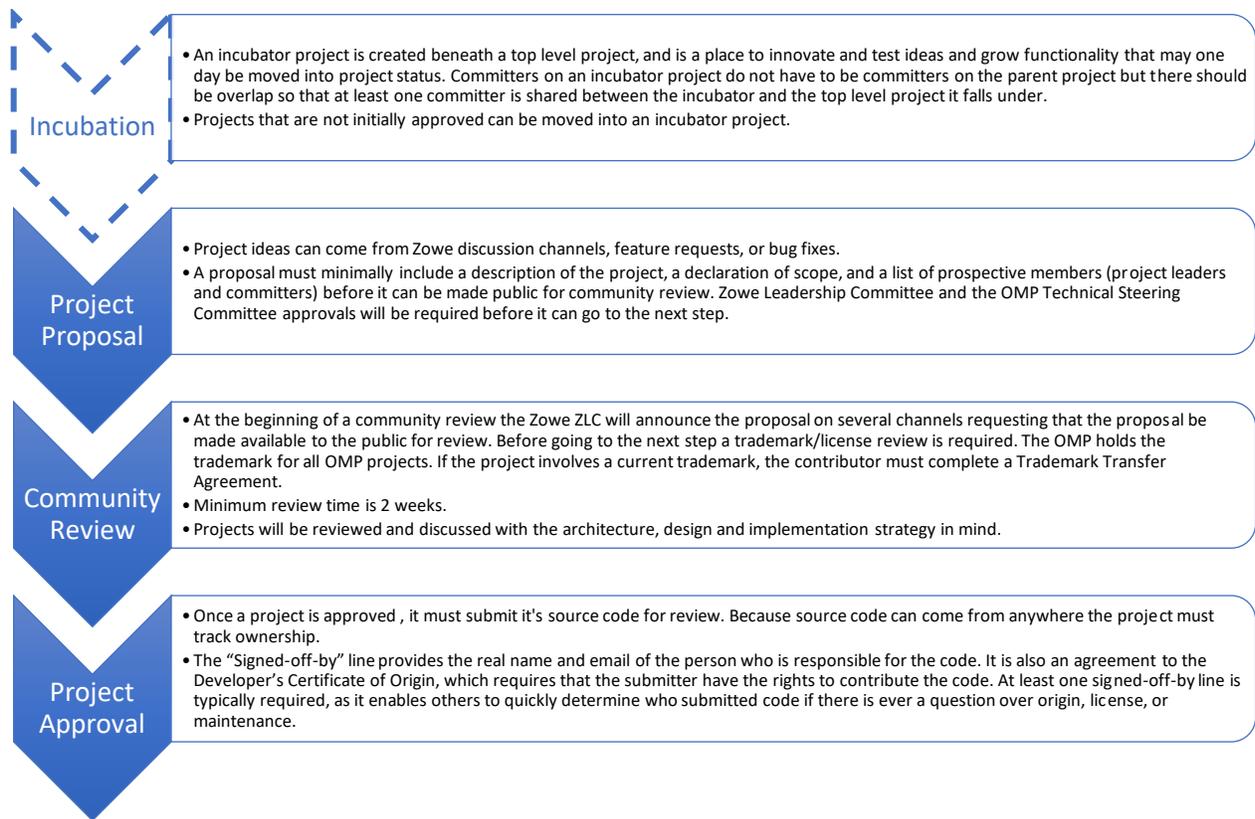
A project is the main operational unit for Zowe. All development occurs within the context of a project. Projects have leaders, developers, source code, builds, downloads, etc... Projects are organized in a hierarchy with a top-level project beneath which individual sub projects may reside.

Scope

Every Zowe project has a charter which must describe the purpose, scope and rules for the project. Sub-projects can operate under the charter of their top-level project. The Zowe ZLC approves the charter of each project. Initiatives and code that is found to be outside the scope of a project may result in termination of the project, as approved by the Zowe ZLC members.

Project Lifecycle

Transitions between project phases are open and transparent.



Development Process

Project Leads will work with the ZLC to produce a development for each iteration cycle.

Each Project must identify, and make available on its web site, the requirements and prioritizations it is working against in the current iteration cycle. In addition, each Project must post an iteration plan showing the date and content of the next major release, including any major milestones, and must keep this plan up to date.

The Committers of a Project or sub-project decide which changes may be committed to the master code base of a Project or sub-project respectively. The ZLC defines the decision process, but that process must include the ability for Committers to veto the change. The decision process employed may change with the phase of development. Common decision processes include:

- Retroactive - changes are proactively made by Committers but can be vetoed by a single Committer.
- Proactive - for efficiency, some code changes from some Committers (e.g. feature additions, bug fixes) may be approved in advance, or approved in principle based on an outline of the work, in which case they may be committed first and changed as needed, with conflicts resolved by majority vote of the Committers of the Project or sub-project, as applicable.

- Three Positive - No code is committed without a vote; three +1 ('yes' votes) with no -1 ('no' votes or vetoes) are needed to approve a code change.

Vetoes must be followed by an explanation for the veto within 24 hours or the veto becomes invalid. Special rules may be established by the Zowe ZLC for Projects with fewer than three Committers.

The master copy of the code base must reside on the Project web site (Public GitHub) where it is accessible to all users, developers and committers. Committers must check their changes and new work into the master code base as promptly as possible (subject to any check-in voting rules that may be in effect) in order to foster collaboration among widely distributed groups and so that the latest work is always available to everyone.

Each Zowe project is responsible for working with Zowe ZLC to establish a release engineering and build process to ensure that builds can be reliably produced on a regular and frequent basis from the master code base and made available for download from the Project web site. Builds in this context are intended to include not only code but also reports, documentation, and courseware.

Each Project is responsible for establishing test plans and the level of testing appropriate for the Project.

All development technical discussions should use the appropriate communication channel. If discussions are held offline, then a summary must be shared through the appropriate channel to keep the other committers, and any other interested parties up to date.

Licensing

The official licensing for the Zowe top-level project will be the Eclipse Public License (EPL) 2.0. The EPL 2.0 license is Open Source Initiative (OSI) certified and is ideal for multiple enterprises wanting to collaborate on open source software. The weak copy left licensing requirements ensures that any commercial extender of the Zowe technology who makes changes to the base platform must make those derivative works source available to the community. The Zowe community chooses not to add a secondary license for GPL 2.0+ compatibility nor allow dual licensing.

For more information on EPL 2.0 visit the FAQ page here:
<https://www.eclipse.org/legal/epl-2.0/faq.php>

Zowe top-level projects define the standard licensing for their sub-projects. If a project has non-standard licensing requirements, one needs to make a presentation to the Zowe ZLC and the OMP Technical Steering Committee to request their approval. The presentation need only briefly describe the project and why special licensing considerations are necessary.

Source and Intellectual Property Submission Plan

All Zowe committers must be familiar with the [OMP/LF IP Policy](#). Code contributions will be made via the [Developer Certificate of Origin](#).