

# Bylaws

- [Draft Apache Kafka Project Bylaws](#)
  - [Roles and Responsibilities](#)
    - [Committers](#)
    - [Project Management Committee](#)
  - [Decision Making](#)
    - [Voting](#)
    - [Approvals](#)
    - [Vetos](#)
    - [Actions](#)

## Draft Apache Kafka Project Bylaws

This document defines the bylaws under which the Apache Kafka project operates. It defines the roles and responsibilities of the project, who may vote, how voting works, how conflicts are resolved, etc.

Kafka is a project of the [Apache Software Foundation](#). The foundation holds the copyright on all Apache code including the code in the Kafka codebase. The [foundation FAQ](#) explains the operation and background of the foundation.

Kafka operates under a set of principles known collectively as the 'Apache Way'. If you are new to Apache development, please refer to the [Incubator Project](#) for more information on how Apache projects operate.

## Roles and Responsibilities

Apache projects define a set of roles with associated rights and responsibilities. These roles govern what tasks an individual may perform within the project. The roles are defined in the following sections.

### Committers

The project's Committers are responsible for the project's technical management. Committers have access to and responsibility for all of Kafka's source code repository.

Committer access is by invitation only and must be approved by lazy consensus of the active PMC members. A Committer is considered emeritus by their own declaration or by not contributing in any form to the project for over six months. Project contributions considered for this purpose are any code contribution, code review, JIRA or mailing list discussion, documentation contributions, or user support on the mailing list. An emeritus committer may request reinstatement of commit access from the PMC which, if granted, will be sufficient to restore him or her to active committer status.

Commit access can be revoked by consensus by the active PMC (except the committer in question if they are also a PMC member).

All Apache committers are required to have a signed Individual Contributor License Agreement (CLA) on file with the Apache Software Foundation. There is a [Committer FAQ](#) which provides more details on the requirements for Committers.

A committer who makes a sustained contribution to the project may be invited to become a member of the PMC.

### Project Management Committee

The PMC is responsible to the board and the ASF for the management and oversight of Apache Kafka's codebase. The responsibilities of the PMC include

- Deciding what is distributed as products of the Apache Kafka project. In particular all releases must be approved by the PMC.
- Maintaining the project's shared resources, including the codebase repository, mailing lists, websites.
- Speaking on behalf of the project.
- Resolving license disputes regarding products of the project.
- Nominating new PMC members and committers.
- Maintaining these bylaws and other guidelines of the project.

Membership of the PMC is by invitation only and must be approved by a lazy consensus of PMC members. A PMC member is considered emeritus by their own declaration or by no contribution to the project for a six month period. Project contributions considered for this purpose are code contribution, code review, JIRA or mailing list discussion, documentation contributions, or user support on the mailing list. An emeritus member may request reinstatement to the PMC, which, if granted, will be sufficient to restore him or her to active PMC member.

Membership of the PMC can be revoked by a consensus vote of all the active PMC members other than the member in question.

The chair of the PMC is appointed by the ASF board. The chair is an office holder of the Apache Software Foundation (Vice President, Apache Kafka) and has primary responsibility to the board for the management of the projects within the scope of the Kafka PMC. The chair reports to the board quarterly on developments within the Kafka project.

When the current chair of the PMC resigns, the PMC votes to recommend a new chair using lazy consensus, but the decision must be ratified by the Apache board.

## Decision Making

Within the Kafka project, different types of decisions require different forms of approval. For example, the previous section describes several decisions which require 'lazy consensus' approval. This section defines how voting is performed, the types of approvals, and which types of decision require which type of approval.

## Voting

Decisions regarding the project are made by votes on the primary project development mailing list ([dev@kafka.apache.org](mailto:dev@kafka.apache.org)). Where necessary, PMC voting may take place on the private Kafka PMC mailing list. Votes are clearly indicated by subject line starting with [VOTE]. Votes may contain multiple items for approval and these should be clearly separated. Voting is carried out by replying to the vote mail. Voting may take four flavors

Vote	Meaning
+1	'Yes,' 'Agree,' or 'the action should be performed.'
+0	Neutral about the proposed action (or mildly negative but not enough so to want to block it).
-1	This is a negative vote. On issues where consensus is required, this vote counts as a <b>veto</b> . All vetoes must contain an explanation of why the veto is appropriate. Vetoes with no explanation are void. It may also be appropriate for a -1 vote to include an alternative course of action.

All participants in the Kafka project are encouraged to show their agreement with or against a particular action by voting. For technical decisions, only the votes of active committers are binding. Non-binding votes are still useful for those with binding votes to understand the perception of an action in the wider Kafka community. For PMC decisions, only the votes of active PMC members are binding.

Voting can also be applied to changes already made to the Kafka codebase. These typically take the form of a veto (-1) in reply to the commit message sent when the commit is made. Note that this should be a rare occurrence. All efforts should be made to discuss issues when they are still patches before the code is committed.

Only active (i.e. non-emeritus) committers and PMC members have binding votes.

## Approvals

These are the types of approvals that can be sought. Different actions require different types of approvals.

Approval Type	Definition
Lazy Consensus	Lazy consensus requires 3 binding +1 votes and no binding vetoes.
Lazy Majority	A lazy majority vote requires 3 binding +1 votes and more binding +1 votes than -1 votes.
Lazy Approval	An action with lazy approval is implicitly allowed unless a -1 vote is received, at which time, depending on the type of action, either lazy majority or lazy consensus approval must be obtained.
2/3 Majority	Some actions require a 2/3 majority of active committers or PMC members to pass. Such actions typically affect the foundation of the project (e.g. adopting a new codebase to replace an existing product). The higher threshold is designed to ensure such changes are strongly supported. To pass this vote requires at least 2/3 of binding vote holders to vote +1.

## Vetos

A valid, binding veto cannot be overruled. If a veto is cast, it must be accompanied by a valid reason explaining the reasons for the veto. The validity of a veto, if challenged, can be confirmed by anyone who has a binding vote. This does not necessarily signify agreement with the veto - merely that the veto is valid.

If you disagree with a valid veto, you must lobby the person casting the veto to withdraw their veto. If a veto is not withdrawn, the action that has been vetoed must be reversed in a timely manner.

## Actions

Actions	Description	Approval	Binding Votes	Minimum Length	Mailing List
Code Change	A change made to a codebase of the project and committed by a committer. This includes source code, documentation, website content, etc.	one +1 from a committer who has not authored the patch followed by a Lazy approval (not counting the vote of the contributor), moving to lazy majority if a -1 is received	Active committers	0	JIRA or Review Board (with notification sent to <a href="mailto:dev@kafka.apache.org">dev@kafka.apache.org</a> )

Release Plan	Defines the timetable and actions for a release. The plan also nominates a Release Manager.	Lazy majority	Active committers	3	dev@kafka.apache.org
Product Release	When a release of one of the project's products is ready, a vote is required to accept the release as an official release of the project.	Lazy Majority	Active PMC members	3	dev@kafka.apache.org
Adoption of New Codebase	When the codebase for an existing, released product is to be replaced with an alternative codebase. If such a vote fails to gain approval, the existing code base will continue. This also covers the creation of new sub-projects <i>and submodules</i> within the project.	2/3 majority	Active PMC members	6	dev@kafka.apache.org
New Committer	When a new committer is proposed for the project.	Lazy consensus	Active PMC members	7	private@kafka.apache.org
New PMC Member	When a committer is proposed for the PMC.	Lazy consensus	Active PMC members	3	private@kafka.apache.org
Emeritus PMC Member re-instatement	When an emeritus PMC member requests to be re-instated as an active PMC member.	Lazy consensus	Active PMC members	3	private@kafka.apache.org
Emeritus Committer re-instatement	When an emeritus committer requests to be re-instated as an active committer.	Lazy consensus	Active PMC members	3	private@kafka.apache.org
Committer Removal	When removal of commit privileges is sought. <b>Note:</b> Such actions will also be referred to the ASF board by the PMC chair.	Consensus	Active PMC members (excluding the committer in question if a member of the PMC).	6	private@kafka.apache.org
PMC Member Removal	When removal of a PMC member is sought. <b>Note:</b> Such actions will also be referred to the ASF board by the PMC chair.	Consensus	Active PMC members (excluding the member in question).	6	private@kafka.apache.org
Modifying Bylaws	Modifying this document.	2/3 majority	Active PMC members	6	dev@kafka.apache.org