Criteria for Code Submissions

We prefer to receive code contributions in the form of github pull requests. When you submit a pull request, there are a few things to keep in mind. Please ensure code changes:

- **Are Small.** Submissions should be for a single feature/bugfix, don't group submissions together.
- **Are Green.** All code that is committed to Geode must pass ./gradlew precheckin.
- **Have a JIRA.** Changes should should be associated with a ticket in jira.
- **Are Discussed.** Changes that affect public API or introduce new features must be discussed on the mailing list. Geode is a mature product with many active users in production, so new APIs or features need to meet a high standard of quality. If a feature can be developed as an extension of Geode that may be more desirable - see Public APIs.
- **Have a backwards compatible API.** All changes to the public API must be backwards compatible, with the exception of removing deprecated features. Removing deprecated features also requires discussion.
- **Have backwards compatible messaging.** Geode supports rolling upgrades on a live system. Message serialization and disk persistence must be backwards compatible with previous versions. This includes algorithmic changes that don't affect the serialization format. Geode has a framework for backwards compatible serialization - see Managing backward-compatibility.
- **Follow Style Guidelines.** Code should adhere to the Code Style Guide.
- **Have tests.** New features must be accompanied by tests. Bug fixes should also include a test for the bug. See Writing tests.
- **Have few dependencies.** Avoid introducing new library dependencies when possible. Licenses for third-party dependencies must conform to Apache requirements.
- **Are Safe.** Geode is a heavily concurrent product. Code changes should be carefully considered with regard to thread safety and distributed system safety.
- **Are Secure.** Geode should be secure by default. Don't introduce new vulnerabilities or unsecured backdoors.

Changing the Public API

If you making changes to the public API, make sure dev@geode.apache.org is aware that the changes will affect the public API by sending an email with [DISCUSS] in the subject to describe your changes. Due to the large number of Geode users, we try to avoid breaking compatibility between versions.

Backwards Compatibility

In general, nothing can be removed or modified in way that is not backwards compatible. If something is deprecated, it can be removed after a major release, provided there is a discussion on the dev list. Backwards compatibility affects API's, client applications, and WAN clusters. Support for rolling upgrades is another important consideration.

Javadocs and @since tags

All public and internal APIs should have useful javadocs. For all public APIs (anything not in **/internal/**), new additions to the javadocs should include an @since tag to let the users know when the feature was added. Eg

```
/**
   * Returns a collection of all of the CacheServers
   * that can serve the contents of this Cache> to clients.
   *
   * @see #addCacheServer
   *
   * @since 5.7
   */
  public List<CacheServer> getCacheServers();
```

Any feature which is deprecated should specify a link to the new alternative.

It's best to start small with bug fixes or smaller features. Please feel free to discuss things and ask for help on the mailing list if you are interested in contributing.