Item 90
git_comments:

git_commits:

1. **summary:** GEODE-2644: Adding memberName to the log pattern (#5817)
   message: GEODE-2644: Adding memberName to the log pattern (#5817)

github_issues:

github_issues_comments:

github_pulls:

1. **title:** GEODE-2644: Adding memberName to the log pattern
   body: Thank you for submitting a contribution to Apache Geode. In order to streamline the review of the contribution we ask you to ensure the following steps have been taken: ### For all changes: - [ ] Is there a JIRA ticket associated with this PR? Is it referenced in the commit message? - [ ] Has your PR been rebased against the latest commit within the target branch (typically `develop`)? - [ ] Is your initial contribution a single, squashed commit? - [ ] Does `gradlew build` run cleanly? - [ ] Have you written or updated unit tests to verify your changes? - [ ] If adding new dependencies to the code, are these dependencies licensed in a way that is compatible for inclusion under [ASF 2.0](http://www.apache.org/legal/resolved.html#category-a)? ### Note: Please ensure that once the PR is submitted, check Concourse for build issues and submit an update to your PR as soon as possible. If you need help, please send an email to dev@geode.apache.org.
2. **title:** GEODE-2644: Adding memberName to the log pattern
   body: Thank you for submitting a contribution to Apache Geode. In order to streamline the review of the contribution we ask you to ensure the following steps have been taken: ### For all changes: - [ ] Is there a JIRA ticket associated with this PR? Is it referenced in the commit message? - [ ] Has your PR been rebased against the latest commit within the target branch (typically `develop`)? - [ ] Is your initial contribution a single, squashed commit? - [ ] Does `gradlew build` run cleanly? - [ ] Have you written or updated unit tests to verify your changes? - [ ] If adding new dependencies to the code, are these dependencies licensed in a way that is compatible for inclusion under [ASF 2.0](http://www.apache.org/legal/resolved.html#category-a)? ### Note: Please ensure that once the PR is submitted, check Concourse for build issues and submit an update to your PR as soon as possible. If you need help, please send an email to dev@geode.apache.org.

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Provide ability to configure Geode appenders in log4j2.xml
   description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
2. **summary:** Provide ability to configure Geode appenders in log4j2.xml
   description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control

over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.

3. **summary:** Provide ability to configure Geode appenders in log4j2.xml
   description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
   label: code-design
4. **summary:** Provide ability to configure Geode appenders in log4j2.xml
   description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
5. **summary:** Provide ability to configure Geode appenders in log4j2.xml
   description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
6. **summary:** Provide ability to configure Geode appenders in log4j2.xml
   description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
7. **summary:** Provide ability to configure Geode appenders in log4j2.xml
   description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
8. **summary:** Provide ability to configure Geode appenders in log4j2.xml
   description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
   label: code-design
9. **summary:** Provide ability to configure Geode appenders in log4j2.xml

description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
label: code-design

10. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
11. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
12. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
13. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
14. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
15. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-

controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.

16. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
17. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
18. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
19. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
20. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
21. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control

over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
label: code-design

22. **summary:** Provide ability to configure Geode appenders in log4j2.xml
description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
label: code-design
23. **summary:** Provide ability to configure Geode appenders in log4j2.xml
description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
label: code-design
24. **summary:** Provide ability to configure Geode appenders in log4j2.xml
description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
label: code-design
25. **summary:** Provide ability to configure Geode appenders in log4j2.xml
description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
label: code-design
26. **summary:** Provide ability to configure Geode appenders in log4j2.xml
description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
label: code-design
27. **summary:** Provide ability to configure Geode appenders in log4j2.xml
description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code

would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
label: code-design

28. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
29. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
30. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
31. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
32. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
33. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-

controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
label: code-design

34. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
35. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
36. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
37. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
38. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
39. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the

JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
label: code-design

40. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
41. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
42. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
43. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
44. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
45. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code

would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.

46. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
47. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
48. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
49. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
50. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
51. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
52. **summary:** Provide ability to configure Geode appenders in log4j2.xml

description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.

53. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
54. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
55. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
56. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
57. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.
    label: code-design
58. **summary:** Provide ability to configure Geode appenders in log4j2.xml
    description: Presently Geode dynamically creates, adds and removes AlertAppender and LogWriterAppender by manipulating log4j2 core API. We should move the bulk of the Appender functionality to internal classes and just leave the Appenders registered with log4j2 during the life of the JVM. This allows us to enable and configure our Appenders via log4j2.xml and control the Cache-

controlled lifecycle internally without having to add and remove custom Appender instances. The code would then become simpler, we could avoid invoking log4j2 core APIs, and users would have control over configuring our use of log4j2 completely within the .xml file. Presently, a user cannot configure our AlertAppender or LogWriterAppender in log4j2.xml.

jira_issues_comments:

1. Commit 2af3aab63061a49861e7058c37a3da51a6d81087 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=2af3aab] GEODE-2644: Cleanup logging classes
2. Commit 9ca5355b46f89f2eb00e1fe041159b5faec898a6 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=9ca5355] GEODE-2644: Cleanup config classes (#2518) Add LogConfigSupplier and some additional changes to LogConfig for logging changes.
3. **body:** Commit 8c7a7396f992d96e8086d6492e25c97c2092a3da in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=8c7a739] GEODE-2644: Cleanup LogService (#2533) * Rename LogService getConfigInformation to getConfigurationInfo. * Rename variables, remove SuppressWarnings, remove unused code.
   label: code-design
4. Commit 076fff86683d3f2ed14c978dd5ba4c3031ae6a64 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=076fff8] GEODE-2644: Improve GfshParserAutoCompletionTest assertions (#2534) Improve assertions so that failures in this test can be diagnosed without using the IDE debugger.
5. Commit 65725eb7e466be85260b215cf3a0917f931a21c5 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=65725eb] GEODE-2644: Cleanup logging tests (#2519)
6. Commit 78c4aac8dc18e43f26b14013e106e2c8a7b4c982 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=78c4aac] GEODE-2644: Use LINE_SEPARATOR in logging classes (#2525) Use org.apache.commons.lang.SystemUtils.LINE_SEPARATOR instead of \n.
7. Commit 477cf8c8bdfbcadb7456a1e4255142f9a03b10a8 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=477cf8c] GEODE-2644: Cleanup RuntimeDistributionConfigImpl (#2544)
8. **body:** Commit 763e4177f201eff9b140654e6858152a1a184b8c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=763e417] GEODE-2644: Cleanup logging related files Cleanup files that helped fix logging related test failures or helped with reviewing and understanding code while working on GEODE-2644. * Remove warnings and improve code cleanliness * Improve testing and failure messages
   label: code-design
9. **body:** Commit 763e4177f201eff9b140654e6858152a1a184b8c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=763e417] GEODE-2644: Cleanup logging related files Cleanup files that helped fix logging related test failures or helped with reviewing and understanding code while working on GEODE-2644. * Remove warnings and improve code cleanliness * Improve testing and failure messages
   label: code-design
10. Commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=5e61da5] GEODE-2644: Add LogFileAssert and LogLevelAssert assertions Introduce custom AssertJ assertions for logging tests. LogFileAssert is different from FileAssert by supporting the matching of partial string contains rather than strict full content matching.
11. Commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=731cd2a] GEODE-2644: Add AlertingTest JUnit Category
12. Commit a77c94b349b3ed7fc873fc29f6852f9936843790 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=a77c94b] GEODE-2644: Add LogConfig and StatisticsConfig for logging
13. Commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=02c3cae] GEODE-2644: Make AlertAppender optional and support log4j2.xml AlertAppender is now configured in log4j2.xml

and it supports sessions that correspond with Cache lifecycle. This allows Geode to pause and resume AlertAppender without resorting to dynamically adding and removing appenders. List of changes: * Change AlertAppender to be pausable and session-oriented * Make AlertAppender support configuration from log4j2.xml * Log4j2 Core dependency is now optional * Internal Alerting interfaces allow Alerting service to be pluggable * Reduce coupling between Alerting and the rest of Geode * Greatly increase test coverage for Alerting

14. Commit cd2bba13e2fb7e79f7634053a6c612050b328e11 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=cd2bba1] GEODE-2644: Expand DistributedSystemMXBean tests Extract these tests from DistributedSystemDUnitTest: * DistributedSystemMXBeanDistributedTest * DistributedSystemMXBeanWithAlertsDistributedTest * DistributedSystemMXBeanWithNotificationsDistributedTest Add new DistributedSystemMXBeanIntegrationTest. Add many new Alert notification tests.
15. Commit 591f3f7840ce38c6100bff2040f396ad2469175e in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=591f3f7] GEODE-2644: Make LogWriterAppender optional and support log4j2.xml LogWriterAppender is now configured in log4j2.xml and it supports sessions that correspond with Cache lifecycle. This allows Geode to pause and resume LogWriterAppender and GeodeConsoleAppender without resorting to dynamically adding and removing appenders. When the Cache exists, log events stop going to the Console and instead go to the Geode log file. Whenever the Cache does not exist, log events go only to the Console. These changes remove as much of the Log4j2 Core manipulation as possible. Also fixed: * GEODE-5789: Geode now updates log level of all Geode loggers. List of changes: * Change LogWriterAppender to be pausable and session-oriented * Add GeodeConsoleAppender which is pausable * Log4j2 Core dependency is now optional * Internal Logging interfaces allow Logging service to be pluggable * Log4j2 xml integration testing now uses JUnit Rule from Log4j2 * Reduce coupling between Logging and the rest of Geode * Greatly increase test coverage for Logging
16. Commit 688f58a4a75cc1297faa1dda201b971346c3a9fe in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=688f58a] GEODE-2644: Update session state caching dependencies for logging
17. Commit 6691d310e1263966991e584f712156d919dabecf in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=6691d31] GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark Remove log level manipulation from benchmark. The internal API that was being used has been deleted and it only altered the log level of one logger: Cache.getLogger, so the impact of this is neglible.
18. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design
19. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline

diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design

20. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
    label: code-design
21. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit

Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design

22. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
    label: code-design
23. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
    label: code-design
24. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit

6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design

25. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design
26. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files"

This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design

27. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
    label: code-design
28. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
    label: code-design
29. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky

SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design

30. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
    label: code-design
31. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files"

This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design

32. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
    label: code-design
33. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
    label: code-design
34. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update

session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design

35. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design
36. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644:

Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design

37. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
    label: code-design
38. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
    label: code-design
39. **body:** Commit 743c925efa608719aca1ee6249d3c8ee3901905c in geode's branch refs/heads/feature/GEODE-6068 from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=743c925] Revert GEODE-2644 because offline diskstore commands generate info level logs GFSH forks the offline diskstore commands in a JVM using log4j2-cli.xml which sets the log level to WARN. However, the forked JVM creates a Cache which then starts generating info level statements to

STDOUT. Revert "GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark" This reverts commit 6691d310e1263966991e584f712156d919dabecf. Revert "GEODE-2644: Update session state caching dependencies for logging" This reverts commit 688f58a4a75cc1297faa1dda201b971346c3a9fe. Revert "GEODE-5637: Fix flaky SingleHopClientExecutorWithLoggingIntegrationTest" This reverts commit 0a8f3dffa1ebaa618fafecc15bbc92ba8ba83696. Revert "GEODE-2644: Make LogWriterAppender optional and support log4j2.xml" This reverts commit 591f3f7840ce38c6100bff2040f396ad2469175e. Revert "GEODE-2644: Expand DistributedSystemMXBean tests" This reverts commit cd2bba13e2fb7e79f7634053a6c612050b328e11. Revert "GEODE-2644: Make AlertAppender optional and support log4j2.xml" This reverts commit 02c3cae1d38098259ff6aa4635c6e74f795bdc50. Revert "GEODE-2644: Add LogConfig and StatisticsConfig for logging" This reverts commit a77c94b349b3ed7fc873fc29f6852f9936843790. Revert "GEODE-2644: Add AlertingTest JUnit Category" This reverts commit 731cd2a7d35b086d34f8bbcb94a83c40739ae8ee. Revert "GEODE-2644: Add LogFileAssert and LogLevelAssert assertions" This reverts commit 5e61da5301469cdbd71ac186573f56c38e9a6cf0. Revert "GEODE-2644: Cleanup logging related files" This reverts commit 763e4177f201eff9b140654e6858152a1a184b8c. Revert "GEODE-2644: Define interfaces for new Log4j2 Appenders" This reverts commit efd8a14027195b6dae1dd3e4d53a33696ed318bb.
label: code-design

40. **body:** Commit 145276390ed69cbee715f12c069116e6f2720b5d in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=1452763] GEODE-2644: Cleanup logging related files Cleanup files that helped fix logging related test failures or helped with reviewing and understanding code while working on GEODE-2644. * Remove warnings and improve code cleanliness * Improve testing and failure messages
label: code-design
41. **body:** Commit 145276390ed69cbee715f12c069116e6f2720b5d in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=1452763] GEODE-2644: Cleanup logging related files Cleanup files that helped fix logging related test failures or helped with reviewing and understanding code while working on GEODE-2644. * Remove warnings and improve code cleanliness * Improve testing and failure messages
label: code-design
42. Commit bac021475dc9a27d4e4f5ed47ae0f2f6228db6e1 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=bac0214] GEODE-2644: Add LogFileAssert and LogLevelAssert assertions Introduce custom AssertJ assertions for logging tests. LogFileAssert is different from FileAssert by supporting the matching of partial string contains rather than strict full content matching.
43. Commit f1a2cc27ec2a8b17ad37cccfe0f87ecc1c4bc8f2 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=f1a2cc2] GEODE-2644: Add AlertingTest JUnit Category
44. Commit 3e8b07c17ee6490b6c733e800065003f8c841904 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=3e8b07c] GEODE-2644: Add LogConfig and StatisticsConfig for logging
45. Commit 5ae86bb22514e8afe567c8a20dd8a320ec34d1dd in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=5ae86bb] GEODE-2644: Make AlertAppender optional and support log4j2.xml AlertAppender is now configured in log4j2.xml and it supports sessions that correspond with Cache lifecycle. This allows Geode to pause and resume AlertAppender without resorting to dynamically adding and removing appenders. List of changes: * Change AlertAppender to be pausable and session-oriented * Make AlertAppender support configuration from log4j2.xml * Log4j2 Core dependency is now optional * Internal Alerting interfaces allow Alerting service to be pluggable * Reduce coupling between Alerting and the rest of Geode * Greatly increase test coverage for Alerting
46. Commit 6709a326e2d16661e34c179cb81b6a46209b2812 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=6709a32] GEODE-2644: Expand DistributedSystemMXBean tests Extract these tests from DistributedSystemDUnitTest: * DistributedSystemMXBeanDistributedTest * DistributedSystemMXBeanWithAlertsDistributedTest * DistributedSystemMXBeanWithNotificationsDistributedTest Add new DistributedSystemMXBeanIntegrationTest. Add many new Alert notification tests.
47. Commit 4be9e7aa4cd90ca5ac4410d04aadfc3a4efcbba0 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=4be9e7a] GEODE-2644:

Make LogWriterAppender optional and support log4j2.xml LogWriterAppender is now configured in log4j2.xml and it supports sessions that correspond with Cache lifecycle. This allows Geode to pause and resume LogWriterAppender and GeodeConsoleAppender without resorting to dynamically adding and removing appenders. When the Cache exists, log events stop going to the Console and instead go to the Geode log file. Whenever the Cache does not exist, log events go only to the Console. These changes remove as much of the Log4j2 Core manipulation as possible. Also fixed: * GEODE-5789: Geode now updates log level of all Geode loggers. List of changes: * Change LogWriterAppender to be pausable and session-oriented * Add GeodeConsoleAppender which is pausable * Log4j2 Core dependency is now optional * Internal Logging interfaces allow Logging service to be pluggable * Log4j2 xml integration testing now uses JUnit Rule from Log4j2 * Reduce coupling between Logging and the rest of Geode * Greatly increase test coverage for Logging

48. Commit 9891073212690579beeeaceaca131c1c98419e08 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=9891073] GEODE-2644: Update session state caching dependencies for logging
49. Commit 8db6d11001e676d356593d2b1ce95c26b609f183 in geode's branch refs/heads/develop from [~apache@the9muses.net] [https://gitbox.apache.org/repos/asf?p=geode.git;h=8db6d11] GEODE-2644: Remove log level from MonitorQueryUnderContentionBenchmark Remove log level manipulation from benchmark. The internal API that was being used has been deleted and it only altered the log level of one logger: Cache.getLogger, so the impact of this is neglible.
50. Commit 413800bc16d05df689a2af5c30797f180aad6088 in geode's branch refs/heads/develop from Kirk Lund [https://gitbox.apache.org/repos/asf?p=geode.git;h=413800b] GEODE-7058: Mark log4j-core optional in geode-core Note: this change requires all commits from GEODE-2644 and GEODE-6122.
51. Commit 1d5bbe1fcd6d0f2043a6406a24a817febc94908c in geode's branch refs/heads/release/1.10.0 from Kirk Lund [https://gitbox.apache.org/repos/asf?p=geode.git;h=1d5bbe1] GEODE-7058: Mark log4j-core optional in geode-core Note: this change requires all commits from GEODE-2644 and GEODE-6122. (cherry picked from commit 413800bc16d05df689a2af5c30797f180aad6088)
52. Commit 413800bc16d05df689a2af5c30797f180aad6088 in geode's branch refs/heads/feature/GEODE-6883-mcast-problems from Kirk Lund [https://gitbox.apache.org/repos/asf?p=geode.git;h=413800b] GEODE-7058: Mark log4j-core optional in geode-core Note: this change requires all commits from GEODE-2644 and GEODE-6122.
53. Commit 63034887c70e7bb7ec5e086b5154b626049a84c0 in geode's branch refs/heads/release/1.9.1 from Kirk Lund [https://gitbox.apache.org/repos/asf?p=geode.git;h=6303488] GEODE-7058: Mark log4j-core optional in geode-core Note: this change requires all commits from GEODE-2644 and GEODE-6122. (cherry picked from commit 413800bc16d05df689a2af5c30797f180aad6088)
54. nabarunnag merged pull request #5795: URL: https://github.com/apache/geode/pull/5795 --- This is an automated message from the Apache Git Service. To respond to the message, please log on to GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
55. nabarunnag opened a new pull request #5816: URL: https://github.com/apache/geode/pull/5816 Reverts apache/geode#5795 -- This is an automated message from the Apache Git Service. To respond to the message, please log on to GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
56. nabarunnag merged pull request #5816: URL: https://github.com/apache/geode/pull/5816 --- This is an automated message from the Apache Git Service. To respond to the message, please log on to GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
57. nabarunnag opened a new pull request #5817: URL: https://github.com/apache/geode/pull/5817 Thank you for submitting a contribution to Apache Geode. In order to streamline the review of the contribution we ask you to ensure the following steps have been taken: ### For all changes: - [ ] Is there a JIRA ticket associated with this PR? Is it referenced in the commit message? - [ ] Has your PR been rebased against the latest commit within the target branch (typically `develop`)? - [ ] Is your initial contribution a single, squashed commit? - [ ] Does `gradlew build` run cleanly? - [ ] Have you written or updated unit tests to verify your changes? - [ ] If adding new dependencies to the code, are these dependencies licensed in a way that is compatible for inclusion under [ASF 2.0] (http://www.apache.org/legal/resolved.html#category-a)? ### Note: Please ensure that once the PR is submitted, check Concourse for build issues and submit an update to your PR as soon as possible. If you need help, please send an email to dev@geode.apache.org. -- This is an automated message from the Apache Git Service. To respond to the message, please log

on to GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org

58. nabarunnag merged pull request #5817: URL: https://github.com/apache/geode/pull/5817 -- This is an automated message from the Apache Git Service. To respond to the message, please log on to GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
59. Commit d38189e72d618bc523fb4b0502ed74beab861197 in geode's branch refs/heads/develop from Nabarun Nag [https://gitbox.apache.org/repos/asf?p=geode.git;h=d38189e] GEODE-2644: Adding memberName to the log pattern (#5817)
60. Commit 27e7d34f2d2da84c19783ba8c1ecf28639e0dde6 in geode's branch refs/heads/support/1.12 from Nabarun Nag [https://gitbox.apache.org/repos/asf?p=geode.git;h=27e7d34] GEODE-2644: Adding memberName to the log pattern (#5817) (cherry picked from commit d38189e72d618bc523fb4b0502ed74beab861197)
61. Commit 3ae87364255cf6b8fdec6e27325bdbbdecf2d76e in geode's branch refs/heads/support/1.13 from Nabarun Nag [https://gitbox.apache.org/repos/asf?p=geode.git;h=3ae8736] GEODE-2644: Adding memberName to the log pattern (#5817) (cherry picked from commit d38189e72d618bc523fb4b0502ed74beab861197)