Item 281
git_comments:

1. eat the interrupt and check for exit conditions
2. there should be a transaction timeout that keeps this thread from sitting around forever if the client goes away The above was done by setting afterCompletionCancelled in txState during cleanup. When client departed, the transaction/JTA will be timed out and cleanup code will be executed.
3. * Licensed to the Apache Software Foundation (ASF) under one or more contributor license * agreements. See the NOTICE file distributed with this work for additional information regarding * copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance with the License. You may obtain a * copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software distributed under the License * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express * or implied. See the License for the specific language governing permissions and limitations under * the License.
4. eat the interrupt and check for exit conditions
5. * Licensed to the Apache Software Foundation (ASF) under one or more contributor license * agreements. See the NOTICE file distributed with this work for additional information regarding * copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance with the License. You may obtain a * copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software distributed under the License * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express * or implied. See the License for the specific language governing permissions and limitations under * the License.
6. * * This class ensures that beforeCompletion and afterCompletion are executed in the same thread. * * @since Geode 1.7.0
7. * Licensed to the Apache Software Foundation (ASF) under one or more contributor license * agreements. See the NOTICE file distributed with this work for additional information regarding * copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance with the License. You may obtain a * copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software distributed under the License * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express * or implied. See the License for the specific language governing permissions and limitations under * the License.
8. * * stop waiting for an afterCompletion to arrive and just exit
9. * Licensed to the Apache Software Foundation (ASF) under one or more contributor license * agreements. See the NOTICE file distributed with this work for additional information regarding * copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance with the License. You may obtain a * copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software distributed under the License * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express * or implied. See the License for the specific language governing permissions and limitations under * the License.
10. * Licensed to the Apache Software Foundation (ASF) under one or more contributor license * agreements. See the NOTICE file distributed with this work for additional information regarding * copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance with the License. You may obtain a * copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software distributed under the License * is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express * or implied. See the License for the specific language governing permissions and limitations under * the License.
11. give the thread a chance to get past the "finished" check by waiting until checkCancelInProgress is called
12. * Licensed to the Apache Software Foundation (ASF) under one or more contributor license * agreements. See the NOTICE file distributed with this work for additional information regarding * copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance with the License. You may obtain a * copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software distributed under the License * is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express * or implied. See the License for the specific language governing permissions and limitations under * the License.
13. * * for client/server JTA transactions we need to have a single thread handle both beforeCompletion * and afterCompletion so that beforeCompletion can obtain locks for the afterCompletion step. * This is that thread
14. if there was a beforeCompletion call then there will be a thread sitting in the waiting pool to execute afterCompletion. Otherwise

git_commits:

1. **summary:** Feature/geode 5624 Use a single thread to ensure beforeCompletion and afterCompletion are executed by the same thread. (#2388)
   message: Feature/geode 5624 Use a single thread to ensure beforeCompletion and afterCompletion are executed by the same thread. (#2388)

github_issues:

github_issues_comments:

github_pulls:

1. **title:** Feature/geode 5624 Use a single thread to ensure beforeCompletion and afterCompletion are executed by the same thread.
   body:

github_pulls_comments:

github_pulls_reviews:

1. change this to "while (!condition.getAsBoolean())" so the impl match the method name "waitForCondition". So you will need to change the callers to flip the logic of the expressions they pass in. change: "!finished" to "finished" change: "status == -1 && !cancelled" to "status != -1 || cancelled".
2. this can be simplified to "() -> status == -1 && !cancelled" No need for the "return" or the "{();}"
3. this can be simplified to "() -> !finished"
4. Move this up to the first line of the method. It should be before we call "executeBeforeCompletion".
5. "this" refers to the "InOrder" line.
6. verify that execute was called first. This happens synchronously so it should have already happened. executeBeforeCompletion does not return until beforeCompletion.doOp is called (and completed). So remove the await on beforeCompletion.doOp. Also add verification of beforeCompletion.isFinished() is true. The 4th line of verification should be that we did afterCompletion.doOp and you need an await on that line
7. I was wrong about this. We have a mocked beforeCompletion and afterCompletion so calling beforeCompletion.execute does not wait for doOp to finish

jira_issues:

jira_issues_comments: