

git_comments:

1. * @file This file implements the rolled log deletion. @section license License 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.
2. * * A min_count of zero indicates a request to try to keep all rotated logs * around. By setting min_count to INT_MAX in these cases, we make the rolled * log deletion priority small. * * @note This cannot have a zero value because it is used as the denominator * in a division operation when calculating the log deletion preference.
3. Return the highest priority candidate among the candidates of that type.
4. Select the highest priority type (diags.log, traffic.out, etc.) from which to select a candidate.
5. * * @param[in] logname The unrolled log name. * * @param[in] min_count The minimum number of rolled files to try to keep * around when deleting rolled logs. A zero indicates a desire to keep all * rolled logs around. * * @note The min_count is used as a part of a calculation to determine which * set of deletion candidates should be used for selecting a rolled log file * to delete. If space is particularly constrained, even LogDeletingInfo * instances with a min_count of 0 may be selected for deletion.
6. * The owning references to the set of LogDeletingInfo added to the below * hash map.
7. * @file This contains the rotated log deletion mechanism. @section license License 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. * * RolledLogDeleter is responsible for keeping track of rolled log candidates * and presenting them for deletion in a prioritized order based on size and * last modified time stamp. * * Terminology: * * log type: An unrolled log name that represents a category of rolled log * files that are candidates for deletion. This may be something like * diags.log, traffic.out, etc. * * candidate: A rolled log file which is a candidate for deletion at some * point. This may be something like: * squid.log_some.hostname.com.20191125.19h00m04s-20191125.19h15m04s.old.
9. * The minimum number of rolled log files to try to keep around. * * @note This is guaranteed to be a positive (non-zero) value.
10. * Register a new log type for candidates for log deletion. * * @param[in] log_type The unrolled name for a set of rolled log files to * consider for deletion. This may be something like diags.log, for example. * * @param[in] rolling_min_count The minimum number of rolled log files to * keep around.
11. * Configure rolled log deletion for a set of logs. * * This contains the configuration and set of log deletion candidates for a set * of log files associated with logname. There will be an instance of this for * diags.log and its associated rolled log files, one for traffic.out, etc.
12. * Evaluate a rolled log file to see whether it is a candidate for deletion. * * If the rolled log file is a valid candidate, it will be stored and considered * for deletion upon later calls to deleteALogFile. * * @param[in] log_path The rolled log file path. * * @param[in] file_size The size of the rolled log file. * * @param[in] modification_time The time the rolled log file was last modified. * candidate for deletion. * * @return True if the rolled log file is a deletion candidate, false otherwise.
13. * The unrolled log name (such as "diags.log").
14. * Whether there are any candidates for possible deletion. * * @return True if there are candidates for deletion, false otherwise.
15. * The number of tracked candidates.
16. * The set of candidates for deletion keyed by log_type.
17. * Clear the internal candidates array.

18. * Retrieve the next rolled log file to delete. * * This removes the returned rolled file from the candidates list. * * @return The next rolled log candidate to delete or nullptr if there is no * such candidate.
19. * Retrieve the number of rolled log deletion candidates. * * @return The number of rolled logs that are candidates for deletion.
20. * The filename for this rolled log deletion candidate. * * For example:
/var/log/my_log.log_a_host_name.20191122.20h18m35s-20191122.20h18m51s.old
21. The second candidate should be the second oldest.
22. Everything has been taken.
23. Add some candidates.
24. Even with so many diags.log files, the traffic.out one should be selected first because the min_count of diags.log is 0.
25. Stub
26. The first candidate should be the oldest modified one.
27. The user has requested a higher number of traffic.out files, but since there are so many of them, the oldest of them should be selected next.
28. Now, there's only traffic.out files.
29. * @file Catch-based tests for RolledLogDeleter. @section license License 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.
30. The second candidate should be the remaining one.
31. Intentionally insert them out of order (that is the first one to delete is the second added).
32. The previous tests verify selection within a log_type which is done based * upon last modified time stamp. These tests focus on selection of * candidates across log types, which is based upon number of candidates and * the desired min_count.
33. Since the time stamps of both are the same, selection should be made based upon min_count.
34. Now there's only diags.log files.
35. Next, squid.log should be chosen.
36. Intentionally insert them out of order.
37. The third candidate should be the remaining one.
38. * @file Test a plugin's interaction with the logging interface. @section license License 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.
39. This is made to match the std::filesystem::path::filename behavior:
<https://en.cppreference.com/w/cpp/filesystem/path/filename>
40. auto create
41. auto_created
42. The majority of register_rolled_log_auto_delete() updates come in through LogObject. However, not all ATS logs are managed by LogObject. The following register these other core logs for log rotation deletion.
43. The set of files in the logs dir may change between iterations to check for logs to delete. To deal with this, we simply clear our internal candidates metadata and regenerate it on each iteration.
44. Nothing to do if auto-deletion is not configured.
45. * Register rolled logs of logname for auto-deletion when there are space * constraints. * * @param[in] logname The name of the unrolled log to register, such as * "diags.log". * * @param[in] rolling_min_count The minimum amount of rolled logs of logname * to try to keep around. A value of 0 expresses a desire to keep all rolled * files, if possible.
46. auto_created

47. minimum number of rolled logs to be kept, 0 no limit

48. rolling_min_count

git_commits:

1. summary: auto delete rolled log file fixes

message: auto delete rolled log file fixes This fixes auto delete registration for: - core log files that were not getting deleted (such as error.log and manager.log) as well as... - plugin log files. To make log deletion registration more automatic, I placed it in LogObject initialization so that it doesn't have to happen in as many disparate places. By doing this, plugin log auto delete comes for free. This is why nothing in the core plugin implementation needed to be changed to register their log files, for instance. Not all log objects are managed by LogObject, however, so we still have some manual calls to registration in LogConfig for core log files (such as traffic.out and diags.log, for example). Before this change, LogConfig manipulated deleting_info to create and maintain candidates. To encapsulate this logic, I created RolledLogDeleter and changed LogConfig to interact with an instance of that. This simplified the LogConfig logic while also enabling unit testability of the main log deletion feature. This also: - Adds a rolling_max_count autest. - Fixes a shutdown leak concerning LogDeletingInfo. - Fixes a paramater mismatch between TextLogObject and LogObject.

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

jira_issues_comments: