

**git\_comments:**

1. Replicating the behavior of how logging module was loaded in logging\_config.py
2. Create the directory structure
3. Replace slashes by dots
4. Add the `__init__` for the directories This is required for Python 2.7
5. Launch a process through DagFileProcessorAgent, which will try reload the logging module.
6. `shutil.rmtree(self.settings_root)` Reset config
7. **comment:** Since we are reloading logging config not creating this file, we should expect it to be nonexistent.  
**label:** code-design
8. **comment:** Starting dag processing with 0 max\_runs to avoid redundant operations.  
**label:** code-design

**git\_commits:**

1. **summary:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent (#4253)  
**message:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent (#4253)

**github\_issues:****github\_issues\_comments:****github\_pulls:**

1. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent  
**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests `__OR__` does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](<http://chris.beams.io/posts/git-commit/>)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`
2. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent  
**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests `__OR__` does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is

instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`

3. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent

**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`

4. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent

**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`

5. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent

**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not

need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`

6. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent

**body:** #### Jira - [x] My PR addresses the following [Airflow Jira]

(<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. -

<https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`

7. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent

**body:** #### Jira - [x] My PR addresses the following [Airflow Jira]

(<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. -

<https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`

8. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent

**body:** #### Jira - [x] My PR addresses the following [Airflow Jira]

(<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. -

<https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about

my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`

9. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent

**body:** #### Jira - [x] My PR addresses the following [Airflow Jira]

(<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. -

<https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`

10. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent

**body:** #### Jira - [x] My PR addresses the following [Airflow Jira]

(<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. -

<https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`

11. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent  
**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](<http://chris.beams.io/posts/git-commit/>)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`
12. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent  
**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](<http://chris.beams.io/posts/git-commit/>)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`
13. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent  
**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utils/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](<http://chris.beams.io/posts/git-commit/>)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with

- a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autotest documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`
14. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent  
**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utlis/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](<http://chris.beams.io/posts/git-commit/>)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autotest documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`  
**label:** code-design
15. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent  
**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utlis/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](<http://chris.beams.io/posts/git-commit/>)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autotest documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`
16. **title:** [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent  
**body:** #### Jira - [x] My PR addresses the following [Airflow Jira] (<https://issues.apache.org/jira/browse/AIRFLOW-3414>) issues and references them in the PR title. - <https://issues.apache.org/jira/browse/AIRFLOW-3414> #### Description - [x] Here are some details about my PR, including screenshots of any UI changes: `six.moves.reload\_module` will throw exception if the module it is trying to reload has not been loaded yet. This will be the case if we loaded custom logging class and then try to reload the default logging module. This change will make sure we always load the already loaded logging module. #### Tests - [x] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: tests/utlis/test\_dag\_processing.py Note that the unit test covers the case where we loaded BOTH default logging module and are trying to reload custom logging module but not the case where default logging module is not loaded. This is because the test env already has the default logging module loaded and it would be risky to modify the loaded module list. The case is

instead covered in local tests( using custom logging and the cluster is healthy). #### Commits - [x] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" #### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. #### Code Quality - [x] Passes `flake8`

## github\_pulls\_comments:

1. # [Codecov](https://codecov.io/gh/apache/incubator-airflow/pull/4253?src=pr&el=h1) Report > Merging [#4253](https://codecov.io/gh/apache/incubator-airflow/pull/4253?src=pr&el=desc) into [master] (https://codecov.io/gh/apache/incubator-airflow/commit/d1d612e5c78c89487a05ab3cc86f7d3472ac2a5d?src=pr&el=desc) will **\*\*decrease\*\*** coverage by ` $<.01\%$ `. > The diff coverage is `85.71%`. [![Impacted file tree graph](https://codecov.io/gh/apache/incubator-airflow/pull/4253/graphs/tree.svg?width=650&token=WdLKLKHOAU&height=150&src=pr)](https://codecov.io/gh/apache/incubator-airflow/pull/4253?src=pr&el=tree) ``diff @@ Coverage Diff @@ ## master #4253 +/- ##  
===== - Coverage 78.07% 78.07% -0.01%  
===== Files 201 201 Lines 16455 16458 +3  
===== + Hits 12848 12850 +2 - Misses 3607 3608  
+1 `` | [Impacted Files](https://codecov.io/gh/apache/incubator-airflow/pull/4253?src=pr&el=tree) |  
Coverage  $\Delta$  | |---|---|---| | [airflow/settings.py](https://codecov.io/gh/apache/incubator-airflow/pull/4253/diff?src=pr&el=tree#diff-YWlyZmxvdY9zZXR0aW5ncy5weQ==) | `80.41% <100%>`  
( $\emptyset$ ) | :arrow\_up: | | [airflow/logging\_config.py](https://codecov.io/gh/apache/incubator-airflow/pull/4253/diff?src=pr&el=tree#diff-YWlyZmxvdY9sb2dnZW5nX2NvbWZpZy5weQ==) |  
`97.56% <100%> (+0.06%)` | :arrow\_up: | | [airflow/utis/dag\_processing.py]  
(https://codecov.io/gh/apache/incubator-airflow/pull/4253/diff?src=pr&el=tree#diff-YWlyZmxvdY91dGlscy9kYWdfcHJvY2Vzc2luZy5weQ==) | `59.67% <66.66%> (+0.14%)` | :arrow\_up:  
| | [airflow/models.py](https://codecov.io/gh/apache/incubator-airflow/pull/4253/diff?src=pr&el=tree#diff-YWlyZmxvdY9tb2RlbHMucHk=) | `92.29% <0%> (-0.05%)` | :arrow\_down: | -----  
[Continue to review full report at Codecov](https://codecov.io/gh/apache/incubator-airflow/pull/4253?src=pr&el=continue). > **\*\*Legend\*\*** - [Click here to learn more](https://docs.codecov.io/docs/codecov-delta) > ` $\Delta$  = absolute <relative> (impact)`, ` $\emptyset$  = not affected`, ` $?$  = missing data` > Powered by  
[Codecov](https://codecov.io/gh/apache/incubator-airflow/pull/4253?src=pr&el=footer). Last update  
[d1d612e...eccc75a](https://codecov.io/gh/apache/incubator-airflow/pull/4253?src=pr&el=lastupdated).  
Read the [comment docs](https://docs.codecov.io/docs/pull-request-comments).
2. @Fokko @ashb @feng-tao @XD-DENG @kaxil @saguziel @aoen @YingboWang PTAL
3. @KevinYang21, I am oncall this week. will take a look next week if this pr is still open
4. @feng-tao lol I'm oncall too :D Don't worry about it, we got plenty of committers ;) Ty for being caring.
5. @kaxil Thank you for reviewing! Replied the comments in their own thread.
6. Looks sensible to me. @feng-tao @ashb any comments?

## github\_pulls\_reviews:

1. Thanks @KevinYang21 ! Is there any specific reason that we need to return `logging\_config`? -----  
----- **\*\*Before this PR\*\***: `configure\_logging()` is invoked in `airflow/settings.py`, but seems what it returns is not used (its return value is not passed to any variable actually, https://github.com/apache/incubator-airflow/pull/4253/files#diff-535b21d0d7b5e33649e0eb521b41f157L264). **\*\*After this PR\*\***: `logging\_class\_path` is returned by `configure\_logging()`, then used in `airflow/utis/dag\_processing.py`. But `logging\_config` is still not used anywhere. ----- So I'm thinking if we really need to include `logging\_config` in the return values? Kindly let me know if I missed anything (quite likely). Thanks!
2. @XD-DENG Thank you for reviewing. I actually don't think the return value is used anywhere. I was playing it safe by keep the original return value. After you mentioned, I think we can just remove that return value--we can always add it back if it is needed in the future.
3. Do we need `[0]` here ?? `rsplit('.',1)` would just have one item right??
4. Do we need to assign this to a variable ??



5. It will actually [return a list as split](https://docs.python.org/2/library/string.html#string.rsplit). If not it will fail the newly added unit test :D
6. Ya the logging\_class\_path is used [here](https://github.com/apache/incubator-airflow/pull/4253/files#diff-4fd65ce02db58eed692eef6e01d05R546). Just trying to keep it consistent between two usage.
7. **body:** Oh yes, my bad 😊  
**label:** code-design
8. 🍏

#### jira\_issues:

1. **summary:** reload\_module not working with custom logging class  
**description:** If using custom logging class, the reload\_module in dag\_processing.py will fail because it will try to reload default logging class, which is not loaded at the first place.
2. **summary:** reload\_module not working with custom logging class  
**description:** If using custom logging class, the reload\_module in dag\_processing.py will fail because it will try to reload default logging class, which is not loaded at the first place.

#### jira\_issues\_comments:

1. KevinYang21 opened a new pull request #4253: [WIP][AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent URL: https://github.com/apache/incubator-airflow/pull/4253 Make sure you have checked \_all\_ steps below. ### Jira - [ ] My PR addresses the following [Airflow Jira] (https://issues.apache.org/jira/browse/AIRFLOW/) issues and references them in the PR title. For example, "[AIRFLOW-XXX] My Airflow PR" - https://issues.apache.org/jira/browse/AIRFLOW-XXX - In case you are fixing a typo in the documentation you can prepend your commit with [AIRFLOW-XXX], code changes always need a Jira issue. ### Description - [ ] Here are some details about my PR, including screenshots of any UI changes: ### Tests - [ ] My PR adds the following unit tests \_\_OR\_\_ does not need testing for this extremely good reason: ### Commits - [ ] My commits all reference Jira issues in their subject lines, and I have squashed multiple commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good git commit message]" (http://chris.beams.io/posts/git-commit/): 1. Subject is separated from body by a blank line 1. Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1. Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what" and "why", not "how" ### Documentation - [ ] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. ### Code Quality - [ ] Passes `flake8` ----- This is an automated message from the Apache Git Service. To respond to the message, please log on 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
2. kaxil closed pull request #4253: [AIRFLOW-3414] Fix reload\_module in DagFileProcessorAgent URL: https://github.com/apache/incubator-airflow/pull/4253 This is a PR merged from a forked repository. As GitHub hides the original diff on merge, it is displayed below for the sake of provenance: As this is a foreign pull request (from a fork), the diff is supplied below (as it won't show otherwise due to GitHub magic): diff --git a/airflow/logging\_config.py b/airflow/logging\_config.py index 84850b4f82..4865f0567a 100644 --- a/airflow/logging\_config.py +++ b/airflow/logging\_config.py @@ -57,9 +57,9 @@ def configure\_logging(): .format(logging\_class\_path, err) ) else: - from airflow.config\_templates.airflow\_local\_settings import ( - DEFAULT\_LOGGING\_CONFIG as logging\_config - ) + logging\_class\_path = 'airflow.config\_templates.' + 'airflow\_local\_settings.DEFAULT\_LOGGING\_CONFIG' + logging\_config = import\_string(logging\_class\_path) log.debug('Unable to load custom logging, using default config instead') try: @@ -73,7 +73,7 @@ def configure\_logging(): validate\_logging\_config(logging\_config) - return logging\_config + return logging\_class\_path def validate\_logging\_config(logging\_config): diff --git a/airflow/settings.py b/airflow/settings.py index 8f8420ea22..8691fe4e75 100644 --- a/airflow/settings.py +++ b/airflow/settings.py @@ -261,7 +261,7 @@ def configure\_action\_logging(): except Exception: pass -configure\_logging() +logging\_class\_path = configure\_logging() configure\_vars() configure\_adapters() # The webserver imports this file from models.py with the default settings. diff --git a/airflow/utils/dag\_processing.py b/airflow/utils/dag\_processing.py index 0fe9e917ef..6425f77745 100644 --- a/airflow/utils/dag\_processing.py +++ b/airflow/utils/dag\_processing.py @@ -34,6 +34,7 @@ from collections import defaultdict from collections import namedtuple from datetime import



```

timedelta +from importlib import import_module import psutil from six.moves import range,
reload_module @@ -45,6 +46,7 @@ from airflow import configuration as conf from
airflow.dag.base_dag import BaseDag, BaseDagBag from airflow.exceptions import AirflowException
+from airflow.settings import logging_class_path from airflow.utils import timezone from airflow.utils.db
import provide_session from airflow.utils.log.logging_mixin import LoggingMixin @@ -539,7 +541,9
@@ def helper(): # e.g. RotatingFileHandler. And it can cause connection corruption if we # do not
recreate the SQLA connection pool. os.environ['CONFIG_PROCESSOR_MANAGER_LOGGER'] =
'True' - reload_module(airflow.config_templates.airflow_local_settings) + # Replicating the behavior of
how logging module was loaded + # in logging_config.py +
reload_module(import_module(logging_class_path.rsplit('.', 1)[0])) reload_module(airflow.settings) del
os.environ['CONFIG_PROCESSOR_MANAGER_LOGGER'] processor_manager =
DagFileProcessorManager(dag_directory, diff --git a/tests/utils/test_dag_processing.py
b/tests/utils/test_dag_processing.py index 4043caec56..9c85b66c57 100644 ---
a/tests/utils/test_dag_processing.py +++ b/tests/utils/test_dag_processing.py @@ -18,12 +18,15 @@ #
under the License. import os +import sys +import tempfile import unittest from datetime import timedelta
from mock import MagicMock from airflow import configuration as conf +from airflow.configuration
import mkdir_p from airflow.jobs import DagFileProcessor from airflow.jobs import LocalTaskJob as LJ
from airflow.models import DagBag, TaskInstance as TI @@ -38,6 +41,96 @@ DEFAULT_DATE =
timezone.datetime(2016, 1, 1) +SETTINGS_FILE_VALID = "" +LOGGING_CONFIG = { + 'version':
1, + 'disable_existing_loggers': False, + 'formatters': { + 'airflow.task': { + 'format': '[%(asctime)s]
{%(filename)s:%(lineno)d} %(levelname)s - %(message)s' + }, + }, + 'handlers': { + 'console':
{ + 'class': 'logging.StreamHandler', + 'formatter': 'airflow.task', + 'stream': 'ext://sys.stdout' + }, + 'task': {
+ 'class': 'logging.StreamHandler', + 'formatter': 'airflow.task', + 'stream': 'ext://sys.stdout' + }, + }, +
'loggers': { + 'airflow': { + 'handlers': ['console'], + 'level': 'INFO', + 'propagate': False + }, + 'airflow.task':
{ + 'handlers': ['task'], + 'level': 'INFO', + 'propagate': False, + }, + } + } + "" +
+SETTINGS_DEFAULT_NAME = 'custom_airflow_local_settings' + +class settings_context(object):
+ "" + Sets a settings file and puts it in the Python classpath + + :param content: + The content of the
settings file + "" + + def __init__(self, content, dir=None, name='LOGGING_CONFIG'): + self.content
= content + self.settings_root = tempfile.mkdtemp() + filename = "
{}.py".format(SETTINGS_DEFAULT_NAME) + + if dir: + # Replace slashes by dots + self.module =
dir.replace('/', '.') + '.' + SETTINGS_DEFAULT_NAME + '.' + name + + # Create the directory structure +
dir_path = os.path.join(self.settings_root, dir) + mkdir_p(dir_path) + + # Add the __init__ for the
directories + # This is required for Python 2.7 + basedir = self.settings_root + for part in dir.split('/'):
+ open(os.path.join(basedir, '__init__.py'), 'w').close() + basedir = os.path.join(basedir, part) +
open(os.path.join(basedir, '__init__.py'), 'w').close() + + self.settings_file = os.path.join(dir_path,
filename) + else: + self.module = SETTINGS_DEFAULT_NAME + '.' + name + self.settings_file =
os.path.join(self.settings_root, filename) + + def __enter__(self): + with open(self.settings_file, 'w') as
handle: + handle.writelines(self.content) + sys.path.append(self.settings_root) + conf.set( + 'core', +
'logging_config_class', + self.module + ) + return self.settings_file + + def __exit__(self, *exc_info): + #
shutil.rmtree(self.settings_root) + # Reset config + conf.set('core', 'logging_config_class', "") +
sys.path.remove(self.settings_root) + class TestDagFileProcessorManager(unittest.TestCase): def
test_set_file_paths_when_processor_file_path_not_in_new_file_paths(self): @@ -123,6 +216,54 @@
def test_find_zombies(self): class TestDagFileProcessorAgent(unittest.TestCase): + def
test_reload_module(self): + "" + Configure the context to have core.logging_config_class set to a fake
logging + class path, thus when reloading logging module the airflow.processor_manager + logger should
not be configured. + "" + with settings_context(SETTINGS_FILE_VALID): + # Launch a process
through DagFileProcessorAgent, which will try + # reload the logging module. + def
processor_factory(file_path, zombies): + return DagFileProcessor(file_path, + False, + [], + zombies) + +
test_dag_path = os.path.join(TEST_DAG_FOLDER, 'test_scheduler_dags.py') + async_mode = 'sqlite'
not in conf.get('core', 'sql_alchemy_conn') + + log_file_loc = conf.get('core',
'DAG_PROCESSOR_MANAGER_LOG_LOCATION') + try: + os.remove(log_file_loc) + except
OSError: + pass + + # Starting dag processing with 0 max_runs to avoid redundant operations. +
processor_agent = DagFileProcessorAgent(test_dag_path, + [], + 0, + processor_factory, + async_mode)
+ manager_process = \ + processor_agent._launch_process(processor_agent._dag_directory, +
processor_agent._file_paths, + processor_agent._max_runs, + processor_agent._processor_factory, +
processor_agent._child_signal_conn, + processor_agent._stat_queue, + processor_agent._result_queue, +
processor_agent._async_mode) + if not async_mode: + processor_agent.heartbeat() + +
manager_process.join() + + # Since we are reloading logging config not creating this file, + # we should
expect it to be nonexistent. + self.assertFalse(os.path.isfile(log_file_loc)) + def test_parse_once(self): def

```

```
processor_factory(file_path, zombies): return DagFileProcessor(file_path, @@ -164,7 +305,7 @@ def
processor_factory(file_path, zombies): except OSError: pass - # Starting dag processing with 0 max_runs
to avoid redundant operations. + # Starting dag processing with 0 max_runs to avoid redundant
operations. processor_agent = DagFileProcessorAgent(test_dag_path, [], 0, -----
----- This is an automated message from the Apache Git Service. To respond to the
message, please log on 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
```