

Airflow Improvement Proposals

The purpose of an Airflow Improvement Proposal (AIP) is to introduce any major change to Apache Airflow. This is required in order to balance the need to support new features, use cases, while avoiding accidentally introducing half thought-out interfaces that cause needless problems when changed.

What is considered a major change that needs an AIP?

Any of the following should be considered a major change:

- Any major new feature, subsystem, or piece of functionality
- Any change that impacts the public interfaces of the project

What are the "public interfaces" of the project? All of the following are public interfaces that people build around:

- Airflow DAG DSL components: DAG, XCom, TaskInstance and context.
- Web UI
- Core operators (BaseOperator, SubDagOperator, SensorOperator)
- [Airflow 2.0] REST endpoints
- Command line tools and arguments

Proposal lifecycle

Discuss

Most proposals start with an idea. If you have an idea of how Airflow could improve, we encourage you to send an email to dev@airflow.apache.org with a subject starting with *[DISCUSS]*. This email thread will allow you to gather early feedback. We encourage you to start a *Draft* document or share a Google Document with the mailing list detailing your proposal. Use AIP template here to file details on your proposal.

Draft

Anyone is welcome to propose a new Improvement to Airflow. You will need to request edit access to the Confluence page in order to create your AIP document. You can do so by sending an email with your Wiki ID to dev@airflow.apache.org and request permission.

[Create new AIP](#)

Vote

Once you or someone else feels like there's a rough consensus on the idea and there's no strong opposition, you can move your proposal to the Vote phase. For this you will send a new email to the dev list with the subject starting with *[VOTE]*. This phase follows a vote on code modifications for Apache projects as described [here](#). Remember to link the discussion thread to your AIP and ideally also to this vote email. And don't forget to link the AIP to the email as well.

The voting has to last at least 72 hours and it's a good practice to extend this time if most of this period is a weekend. Whole community is encouraged to give +1 and -1 votes. There's also fractional votes (0.5, -0.99) that allow you to express your view on the discussion without giving an explicit approval /block signal. However, only Airflow committers and PMC members have a binding vote. This means that any vote requires at least 3, +1 binding votes to be recognized as passed. In addition, any -1 binding vote will move the proposal back to the Discuss phase in order to address changes.

The vote is a consensus-seeking process, "the goal is not to *win* votes or come to a unanimous agreement, but rather to ensure that there's a forum for people to raise and discuss their concerns, and that nobody feels strongly enough to block the group from moving forward. Consensus-seeking emphasizes discussion over enumeration: *Rough consensus is achieved when all issues are addressed, but not necessarily accommodated.*" (quote from [Working in Public: The Making and Maintenance of Open Source Software](#) by [Nadia Eghbal](#))

If you want to introduce significant changes to your proposal after it has been accepted you can do so following the same procedure.

Accepted

Congratulations! Your proposal has been accepted. Next up is breaking down tasks. You should break down tasks on [Apache Airflow GitHub issues page](#). You can optionally group the tasks together in a [Github Project](#) if it consists of multiple steps and you want to be able to track progress for its implementation or share the progress with others. Once issues are created, make sure to send them to the vote thread (optionally [Airflow slack](#)). Ask a committer to create a label for your AIP and link all your issues with it for easier searchability.

Now you can start contributing and - more importantly - you can, and probably should, encourage others to contribute to your project. However this is optional and up to your discretion. You can learn how to contribute and communicate by reading the [Contributing guide](#) in the GitHub repository. Use your imagination and various communication channels for ways to encourage people if you do not already have contributors following your idea. It is advisable that several people work on the AIP so that the knowledge is shared.

Completed

You should move your AIP to complete after you consider the main bulk of work has been merged to the repository. It is okay to leave open issues for minor follow up tasks like adding additional capabilities or similar.

Abandoned

AIP may be moved to abandoned if there has been no owner or intention of developing it further for over a year or if the owner decides to abandon it. The latter can happen if the creator decides to refactor AIP into a new AIP for example or if there is a proposal that supersedes the original AIP.

AIPs under discussion

Title	Discussion Thread	Created
[WIP] AIP-55 Rule-based timetable with logical composition		
[WIP] AIP-66: Execution of specific DAG versions		
[WIP] AIP-70 Migrating to asynchronous programming	https://lists.apache.org/thread/bhz76jzsb8thwy89sz77fcywhgkj63sw	
AIP-5 Remote DAG Fetcher	https://lists.apache.org/thread.html/224d1e7d1b11e0b8314075f21b1b81708749f2899f4cce5af295e8a8@%3Cdev.airflow.apache.org%3E	
AIP-14 Create composable operators using common interfaces		<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-20 DAG manifest		<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-22: Group ORM models by their logical usage instead of type	https://lists.apache.org/thread.html/fc21e13413a4a11b2c490b0e2f59dd83239711433599fe8693b98d02@%3Cdev.airflow.apache.org%3E	<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-35 Add Signal Based Scheduling To Airflow	[DISCUSS][AIP-35] Add Signal Based Scheduling To Airflow	<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-36 DAG Versioning		<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-45 Remove double dag parsing in airflow run	https://lists.apache.org/thread/tc37cgrx2tojv3zgzk06f5ypk0y2hg	<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-46 Runtime isolation for airflow tasks and dag parsing	https://lists.apache.org/thread/9kw13cxcg06p37shv57hsomnx6zognoc	<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-54 Multi tenancy - View Level Access Control		
AIP-56 Extensible user management	https://lists.apache.org/thread/ck8dsj5w82lvr0cpwr4wiptmydqwnsqc	
AIP-57 Refactor SLA Feature	https://lists.apache.org/thread/kn1rb9l70jk16j83fycp68l2g70wnhzi https://lists.apache.org/thread/kj33fwd9otbbm52my9j3y39rp547g0n5 https://lists.apache.org/thread/l2ho7w18d11vcnon7dxoxghzxhtjw5hf (Multiple threads were created on the topic)	
AIP-59 Performance tests framework		
AIP-60 Standard URI representation for Airflow Datasets	https://lists.apache.org/thread/rf6c80ljjkml0l15h2jys7k713q3os1d	

AIP-61 Hybrid Execution	https://lists.apache.org/thread/94sg7l4m3qjk4b3vfq3lr94oc5f59q4j
AIP-62 Getting Lineage from Hook Instrumentation	https://lists.apache.org/thread/5chxcp0zjcx66d3vs4qlrm8kl6l4s3m2
AIP-63: DAG Versioning	https://lists.apache.org/thread/vvm43tfchyo92hmf40fqvmq0f5845bjr
AIP-64: Keep TaskInstance try history	https://lists.apache.org/thread/vvm43tfchyo92hmf40fqvmq0f5845bjr
AIP-65: Improve DAG history in UI	https://lists.apache.org/thread/vvm43tfchyo92hmf40fqvmq0f5845bjr
AIP-67 Multi-team deployment of Airflow components	https://lists.apache.org/thread/6k3sxxvboko4fw7qdh78y3qhf26vh3q
AIP-68 Extended Plugin Interface for Custom Grid View Panels	https://lists.apache.org/thread/8hlltm9brdxqyf8jyw1syrfb11hl52k5
AIP-69 Remote Executor	https://lists.apache.org/thread/8hlltm9brdxqyf8jyw1syrfb11hl52k5

Accepted AIPs

AIPs which have been voted on and accepted for inclusion in Airflow, but not yet (fully) implemented.

Title	Discussion Thread	Project	InRelease	Created
AIP-26 Production-ready Airflow Docker Image	AIP-26 Discussion			
AIP-38 Modern Web Application	[DISCUSS][AIP-38 Modern Web Application]	AIP-38: Modern Web Application		<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-44 Airflow Internal API	https://lists.apache.org/thread/nsmo339m618kjsdkwq83z8omrt08zh3			<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-47 New design of Airflow System Tests	https://lists.apache.org/thread/htd4013yn483qfhwv11vc26jpf2yvjp			<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-49 OpenTelemetry Support for Apache Airflow	[DISCUSSION] AIP-49 OpenTelemetry Support for Apache Airflow			<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-50 Trigger DAG UI Extension with Flexible User Form Concept	https://lists.apache.org/thread/kxkctcbh9drfw065dgvr673zl0xyfl3r			<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-51 Removing Executor Coupling from Core Airflow	https://lists.apache.org/thread/yd4tcgyqft664dt8clgx48gyvzfddnjh https://github.com/apache/airflow/discussions/27241	https://github.com/orgs/apache/projects/162		<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>

Completed AIPs

Title	Discussion Thread	In Release	Created
AIP-3 Drop support for Python 2	[DISCUSS] AIP-3: Drop Python2 support in Airflow 2.0	2.0.0	<code>\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>

AIP-6 Enforce the usage of a code formatter & linter	AIP-6: Enforce the usage of a code formatter	2.0.0	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-7 Simplified development workflow	AIP-7 Simplified development workflow		<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-8 Split Providers into Separate Packages for Airflow 2.0	AIP-8 Split Hooks/Operators into Separate Packages Voting discussion.	2.0.0	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-9 Automated Dependency Management	AIP-9 Automated Dependency Management	2.0.0	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-10 Multi-layered and multi-stage official Airflow CI image	Muti-layered official image for Airflow	1.10.4	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-11 Create a Landing Page for Apache Airflow	[PROPOSAL] Add a landing page for Apache Airflow	1.10.7	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-15 Support Multiple-Schedulers for HA & Better Scheduling Performance	Multiple Schedulers - "scheduler_lock" A Naive Multi-Scheduler Architecture Experiment of Airflow [PROPOSAL][AIP-15 Support Multiple-Schedulers for HA & Better Scheduling Performance]	2.0.0	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-16: Use nested commands in CLI	[VOTE] AIP-16: CLI: Use nested commands instead of flags CLI: Use nested commands instead of flags	2.0.0	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-17: Consolidate and de-duplicate sensor tasks in airflow Smart Sensor	[Discuss] Airflow sensor optimization	2.0.0	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-21: Changes in import paths	[VOTE] Changes in import paths	2.0.0	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-23 Migrate out of Travis CI	Discussion Thread	1.10.10	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-24 DAG Persistence in DB using JSON for Airflow Webserver and (optional) Scheduler	Airflow DAG Serialisation	1.10.7	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-27 Migrate to pytest	[PROPOSAL] Migrate to Pytest	1.10.7	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-31: "TaskFlow API" for clearer/simpler DAG definition	[DISCUSS] Airflow functional DAGs Slack thread	2.0.0	2020-02-21
AIP-32: Airflow REST API	[PROPOSAL][AIP-32] Airflow REST API	2.0.0	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-33 Secrets backend	[DISCUSS] AIP-33 secrets backend	1.10.10	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-34 TaskGroup: A UI task grouping concept as an alternative to SubDagOperator	Pony Mail	2.0.0	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>

AIP-39 Richer scheduler_interval	Scoping out a new feature for 2.1: improving schedule_interval [DISCUSS][AIP-39] Richer (and pluggable) schedule_interval on DAGs	2.2.0	\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-40: Deferrable ("Async") Operators	[DISCUSS][AIP-40] Deferrable ("Async") Operators	2.2.0	\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-42 Dynamic Task Mapping	https://lists.apache.org/thread/xhq5o3b3h879cWSC93xwq0836mwc5k6p	2.3.0	\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-43 DAG Processor separation	https://lists.apache.org/thread/rgovdphjxtcqfmcfdrbfntwdr2nj71	2.4.0	\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-43 DAG Processor separation	https://lists.apache.org/thread/rgovdphjxtcqfmcfdrbfntwdr2nj71		\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-48 Data Dependency Management and Data Driven Scheduling	https://lists.apache.org/thread/vnjv32zv3fq92t5nqf0qqy03xphwm8	2.4.0	\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-53 OpenLineage in Airflow	https://lists.apache.org/thread/2brvI4ynkxcff86zIokkb47wb5gx8hw7	2.7.0	\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())

"Umbrella" AIPs

AIPs that are too big to be treated as "deliverable" AIPs on its own - they are split into smaller AIPs.

Title	Discussion Thread	In Release	Created
AIP-1: Improve Airflow Security [Deps: AIP-43]			\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-4 Automation of System Tests [Deps: AIP-47]	Support for integration tests in Airflow		\$action.dateFormatter.formatGivenString("yyyy-MM-dd", \$content.getCreationDate())

Abandoned AIPs

Title	State	Discussion Thread	Created
AIP-2 Simplify process launching	Draft		\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-12 Persist DAG into DB	Abandoned in favour of AIP-24	[DISCUSS] AIP-12 Persist DAG into DB	\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-13: OpenAPI 3 based API definition	Abandoned in favour of AIP-32		\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-18 Persist all information from DAG file in DB	Abandoned in favour of AIP-24	TBD	\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())
AIP-19 Making the webserver stateless	Abandoned in favour of AIP-24		\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())

AIP-25 The Knative Executor	Abandoned		<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-28: Add AsyncExecutor option	Abandoned in favour of AIP-40: Deferrable ("Async") Operators	apache-airflow.slack.com - sig-async-operators	<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
AIP-29: AWS Fargate Executor	Abandoned in favour of AIP-51		
AIP-37 Virtualenv management inside Airflow	Draft		<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>
Deprecated- AIP-5 DagFetcher	Abandoned in favour of new https://cwiki.apache.org/confluence/display/AIRFLOW/AIP-5+Remote+DAG+Fetcher		<code>\$action.dateFormatter. formatGivenString("yyyy-MM-dd", \$content.getCreationDate())</code>

(The above list is automatically generated based on page labels. Do not edit manually)