Hortonworks Data Platform

Workflow Management

(October 30, 2017)

docs.hortonworks.com

Hortonworks Data Platform: Workflow Management

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, ZooKeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, training and partner-enablement services. All of our technology is, and will remain, free and open source.

Please visit the Hortonworks Data Platform page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. You can contact us directly to discuss your specific needs.

Table of Contents

List of Figures

List of Tables

1. Workflow Manager Basics

Workflow Manager, which can be accessed as a View in Ambari, allows you to easily create and schedule workflows and monitor workflow jobs. It is based on the Apache Oozie workflow engine that allows users to connect and automate the execution of big data processing tasks into a defined workflow. Workflow Manager integrates with the Hortonworks Data Platform (HDP) and supports Hadoop jobs for Hive, Sqoop, Pig, MapReduce, Spark, and more. In addition, it can be used to perform Java, Linux shell, distcp, SSH, email, and other operations.

The following content describes the *design* component and the *dashboard* component that are included with Workflow Manager.

Workflow Manager Design Component [1]

Workflow Manager Dashboard Component [2]

Workflow Action Types [3]

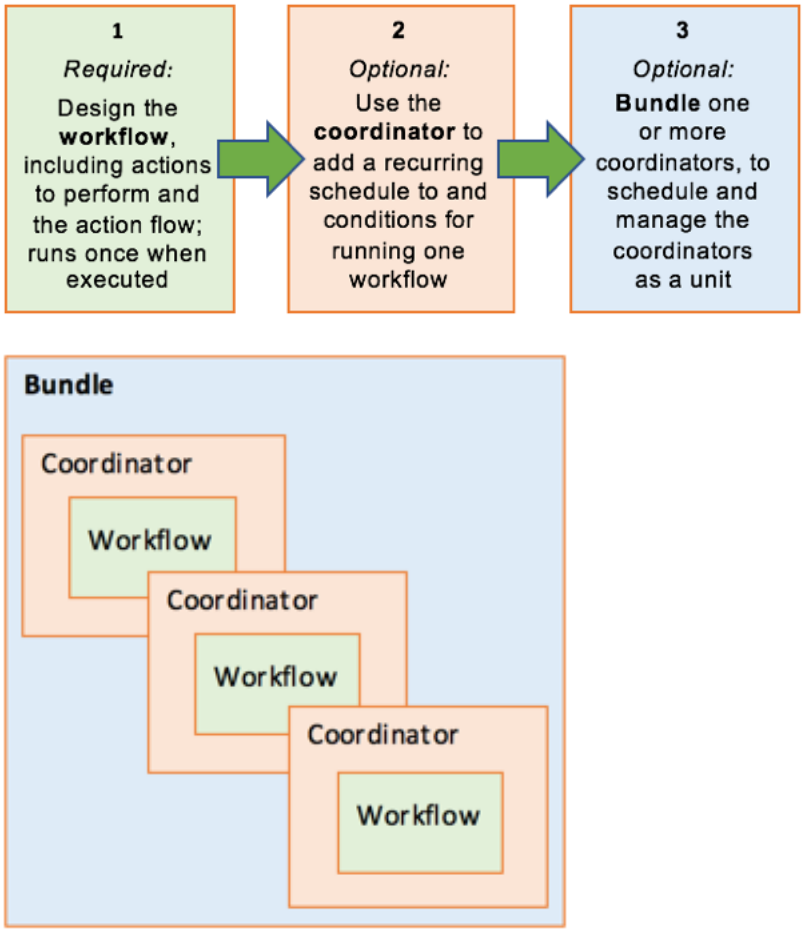
1.1. Workflow Manager Design Component

You can use the design component of Workflow Manager to create and graphically lay out your workflow. You can select action nodes and control nodes and implement them in a defined sequence to ingest, transform, and analyze data. *Action nodes* trigger a computation or processing task. *Control nodes* control the workflow execution path. You can add Oozie actions and other operations as action nodes on the workflow, and you can add fork and decision points as control nodes to your workflow. You can include a maximum of 400 nodes in a workflow.

From the designer, you can also schedule workflows using coordinators, and group multiple coordinators into bundles. A *coordinator* enables you to automate the recurring execution of a workflow based on time, an external event, or the availability of specific data files. The coordinator can define parameters only around a single workflow. Coordinators make it easy to manage concurrency and keep track of the completion of each workflow instance. The coordinator controls how many instances of a workflow can execute in parallel, as well as controlling how many instances are brought into a waiting state before the next workflow instance can run.

Coordinator jobs can be combined to create a chain of jobs that can be managed as a single entity. This logical chaining together of multiple, interdependent coordinator jobs is referred to as a *bundle*, also called a data application pipeline. By using bundled coordinator jobs, the output of one workflow that runs at one frequency can be used as the initial input for another job that runs at a different frequency. You can even bundle the same workflow with different coordinator schedules. To create a bundle, you must configure one or more workflows as coordinator jobs, then configure the coordinator jobs as part of a bundle.

Figure 1.1. Relationship of Workflows, Coordinators, and Bundles



More Information

You can read further details about workflows, coordinators, and bundles in the following Apache documentation:

- Apache Oozie Workflow Functional Specification
- Apache Oozie Coordinator Functional Specification
- Apache Oozie Bundle Functional Specification

1.2. Workflow Manager Dashboard Component

The dashboard provides monitoring capabilities for jobs submitted to Workflow Manager. There are three types of jobs in Workflow Manager: *Workflow* jobs, *coordinator* jobs, and *bundle* jobs. Each job type is viewed in a separate table in the dashboard.

You can perform actions such as run, kill, suspend, and resume on individual jobs, or you can select multiple jobs of a single type and perform these actions on several jobs at once.

From the dashboard, you can access details about a job, view job logs, and view the XML structure of the job.

Workflow jobs are launched from a client and run on the server engine. The Oozie metadata database contains the workflow definitions, variables, state information, and other data required to run the workflow jobs.

1.3. Workflow Action Types

Each action in a workflow represents a job to be run. For example, a Hive action runs a Hive job. Most actions in a workflow are executed asynchronously, so they have to wait until the previous action completes. However, file system operations on HDFS are executed synchronously.

Identical workflows can be run concurrently when properly parameterized. The workflow engine can detect completion of computation and processing actions. Each action is given a unique callback URL. The action invokes the URL when its task is complete. The workflow engine also has a mechanism to poll actions for completion, for cases in which the action fails to invoke the callback URL or the type of task cannot invoke the callback URL upon completion.

Workflow action types

Hive Action	Used for asynchronously executing Hive and Hive2 scripts and Sqoop jobs. The workflow job waits until the Hive job completes before continuing to the next action.	To run the Hive job, you have to configure the Hive action with the `resourceManager`, `nameNode`, and `Hive script` elements, as well as other necessary parameters and configuration.
Hive2 Action	The Hive2 action runs Beeline to connect to Hive Server 2. The workflow job will wait until the Hive Server 2 job completes before continuing to the next action.	To run the Hive Server 2 job, you have to configure the Hive2 action with the `resourceManager`, `nameNode`, `jdbc-url`, and `password` elements, and either `Hive script` or `query` elements, as well as other necessary parameters and configuration.
Sqoop Action	The workflow job waits until the Sqoop job completes before continuing to the next action.	The Sqoop action requires Apache Hadoop 0.23. To run the Sqoop job, you have to configure the Sqoop action with the `resourceManager`, `nameNode`, and `Sqoop command` or `arg` elements, as well as configuration.
Pig Action	The workflow job waits until the Pig job completes before continuing to the next action.	The Pig action has to be configured with the `resourceManager`, `nameNode`, Pig script, and other necessary parameters and configuration to run the Pig job.
Sub-workflow (Sub-wf) Action	The sub-workflow action runs a child workflow job. The parent workflow job waits until the child workflow job has completed.	The child workflow job can be in the same Oozie system or in another Oozie system.
Java Action	Java applications are executed in the Hadoop cluster as map-reduce jobs with a single Mapper task. The workflow job waits until the Java application completes its execution before continuing to the next action.	The Java action has to be configured with the `resourceManager`, `nameNode`, main Java class, JVM options, and arguments.
Shell Action	Shell commands must complete before going to the next action.	The standard output of the shell command can be used to make decisions.
DistCp (distcp) Action	The DistCp action uses Hadoop "distributed copy" to copy files from one cluster to another or within the same cluster.	Both Hadoop clusters have to be configured with proxyuser for the Oozie process. **IMPORTANT:** The DistCp action may not work properly with all configurations (secure, insecure) in all versions of Hadoop.

MapReduce (MR) Action	The workflow job waits until the Hadoop MapReduce job completes before continuing to the next action in the workflow.	
SSH Action	Runs a remote secure shell command on a remote machine. The workflow waits for the SSH command to complete.	SSH commands are executed in the home directory of the defined user on the remote host. **Important:** SSH actions are deprecated in Oozie schema 0.1, and removed in Oozie schema 0.2.
Spark Action	The workflow job waits until the Spark job completes before continuing to the next action.	To run the Spark job, you have to configure the Spark action with the `resourceManager`, `nameNode`, and Spark master elements, as well as other necessary elements, arguments, and configuration. **Important:** The `yarn-client` execution mode for the Oozie Spark action is no longer supported and has been removed. Workflow Manager and Oozie continue to support `yarn-cluster` mode.
Email Actions	Email jobs are sent synchronously.	An email must contain an address, a subject and a body.
HDFS (FS) Action	Allows manipulation of files and directories in HDFS. File system operations are executed synchronously from within the FS action, but asynchronously within the overall workflow.	
Custom Action	Allows you to create a customized action by entering the appropriate XML.	Ensure that the JAR containing the Java code and the XML schema definition (XSD) for the custom action have been deployed. The XSD must also be specified in the Oozie configuration.

More Information

For more information about actions, see the following Apache documentation:

- Apache Oozie Workflow Functional Specification
- Apache Oozie Hive Action Extension
- Apache Oozie Hive2 Action Extension
- Apache Oozie Sqoop Action Extension
- Apache Oozie DistCp Action Extension
- Apache Oozie Spark Action Extension

2. Content Roadmap for Workflow Manager

The following resources provide additional information that can be useful when using Workflow Manager and associated Apache components.

Table 2.1. Workflow Manager Content roadmap

Task	Resources	Source	Description
Understanding	Apache Oozie Specification	Apache wiki	Apache Oozie is the engine behind Workflow Manager. Understanding how Oozie works can help you understand how to use Workflow Manager.
Installing and Upgrading	Ambari Automated Install Guide	Hortonworks	Ambari provides an end-to-end management and monitoring solution for your HDP cluster. Using the Ambari Web UI and REST APIs, you can deploy, operate, manage configuration changes, and monitor services for all nodes in your cluster from a central point.
	Non-Ambari Cluster Installation Guide	Hortonworks	Describes the information and materials you need to get ready to install the Hortonworks Data Platform (HDP) manually.
	Ambari Upgrade Guide	Hortonworks	Ambari and the HDP Stack being managed by Ambari can be upgraded independently. This guide provides information on: Getting ready to upgrade Ambari and HDP, Upgrading Ambari, and Upgrading HDP.
	Non-Ambari Cluster Upgrade Guide	Hortonworks	These instructions cover the upgrade between two minor releases. If you need to upgrade between two maintenance releases, follow the upgrade instructions in the HDP Release Notes.
Administering	Configuring Workflow Manager View	Hortonworks	Describes how to set up and create the Workflow Manager instance in Ambari.
Developing	Apache Ambari Workflow Manager View for Apache Oozie	Hortonworks	Hortonworks Community Connection (HCC) articles that provide an example of each of the node actions available in Workflow Manager.
	Apache Hive Action	Apache wiki	Provides information about the Hive action extension that can be helpful when configuring a Hive node in WFM.
	Apache Hive2 Action	Apache wiki	Provides information about the Hive2 action extension that can be helpful when configuring a Hive2 node in WFM.
	Apache Sqoop Action	Apache wiki	Provides information about the Sqoop action extension that can be helpful when configuring a Sqoop node in WFM.
	Apache Shell Action	Apache wiki	Provides information about the Shell action extension that can be helpful when configuring a Shell node in WFM.
	Apache Spark Action	Apache wiki	Provides information about the Spark action extension that can be helpful when configuring a Spark node in WFM.

Task	Resources	Source	Description
	Apache MapReduce, Pig, File System (FS), SSH, Sub-Workflow, and Java Actions	Apache wiki	Provides information about the actions initially included with Oozie; can be helpful when configuring these node types in WFM.
	Apache Email Action	Apache wiki	Provides information about the emal action extension that can be helpful when configuring a email node in WFM.
	Apache Custom Actions	Apache wiki	Provides information about the custom action extension that can be helpful when configuring a custom node in WFM.
	Apache Java Cookbook	Apache wiki	Provides information about the Java action that can be helpful when configuring a Java node in WFM.
Security	Hadoop Security Guide	Hortonworks	Provides details of the security features implemented in the Hortonworks Data Platform (HDP).
High Availability	Apache Hadoop High Availability	Hortonworks	Provides details for system administrators who need to configure the cluster for High Availability.
Other resources	Hortonworks Community Connection (HCC)	Hortonworks	Provides access to a community of big data users, developers, data scientists, and so forth.

3. Quick Start

For an overview of the features and functionality available as part of Workflow Manager, see the following content:

3.1. Configuring a Workflow Manager Ambari View

You access Workflow Manager (WFM) as an Ambari View. You create an instance of the WFM View from the Ambari web UI. You can create multiple instances of a View and you can assign different users to different Views.

Instructions for creating a WFM View are included in the *Apache Ambari Views* guide:

- Configuring Your Cluster for Workflow Manager View
- If using Kerberos: Kerberos Setup for Workflow Manager
- Creating and Configuring a Workflow Manager View Instance

More Information

Configuring Views for Kerberos in the *Apache Ambari Views* guide

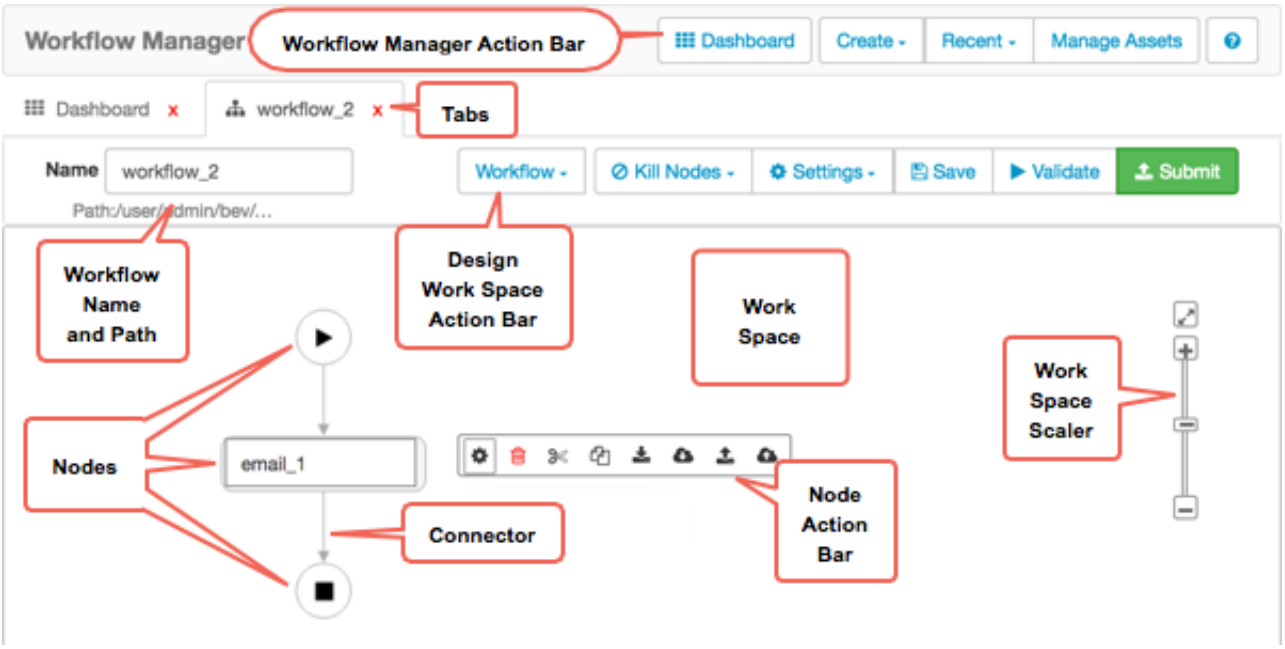
Managing Cluster Roles in the *Apache Ambari Administration* guide

3.2. UI Elements of Workflow Manager

There are two components that make up Workflow Manager: The designer and the dashboard.

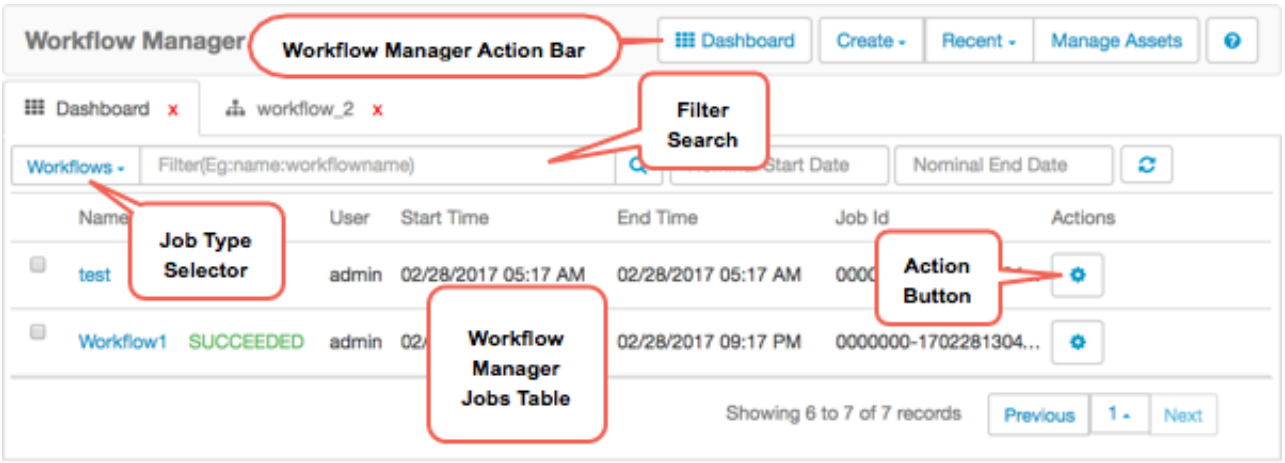
Below are the primary elements of the design component.

Figure 3.1. Workflow Manager Design Component



Below are the primary elements of the dashboard.

Figure 3.2. Workflow Manager Dashboard Component



3.3. Understanding the Design Component

You can create and run workflows, coordinators, and bundles from Workflow Manager.

You create a workflow in Workflow Manager using a graphing flow tool. The type of graph you create is known as a directed acyclic graph (DAG). This type of graph has a single direction and can never be cyclic, so a particular action node can never be referenced by more than one other node in a graph. A workflow must begin with a *start* node, followed by one or more action nodes, and end with an *end* node. Other control nodes or action nodes can be included.

An *action node* represents an Oozie action in the workflow, such as a Hive, Sqoop, Spark, Pig, or Java action, among others. *Control nodes* direct the execution flow in the workflow. Control nodes include the start and end nodes, as well as the fork and decision nodes. In Workflow Manager, the start and end nodes are preconfigured and cannot be modified.

A workflow succeeds when it reaches the end node. If a workflow fails, it transitions to a kill node and stops. The workflow reports the error message that you specify in the message element in the workflow definition.

Using a coordinator, you can assign a schedule and frequency to a workflow. You can also identify specific events that trigger start and end actions in the workflow. If you want to manage multiple recurring workflow jobs as a group, you must add each workflow to a coordinator, then add the coordinators to a bundle.

When you save, validate, or submit a workflow, coordinator, or bundle, they are each saved in their own XML file, also called an application file.

Before a workflow, coordinator, or bundle job can be executed, as a first step the job XML file and all other files necessary to run components of the workflow must be deployed to HDFS. This might include JAR files for Map/Reduce jobs, shells for streaming Map/Reduce jobs, native libraries, Pig scripts, and other resource files. These files are often packaged together as a ZIP file and referred to as a workflow application.

For a practical example of using Workflow Manager to create and monitor a simple Extract-Transform-Load workflow, see "Sample ETL Use Case".

More Information

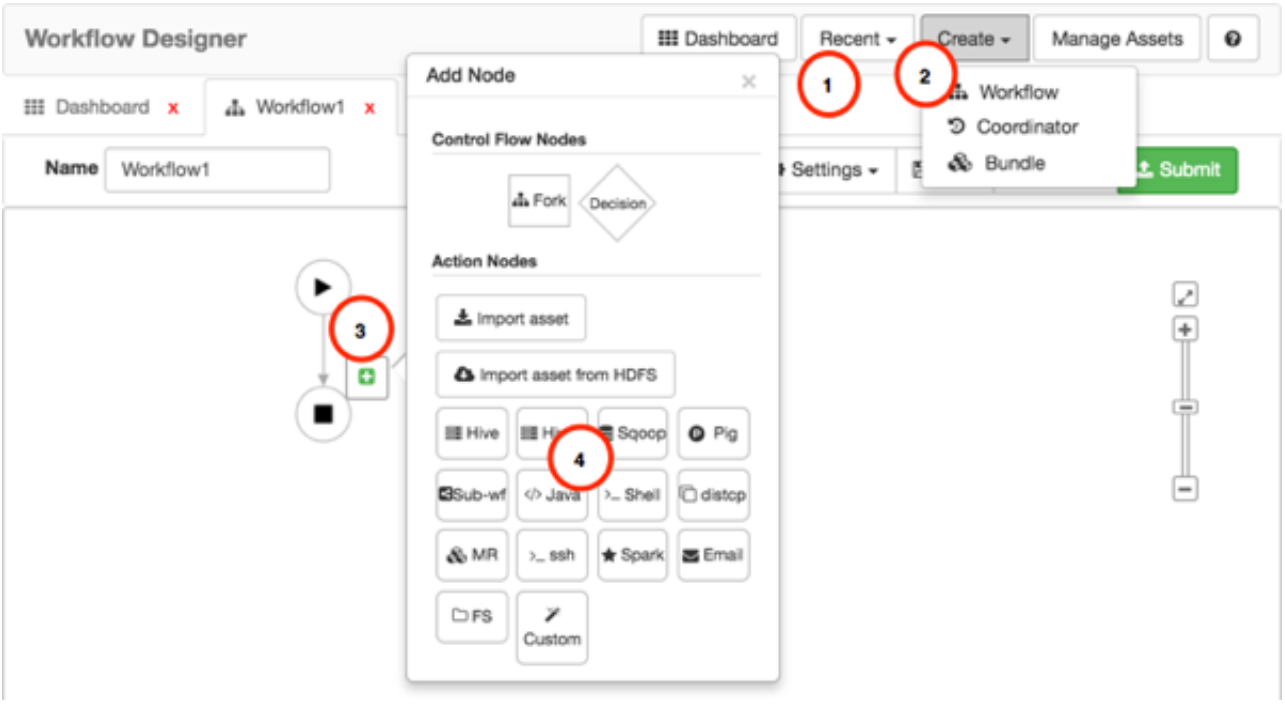
See "Workflow Manager Design Component" in Workflow Manager Basics.

3.4. Using the Design Component

The following information steps you through some of the functionality available in the Workflow Manager design component.

Design Component Functionality

Figure 3.3. Design Workspace, Labeled 1-4



1. Open recently viewed workflows by clicking Recent and double-clicking on a workflow name.

   The design workspace opens with the selected workflow displayed.

2. Create a new workflow, coordinator, or bundle from the Create menu.

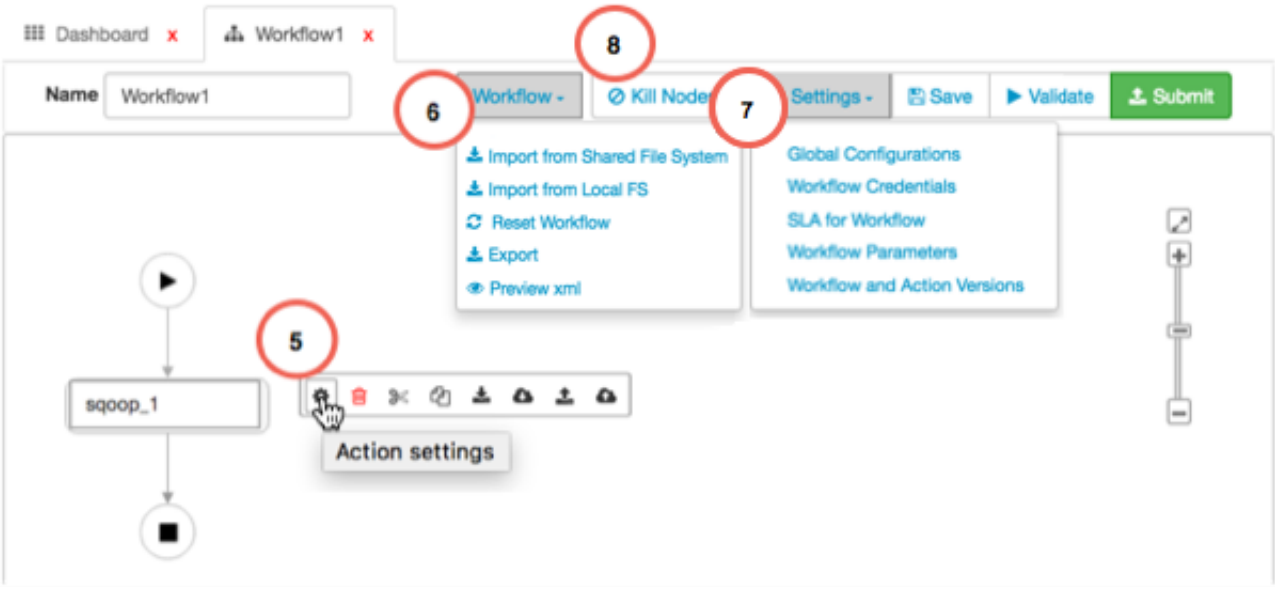
3. Access the list of Action or Control Nodes to add to the workflow by clicking the connecting arrow on the workflow, then clicking the green + icon.

4. Add a node or import an existing asset by clicking one of the items in the Add Node popup.

5. Perform various actions on the node you added by clicking any of the icons on the Node Action Bar. Configure the Action Node or Decision Node by clicking on the Action Settings icon to open the configuration dialog box.

   If you rename the node, note that spaces are not allowed in node names.

Figure 3.4. Design Workspace, Labeled 5-8



6. You can perform various actions on a workflow by clicking Workflow and selecting one of the following items from the menu: Import from Shared Files System, Import from Local FS, Reset Workflow, Export, or Preview XML.

   Reset Workflow discards any changes made to the workflow that have not been saved.

   Important

   If you import an existing workflow XML file, any comments in the file will be lost.

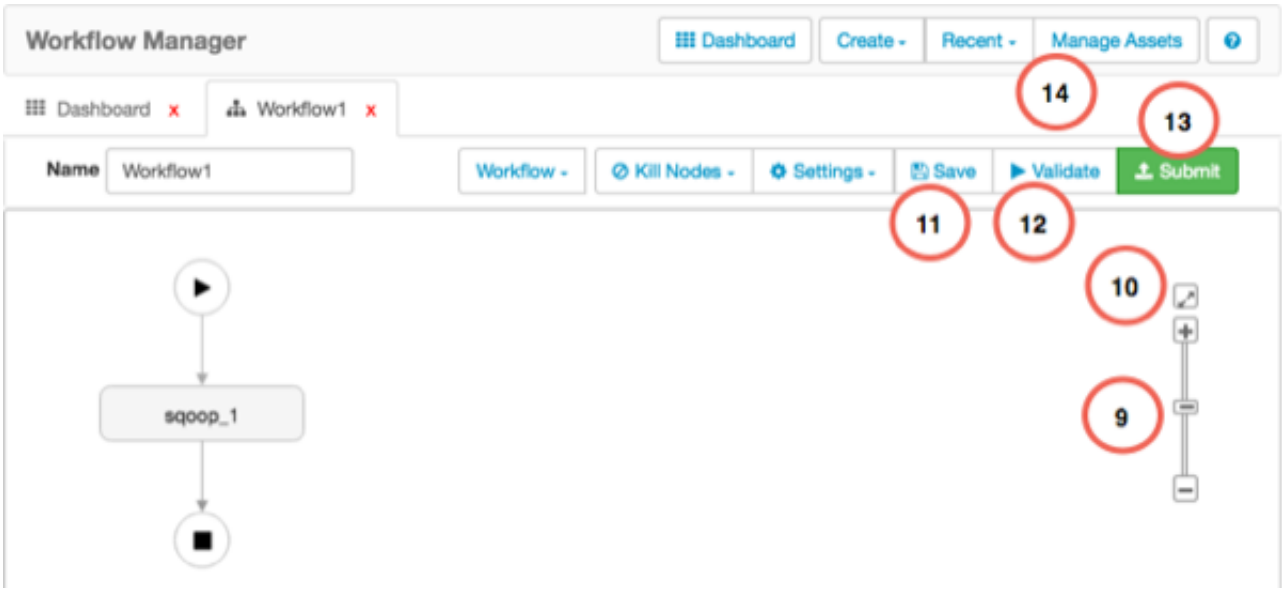
7. You can configure various settings that you can apply to workflows by clicking Settings and selecting one of the following items from the menu: Global Configurations, Workflow Credentials, SLA for Workflow, Workflow Parameters, Workflow and Action Versions.

8. Configure or modify stopping points for workflows by clicking Kill Nodes and selecting Create Kill Node or Manage Kill Nodes.

   Once configured, you select a kill node from the Action Settings for a node.

9. Resize the workflow within the workspace by dragging the slider on the scaler bar.

Figure 3.5. Design Workspace, Labeled 9-14



10. Reposition the workflow to the center of the workspace by clicking the (Reposition) icon.

11. Save a nonvalidated draft or a validated version of the workflow by clicking Save.

    You can save the workflow to any location in HDFS for which you have write permissions.

12. Verify that the XML code in your workflow is correct by clicking Validate.

    Validation only confirms that the XML is properly structured. It does not verify if the workflow will succeed when run.

13. Pass the valid workflow to the job engine in preparation for running the job by clicking Submit.

    When submitting, you can choose whether or not to start the job. If you do not start the job when you submit it, you can start the job from the dashboard. Submitting a workflow results in the workflow displaying in the dashboard.

14. View or delete assets you created by clicking Manage Assets.

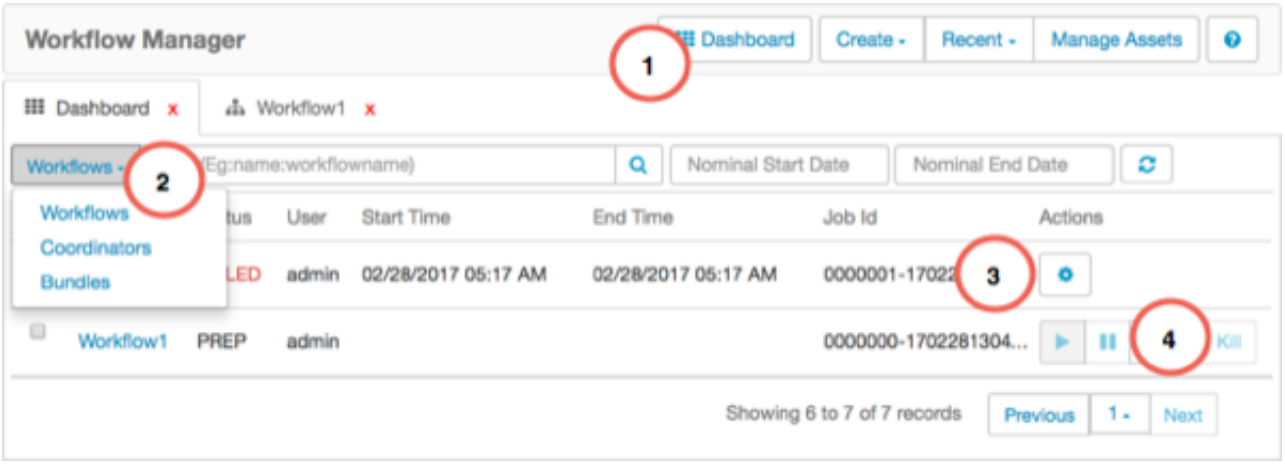
    Only the user who created an asset can view or delete the asset in Asset Manager.

3.5. Using the Dashboard Component

From the dashboard jobs table you can view workflow, coordinator, and bundle jobs, and perform actions on the jobs. You can also access details about each job by clicking the name of a job.

The following information steps you through some of the functionality available in the Workflow Manager dashboard component.

Figure 3.6. Dashboard Jobs Table, Labeled 1-6



1. Access the dashboard by clicking the Dashboard icon.
2. Choose which type of jobs to view by clicking the **Job Type Selector** and selecting Workflows, Coordinators, or Bundles.

   Only jobs that have been submitted to Workflow Manager can display in the table. Saved jobs that have not been submitted cannot be viewed.
3. Display the actions to perform on any job by clicking the **Action** icon.
4. From the action list, you can Start or Resume, Suspend, Retry, or Kill a job.
5. You can filter jobs in the list by name, user, or job ID or by the nominal start or end date and time.
6. You can view additional details about any workflow, coordinator, or bundle job by clicking the name of the job.

   The Job Details page opens.

3.6. Dashboard Job Details Tabs

You can access details about any workflow, coordinator, or bundle job by clicking the name of a job in the dashboard jobs table to access the job details tabs.

Following are examples of the kinds of information provided for each job type.

Details Available for Workflow, Coordinator, and Bundle Jobs [14]

Details Available Only for Workflow Jobs [14]

Details Available Only for Coordinator Jobs [15]

Details Available Only for Bundle Jobs [15]

3.6.1. Details Available for Workflow, Coordinator, and Bundle Jobs

Info Tab	The content of this tab is similar for workflow, coordinator, and bundle jobs. The Info tab provides all of the information available on the jobs table, plus some additional details.
Log Tab	The content of this tab is similar for workflow, coordinator, and bundle jobs. The Log tab displays the WFM log details for workflow, coordinator, and bundle jobs. You can scroll through the log or filter the log using the Search Filter. See the Apache documentation for details about how to use the filter. For coordinator jobs, you can also filter by actions.
Error Log Tab	The content of this tab is similar workflow, coordinator, and bundle jobs. Displays details about any errors encountered when executing the job.
Audit Log Tab	The content of this tab is similar for workflow, coordinator, and bundle jobs. Provides audit details for the job.
Configuration Tab	The content of this tab is similar for workflow, coordinator, and bundle jobs. Displays Oozie configuration information.
Definition Tab	The content of this tab is similar for workflow, coordinator, and bundle jobs. The Definition tab displays the XML definition for the job. The content corresponds to the parameters set when creating the workflow, coordinator, or bundle.

3.6.2. Details Available Only for Workflow Jobs

Action Tab	The content of this tab is available only for workflow jobs. From the Action tab you can view information about each action in the workflow.

You can view further details about each action by clicking the name of the action node in the Name column of the table. Details display in the Info and Configuration tabs below the table.

Tip

You can access the Resource Manager (YARN) log files by clicking the icon in the Job URL column of the table.

Flow Graph Tab — The content of this tab is only available for workflow jobs.

The Flow Graph tab displays the workflow graph. You can view details about any node by clicking on the node. You can modify the workflow by clicking Edit Workflow in the tab bar, which displays the workflow in the workspace of the design component.

Edit Workflow Tab — This function is accessible only from the workflow details.

Clicking Edit Workflow opens the workflow graph in the design component so that you can edit the workflow.

3.6.3. Details Available Only for Coordinator Jobs

Workflow Jobs — The content of this tab is only available for coordinator jobs.

Displays information about the workflow job that is scheduled through the coordinator. By clicking the workflow name in the ID column, you can access details about the workflow that is associated with the coordinator.

Action Reruns Tab — The content of this tab is only available for coordinator jobs.

Provides details about any terminated coordinator actions that were rerun.

3.6.4. Details Available Only for Bundle Jobs

Coordinator Jobs Tab — The content of this tab is only available for bundle jobs.

Displays information about the coordinator jobs that are included in the bundle.

4. Designing Workflows Using the Design Component

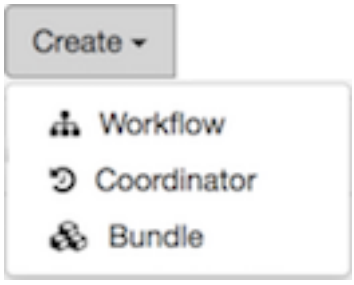
See the following content to learn how to use the Workflow Manager design component to create, modify, save, and submit workflows, coordinators, and bundles.

4.1. Create a Workflow

When you first access Workflow Manager, you need to create a workflow, add action and control nodes, and save or submit the workflow.

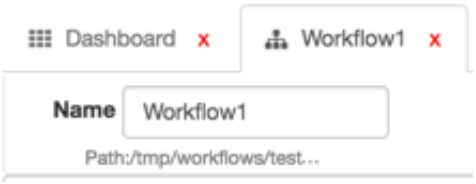
Steps

1. Click the **Create** action menu, then click **Workflow**.



   The Designer workspace displays with a basic workflow graph that includes a start node, an end node, and a connector between the nodes.

2. Type a name for the workflow job in the **Name** field.



Tip

The name you provide in the workspace Name field is the name of the *job*, and it is the name displayed in the dashboard when you submit the workflow to execute. This name is not automatically assigned to the workflow *file* when you save the workflow. If you want the job name and the file name to match, you must enter the same name when you save the workflow.

3. You can now add nodes to the workflow graph, or you can import an existing workflow.

More Information

Section 4.2, "Add Nodes to a Workflow" [17]

Section 4.5, "Import an Existing Workflow" [27]

Section 4.3, "Save a Workflow Draft" [23]

4.2. Add Nodes to a Workflow

You can add either an action node or a control flow node to a workflow. A workflow can include a maximum of 400 nodes.

Note

If a variable is part of an EL expression, you might not be prompted for the value of the variable. In this case, it is best to add the variable in the Custom Job Properties in the Submit dialog box.

More Information

See the following content to learn how to add and configure various node types in a workflow graph.

Add Action Nodes [17]

Add Fork Control Nodes [20]

Add Decision Control Nodes [21]

4.2.1. Add Action Nodes

Steps

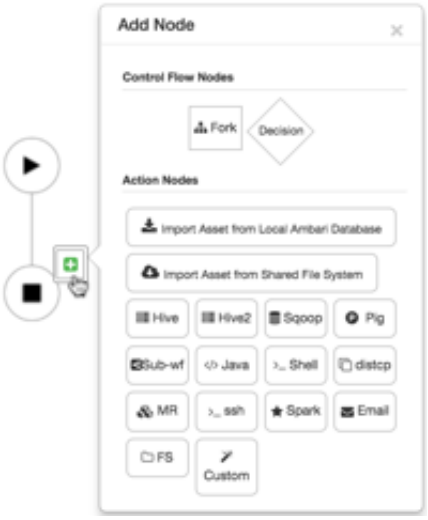
1. Click on the connector between the start and end nodes.

   The + (Add Node) icon displays.

2. Click the green + icon.

   The Add Node dialog box displays.

Figure 4.1. Access the Add Node dialog box



Tip

If using a Custom Action Node, ensure that the JAR containing the Java code and the XML schema definition (XSD) for the custom action have been deployed. The XSD must also be specified in the Oozie configuration.

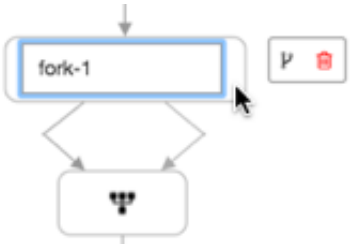
3. Click any node type in the **Action Nodes** section to add the node to the workflow.

   a. Optional: Double-click on the node and type an identifying name for the node.

      Each node in the flow must have a unique name. Spaces are not allowed in the node name.

Figure 4.2. Rename an action node

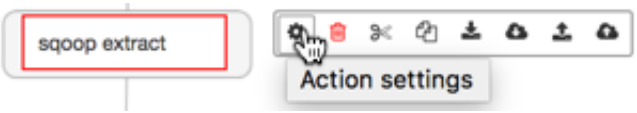
Figure 4.3. Rename a control node



b. Click on the **Action Settings** icon and enter the appropriate parameters for the node type.

The Fork control node does not have action settings.

Figure 4.4. Access the Action Settings



c. Enter the settings for the selected action.

Tip

When entering commands in the action settings, do not enter the component command name. For example, if you were importing data using Sqoop, you would only enter **import –connect <path>** in the Command field, excluding **sqoop**.

More Information

Import an Existing Workflow [27]

Apache Hive Action

Apache Hive2 Action

Apache Sqoop Action

Apache Shell Action

Apache Spark Action

Apache MapReduce, Pig, File System (FS), SSH, Sub-Workflow, and Java Actions

Apache Email Action

Apache Custom Actions

Apache Oozie CLI Utilities

4.2.2. Add Fork Control Nodes

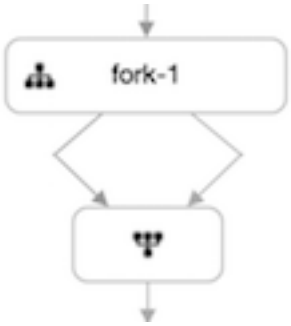
You can use Fork control nodes to direct the flow of actions in the workflow. A fork allows you to create multiple concurrent execution paths that branch from a single action. The paths rejoin at a common point called the join node. A join node waits until every concurrent execution path of a previous fork node arrives to it.

The fork and join nodes must be used in pairs. The join node assumes concurrent execution paths are children of the same fork node.

Steps

1. Add a Fork node by clicking on the connector, clicking the + icon, and selecting **Fork**.

   A fork node and join node are added to the workflow graph, with two forks (connectors).

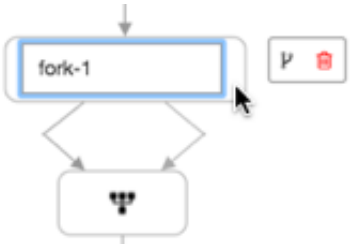


   The join node cannot be modified.

2. Optional: Add another fork by clicking on the fork node, then clicking the (Add Fork) icon.

3. Optional: Double-click on the node and type an identifying name for the node.

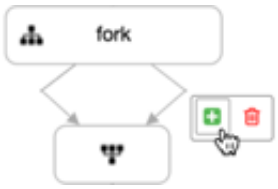
   Spaces are not allowed in node names.



4. Add nodes to the forks:

   a. Click on one of the forks between the fork node and join node.

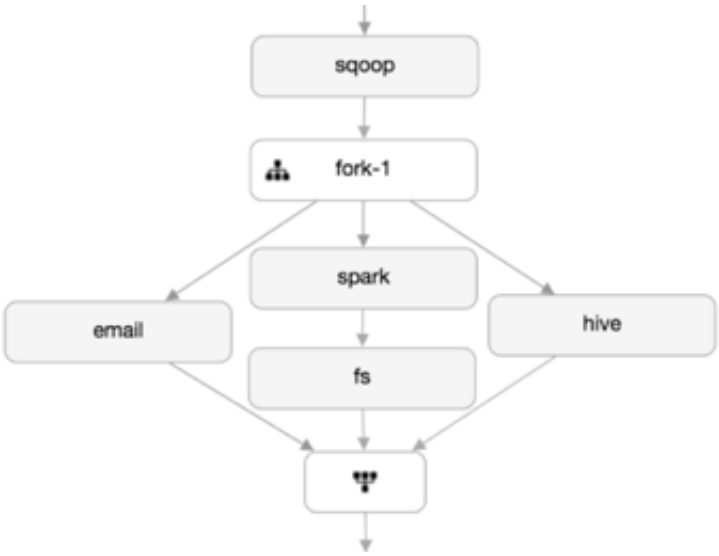
b. Click the + icon and select a node.



c. Repeat steps 2a and 2b for the other forks.

Each fork must have at least one node associated with it. You can have multiple nodes for each fork.

Figure 4.5. Example of a Fork Node



4.2.3. Add Decision Control Nodes

You can use the Decision control nodes to direct the flow of actions in the workflow based on specific circumstances.

Note

If a variable is part of an EL expression, such as in a decision node, you might not be prompted for the value of the variable. In this case, it is best to add the variable in the Custom Job Properties in the Submit dialog box.

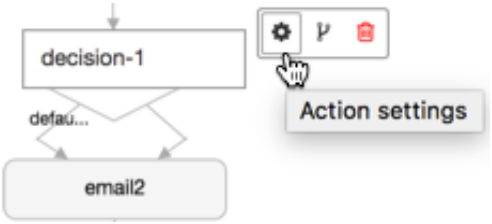
Steps

1. Add a Decision node by clicking on the connector in the graph, clicking the + icon, and selecting **Decision**.

   A decision node is added to the workflow graph, with two decision branches (connectors).

The branch labeled Default identifies the branch the workflow follows if the condition you identify is not met.

2. Click on the decision node, then click the **Action Settings** icon.



3. Add the conditions under which the workflow should progress to the next node in each branch of the workflow.

   Conditions can be set on an action node, a kill node, or an end node.

4. Optional: Add a node to a decision branch.

   a. Click on one of the connectors extending from the Decision node, then click the + icon.

   The Add Node dialog box displays.

   b. Select a node to add to the branch and modify the node settings.

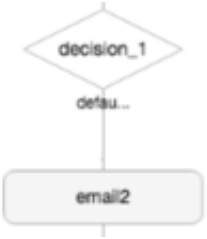
5. Optional: Add a node to the other branch.

Tip

You can nest decision nodes inside of decision nodes and can add forks to decision branches.

6. Optional: Delete a connector.

   You cannot delete the connector that contains the Default setting.



4.3. Save a Workflow Draft

You can save a workflow, whether valid or invalid, to any location on HDFS to which you have access permissions. You can save an invalid draft of your workflow and access it later to complete it.

About This Task

Each workflow file must have a unique name within a single directory. If you want to save a workflow into a directory that already contains a workflow of the same name, you must either rename the existing workflow, add the file to a subdirectory, or overwrite the file during the save procedure.

Tip

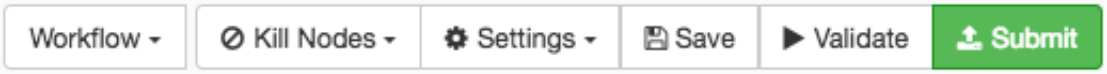
When you save a workflow, the name you assign is given only to the workflow application file. It does not automatically match the name displayed for the workflow job in the dashboard. WFM assigns to the workflow job the name you enter in the Name field in the design workspace. If you want the job name and the application file name to match, you must assign the same name in the Name field and to the application file.

You can save valid workflows or invalid workflows as drafts. The default name for a saved workflow is always `workflow.xml.draft` for a draft workflow that is not validated or

`workflow.xml` for a valid workflow. You can change the workflow file name during the save operation.

Steps

1. With a draft workflow open, click the **Save** icon on the workspace Action Menu.



2. Click **Browse** to select an HDFS directory in which to save the workflow file.

Tip

If the File Browser window opens but does not display a directory list, you might have lost network connectivity or the required data node might be unresponsive.

3. Navigate to the directory you want, click on the directory name, and expand the directory.

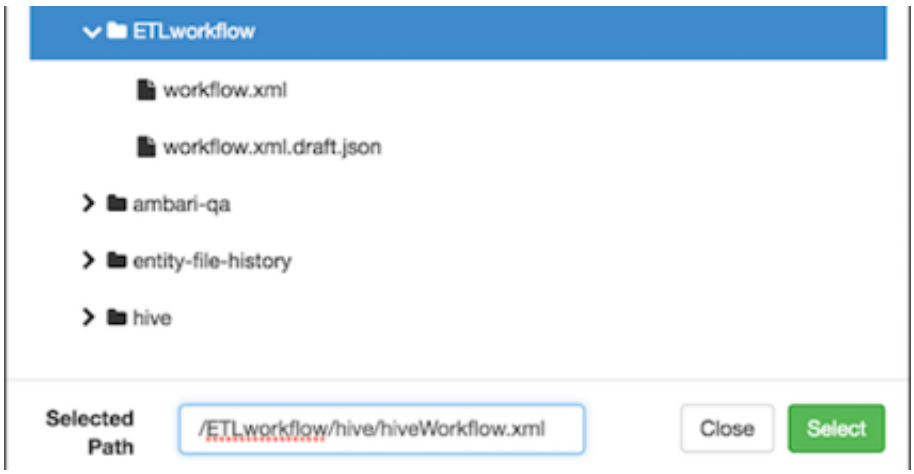
4. Optional: If a workflow file already exists in the directory:

   a. To overwrite the existing file, click **Select**, check **Overwrite** in the Save Workflow dialog box, and then click **Submit**.

   b. To keep the existing file and the new workflow file, in the **Selected Path** field, append a new name for the file or append a new subdirectory to the path, click **Select**, and then click **Submit**.

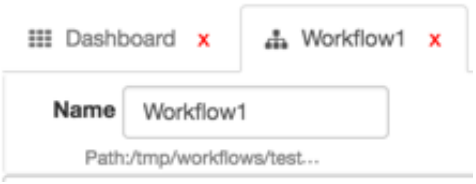
   The workflow file must use the `.xml` extension.

Figure 4.6. Example of saving a workflow to a new directory



   After submitting, the new directory is created and contains the renamed workflow file.

5. Optional: Click in the **Name** field and type a name for the workflow job.



Tip

The name displayed in the Name field is the name used for a job in the dashboard. If you want the job name and the workflow file name to match, you must change both names. The tab name also reflects the content entered in the Name field.

4.4. Validate and Save a Workflow

Workflow validation checks that the XML is correctly coded for the workflow. It does not check if the workflow will succeed when run.

During the validation process, the workflow is also saved.

About This Task

Each workflow file must have a unique name within a single directory. If you want to save a workflow into a directory that already contains a workflow of the same name, you must either rename the workflow, add the file to a subdirectory, or overwrite the file during the save procedure.

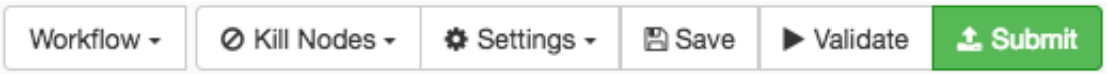
You can save valid workflows or invalid workflows as drafts. The default name for a saved workflow is always "workflow.xml.draft" for a draft workflow that is not validated or "workflow.xml" for a valid workflow. You can change the workflow file name during the save operation.

Tip

When you save a workflow, the name you assign is given only to the workflow application file. It does not automatically match the name displayed for the workflow job in the dashboard. WFM assigns to the workflow job the name you enter in the Name filed in the design workspace. If you want the job name and the application file name to match, you must assign the same name in the Name field and to the application file.

Steps

1. Click the **Validate** icon on the workspace Action Menu.



2. Click **Browse** to select an HDFS directory in which to save the workflow file.

Tip

If the File Browser window opens but does not display a directory list, you might have lost network connectivity or the required data node might be unresponsive.

3. Navigate to the directory you want, click on the directory name, and expand the directory.

4. Optional: If a workflow file already exists in the directory:

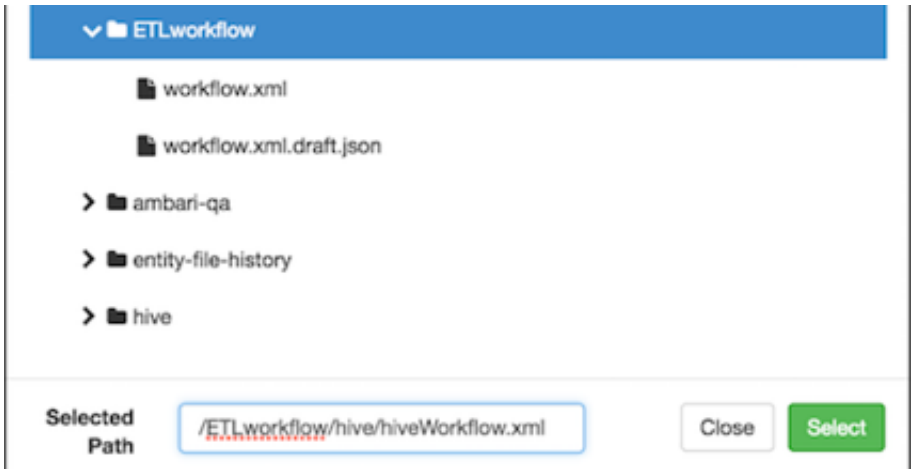
   a. To overwrite the existing file, click **Select**, check **Overwrite** in the Save Workflow dialog box, and then click **Validate**.

   b. To keep the existing file and the new workflow file, in the **Selected Path** field, append a new name for the file or append a new subdirectory to the path, click **Select**, and then click **Validate**.

Important

The workflow file must use the `.xml` extension.

Figure 4.7. Example of saving a workflow to a new directory



After validation, the new directory is created and contains the renamed workflow.

5. Select the **Execution Settings** you want to use.

6. Enter any **Custom Job Properties** you want to use.

7. Click **Validate**.

More Information

Apache Workflow Job Properties

4.5. Import an Existing Workflow

You can import workflows created external to Workflow Manager, or save and import to another WFM View instance any workflows you create within WFM. You can import workflows from within HDFS or from a local file system.

Prerequisite

If you are importing a workflow that was created external to Workflow Manager, ensure that the workflow definition is valid. Workflows created in Workflow Manager are validated, but there is no validation for externally-created workflows. Therefore, Workflow Manager does not provide an error message for an invalid external workflow, nor will an invalid workflow display in the workspace.

Important

If you import an existing workflow XML file, any comments in the file will be lost.

Steps

1. Click **Create** > **Workflow**.

   A new workspace displays with a basic workflow graph.

2. Click the workspace **Workflow** menu and do one of the following.

   - Import from the Hadoop file system:

     a. Click **Import from Shared File System**.

     b. Navigate to the directory that contains the workflow definition file that you want, click on the directory name, and expand the directory.

     c. Verify that a `workflow.xml` definition file exists in the directory.

     ### Tip

     Do not click on the file name. The import action must be performed at the directory level.

     d. Verify that the directory you want is displayed in the **Selected Path** field.

     e. Click **Select**.

   - Import from your local file system:

     a. Click **Import from Local FS**.

     b. Navigate to the local directory that contains the `workflow.xml` definition file that you want and open the file.

   The workflow graph displays in the workspace.

Tip

If the workflow definition is invalid, it does not display in the workspace, and no error message is provided.

More Information

Save a Workflow Draft [23]

4.6. Export a Workflow

You can use the export feature to download a workflow to your local file system.

Steps

1. Create or import a workflow in the workspace.
2. Click **Validate** to ensure that the workflow you want to download has no errors.
3. Click the workspace **Workflow** menu, then click **Export**.
4. The workflow is exported to the default download location on your local file system.
5. Navigate to the download location and verify that the `workflow.xml` file is available.

4.7. Submitting and Executing a Workflow

When you submit a workflow, a validation is performed on the workflow. You cannot submit an invalid workflow.

Prerequisites

A workflow must be valid before you can submit it.

About This Task

When you submit a workflow, the workflow is first saved. You can save workflows to a new directory or overwrite an existing workflow.

Tip

When you save a workflow, the name you assign is given only to the workflow application file. It does not automatically match the name displayed for the workflow job in the dashboard. WFM assigns to the workflow job the name you enter in the Name filed in the design workspace. If you want the job name and the application file name to match, you must assign the same name in the Name field and to the application file.

Once it is saved, the workflow is available to execute as a job; the job does not automatically execute when the workflow is submitted. You must execute a job separately.

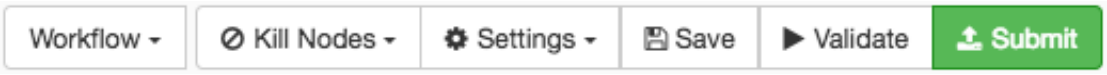
See the following content to learn how to submit and execute workflows.

4.7.1. Submit and Execute an Existing Workflow

You can submit and execute a workflow at the same time.

Steps

1. Create a workflow or import a workflow into the workspace.
2. Click the **Submit** icon on the workspace Action Menu.



   If the workflow is valid, the Submit Workflow dialog box displays.

   If the workflow is invalid, error messages display. You must resolve the identified issues before you can submit the workflow.

3. Choose one of the following:
   - To submit and execute a new workflow:
     a. Click **Browse** and navigate to the directory in which you want to save the workflow.
     b. Click on the directory name, and ensure that the name displays in the **Selected Path** field.
     c. Click **Select**.
   - To submit and execute an existing workflow:
     a. Check **Overwrite** to save the workflow to an existing directory.

        If you don not check Overwrite, you get an error stating that the workflow path already exists.
4. Check **Run on Submit** to execute the workflow job automatically when submission is complete.
5. Click **Submit**.

   A message displays stating that the workflow is saved and providing the job ID.
6. Make note of the job ID and then click **Close**.
7. Verify that the job is running:
   a. Click **Dashboard** in the Workflow Manager Action Bar.

b. In the Dashboard, locate the job and verify that the **Status** is **Running**.

4.7.2. Submit and Execute a New Workflow

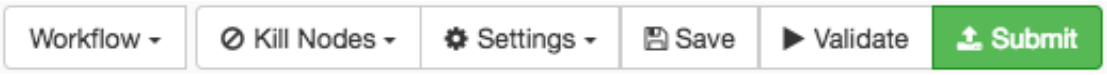
You can submit and execute a workflow at the same time. The workflow must be valid before you can submit it. When you submit the workflow, the workflow file is saved in a location you select. You can save workflow files to any location for which you have access permissions.

Prerequisites

Ensure you have access permissions to the location in which you want to save the workflow.

Steps

1. Create a workflow or import a workflow into the workspace.
2. Click the **Submit** icon on the workspace Action Menu.



   If the workflow is valid, the Submit Workflow dialog box displays.

   If the workflow is invalid, error messages display. You must resolve the identified errors before you can submit the workflow.
3. Click **Browse** and navigate to the directory in which you want to save the workflow.
4. Click on the directory name and ensure that the name displays in the **Selected Path** field.
5. Click **Select** and **Close**.
6. Optional: Check **Overwrite** to replace an existing file of the same name in the same path.

Tip

If you saved the workflow prior to submitting it, then you must select Overwrite or you get an error.

7. Optional: Check **Run on Submit** if you want to execute the workflow job automatically when submission is complete.
8. Click **Submit**.

   A message displays stating that the workflow is saved and providing the job ID.

Tip

If a message displays stating that the job cannot be saved or executed, go to the Ambari UI to see if any services need to be restarted.

9. Make note of the job ID and then click **Close**.

10. Verify that the job is running:

    a. Click **Dashboard** in the Workflow Manager Action Bar.

    b. In the Dashboard, locate the job and verify that the **Status** is **Running**.

       You can locate the job by name or by job ID.

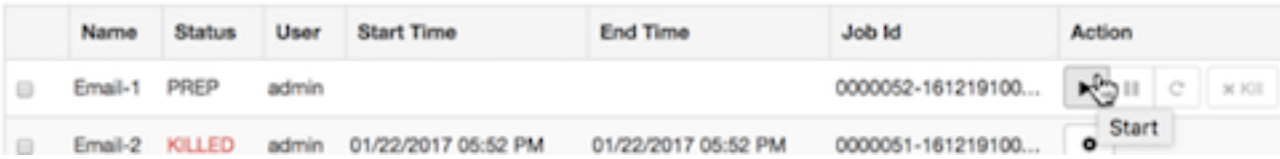
4.7.3. Execute a Submitted Workflow

If you submit a workflow without executing it, you can execute the workflow job any time from the Dashboard.

Steps

1. Click **Dashboard** in the Workflow Manager Action Bar.

2. In the Dashboard, locate the job you want to execute, click the ⚙ (Action) icon, and then click the ▶ (Start) icon.



The Status changes from Prep to Running.

4.8. Modifying Workflows

You can copy, cut, and paste nodes. You can also resize and reposition the workflow graph in the workspace.

See the following content to learn how to copy, move, and delete nodes, and to resize workflow graphs.

Copy or Move Nodes on a Workflow Graph [31]
Remove Nodes from a Workflow Graph [33]
Resize and Move Graph Images [33]

4.8.1. Copy or Move Nodes on a Workflow Graph

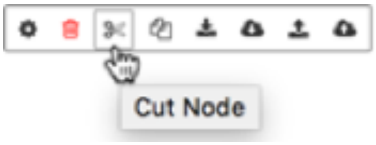
You can duplicate an action node by copying and pasting it into the same workflow or a different workflow. You can move an action node by cutting the node and pasting it to a new location on the same workflow or a different workflow.

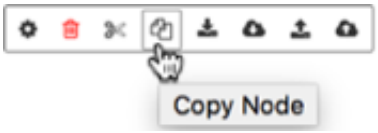
Steps

1. Click on the node you want to duplicate.

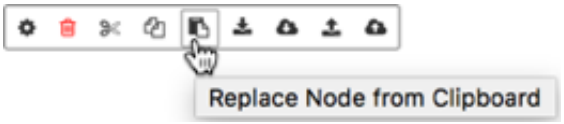
The node action bar displays.

2. Click either **Cut Node** or **Copy Node**.





3. To replace an existing action node, click on the node and select **Replace Node from Clipboard**.

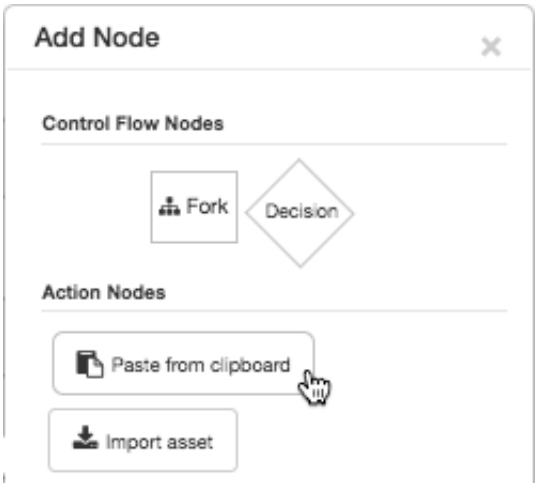


Important

The node is replaced immediately and the action cannot be undone.

When pasting a copied node, "-copy" is appended to the name of the pasted node.

4. To paste a new node into the workflow:
   a. Click the connector line on the workflow where you want to add the node.
   b. Click the green + icon.
   c. Click the **Paste From Clipboard** icon.



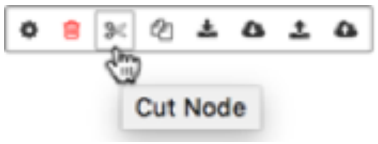
When pasting a copied node, "-copy" is appended to the name of the pasted node.

4.8.2. Remove Nodes from a Workflow Graph

You can remove nodes permanently by using the delete function, or cut the node and have it available on the clipboard.

Steps

1. To cut a node from a graph but keep it on the clipboard for later use:
   a. Click on the node you want to cut.
   b. Click the **Cut Node** icon in the action bar.



2. To permanently delete a node from a graph:
   a. Click on the node you want to delete.
   b. Click the **Delete Node** ( ) icon in the action bar.

4.8.3. Resize and Move Graph Images

There are multiple methods you can use to resize or move a workflow graph in Workflow Manager. This allows you to fit more or less of the image on the workspace, or adjust the location of the graph in the workspace.

See the following content:

4.8.3.1. Resize the Workflow Graph Using the Scaler

1. To increase the size, click the slider on the scaler bar and drag the slider up, or click the + icon.

Figure 4.8. Scaling the workflow

2. To decrease the size, drag the slider down, or click the – icon.

3. To resize and center the graph to fit the workspace, click the icon above the scaler bar.

4.8.3.2. Resize the Workflow Graph Using the Mouse

If your mouse had a scroll function, you can use it to enlarge or reduce the size of the graph.

1. Using the mouse scroll wheel, scroll up to reduce the graph image.

2. Scroll down to enlarge the size of the image.

4.8.3.3. Move the Workflow Graph Using the Scaler

1. To center and resize the graph to fit the workspace, click the icon above the scaler bar.

Figure 4.9. Repositioning the workflow

4.8.3.4. Move the Workflow Graph Using the Mouse

1. Click and hold a connector on the graph or a blank area of the workspace to drag the entire workflow to a new location.

Tip

To scroll the window using the mouse, you must move the cursor outside of the workspace area.

2. Click and hold on any node to drag the node within the workspace.

   The connectors attached to the node move with the node, so dragging a node changes the shape of the graph. Dragging a node cannot be used move the node to a new location within the graph. You can move a node using copy or cut and paste functions.

4.9. Reusing Node Configurations as Assets

You can import or save individual node configurations as *assets* in Workflow Manager (WFM). Assets can be imported or saved to HDFS paths or to an Ambari database. This allows you or others to share and reuse existing or new asset configurations for Hive, Sqoop, Spark, and other workflow actions.

See the following content to learn how to use assets:

Save an Ambari Database Asset to Use Within the WFM Instance [35]
Save an Asset to Shared Storage to Use Across WFM Instances [36]
Import an Asset from the Ambari Database Linked to a Single WFM Instance [37]
Import a Shared Asset File from Shared Storage [38]
Managing Assets [40]

4.9.1. Save an Ambari Database Asset to Use Within the WFM Instance

One way you can save an action node configuration is as an asset in an asset database. This allows you to reuse and share the configuration with multiple users within a single instance of Workflow Manager (WFM) in Ambari. For example, you can share assets among users in a department, who are all using a single instance of WFM.

About This Task

You cannot save an asset with a name that is already being used. Each asset must have a unique name.

Steps

1. Click on the node in the graph that you want to save as an asset.

2. Click **Save Asset to Local Ambari Database** in the Action Bar.

3. Enter a **Name** and **Description**, and then click **Save**.

The asset name must be unique for each Workflow Manager instance.

Tip

Since assets can be shared, providing a clear name and description can help users choose the correct asset quickly.

The asset is now available to be imported into any workflow created within the specific WFM instance. You can manage the asset from the Manage Assets dialog box.

More Information

Section 4.9.3, "Import an Asset from the Ambari Database Linked to a Single WFM Instance" [37]

4.9.2. Save an Asset to Shared Storage to Use Across WFM Instances

You can save an action node configuration as an asset file to reuse and share within and across Workflow Manager (WFM) instances in Ambari. The asset file can be saved to any location on HDFS for which you have access permissions. Anyone using the asset must also have access permissions to the file. For example, you can share assets with multiple departments who are using separate instances of WFM.

Prerequisites

If you want to share the asset file with others, ensure you save the asset to a location with the proper permissions.

About This Task

You cannot save an asset into a directory that already contains an asset of the same name. You must either save the asset with a new name or save it to a uniquely-named location.

If you do not provide a name for an asset file when you publish the asset, the default name given to the asset file is "asset.xml".

Steps

1. Click on the connector in the workflow graph where you want to add the node, and then click the green + icon.
2. Click **Save Asset to Shared File System**.

3. Navigate to the directory you want, click on the directory name, and expand the directory.

4. If an asset file already exists in the directory, in the **Selected Path** field, append a name for the asset file or a name for a new subdirectory, then click **Select**.

After publishing the asset, the new directory is created with the saved asset file.

Tip

Since assets can be shared, providing a descriptive name for the file and directory can help users choose the correct asset quickly.

If you submit an asset file to a directory that already contains an asset of the same name, you receive a notice that there was a problem publishing the asset.

More Information

Section 4.9.4, "Import a Shared Asset File from Shared Storage" [38]

4.9.3. Import an Asset from the Ambari Database Linked to a Single WFM Instance

You can save node asset configurations to an Ambari database and retrieve them from the database by importing them. When importing assets, Workflow Manager displays only the assets for which you have access permissions.

Prerequisites

- The asset must exist in the WFM database.
- You must have permissions for the asset you want to import.

About This Task

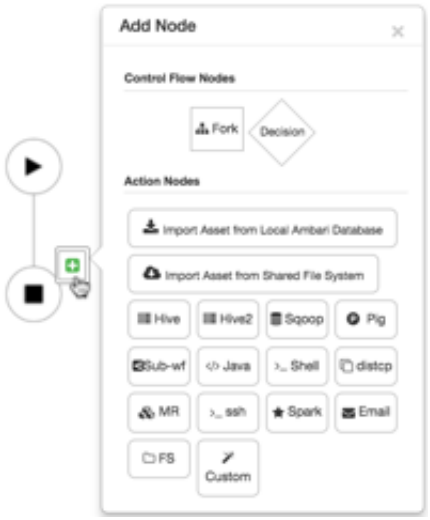
You can import asset configurations into an existing node on the workflow graph, or you can import an asset as a new node on the graph.

Steps

1. To import an asset as a new node on the graph:

   a. Click on the connector in the workflow graph where you want to add the node, and then click the green + icon.

   b. Click **Import Asset from Local Ambari Database** in the Add Node dialog box.



The Import Asset dialog box opens, displaying all assets available in the Workflow Manager instance, regardless of who created them.

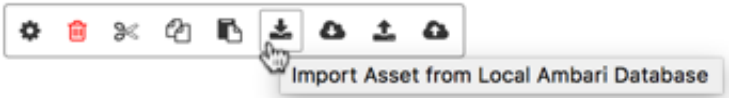
c. Select the asset you want and click **Import**.

The asset is imported into the graph as a new node.

2. To import an asset configuration into an existing node in the graph:

a. Click on the node in the workflow graph.

b. Click **Import Asset from Local Ambari Database** in the node Action Bar.



The Import Asset dialog box opens, displaying only the asset types you can use for the node selected. For example, if you are importing to a Hive node, only Hive assets are available.

c. Select the asset you want and click **Import**.

The asset configuration is imported into the node on the graph.

4.9.4. Import a Shared Asset File from Shared Storage

Asset files allow you to reuse a node configuration on a workflow from any instance of Workflow Manager. You can import asset files from an HDFS path, as long as you have permissions to the file.

Prerequisites

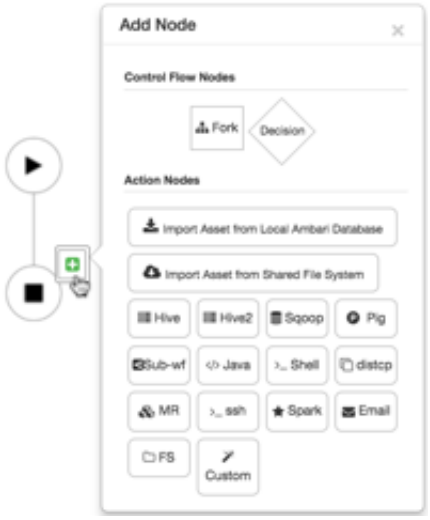
- The asset file must exist in an HDFS directory.
- You must have permissions for the asset file you want to import.

Steps

1. To import an asset as a new node on the graph:

   a. Click on the connector in the workflow graph where you want to add the node, and then click the green + icon.

   b. Click **Import Asset from Shared File System** in the Add Node dialog box.



   The Import Asset dialog box opens, displaying all assets available in the Workflow Manager instance, regardless of who created them.

   c. Select the asset you want and click **Import**.

   The asset is imported into the graph as a new node.

   A new node is added to the workflow graph with the configuration parameters from the imported asset.

2. To import a shared HDFS asset into an existing node in the workflow graph:

   a. Click on the node in the workflow graph.

   b. Click **Import Asset from Shared File System**.

   c. Navigate to the location of the asset file.

   d. Select the file, ensure it displays in the **Selected Path**, and click **Select**.

   The asset configuration is imported into the node on the graph.

4.9.5. Managing Assets

You can view, modify, delete, and edit assets that you created, whether they are database assets or HDFS assets. You must have access permissions to modify or delete an asset. However, how you manage assets and what you can do to them depends on the type of asset you are working with.

See the following content about managing assets:

View or Delete Assets in the Ambari Database [40]
Edit Assets in the Ambari Database [40]
Delete Shared Assets from Shared Storage [41]
Edit Assets in Shared Storage [41]

4.9.5.1. View or Delete Assets in the Ambari Database

From Manage Assets, you can view a list of assets or delete assets that were saved to the Ambari database. Only the user who created a database asset can view or delete that asset in the Asset Manager dialog box.

Steps

1. Click **Manage Assets** in the Workflow Manager Action Bar.

   Asset Manager opens.

2. Filter the list of assets by entering a search term in the **Asset Search** field.

   The search will filter results to match any characters that are part of the name, type, or description of an asset.

3. Delete an asset by clicking the trash icon, and then clicking **Continue**.

4. Click **Cancel** to close the dialog box.

4.9.5.2. Edit Assets in the Ambari Database

You can edit assets by using the Save Asset to Local Ambari Database function. If you want to save the modified asset using the original file name, you must delete the existing asset after importing it into the workflow.

Steps

1. Import a database asset into the workflow.

2. Click the Action Settings icon, modify the configuration, and click Save.

3. Save the modified asset by doing one of the following:

   - Save using the original asset file name.

     a. Click **Manage Assets**.

b. Locate the name of asset you are modifying and click the (delete) icon.

c. Click the node you want to save and click **Save Asset to Local Ambari Database**.

d. Enter the asset name and description and click Save.

- Save using a new asset file name.

a. Click the node you want to save and click **Save Asset to Local Ambari Database**.

b. Enter the new name and description for the asset and click Save.

4.9.5.3. Delete Shared Assets from Shared Storage

You can view or edit shared HDFS assets in Workflow Manager.

If you want to delete an asset from HDFS, you must do so from the Ambari UI.

Steps

1. Click the (Ambari Views) icon, and then click **Files View**.

   Ambari displays the HDFS files and directories.

2. Navigate to the file you want to delete.

3. Click on the row for the file you want to delete.

   An action bar displays above the list of files.

4. Click **Delete** in the action bar, then click **Delete** in the confirmation message.

5. Click , then click the Workflow Manager instance name to return to Workflow Manager.

4.9.5.4. Edit Assets in Shared Storage

You can edit shared HDFS assets in Workflow Manager. If you want to save the modified asset using the original file name, you must delete the existing asset after importing it into the workflow.

Steps

1. Import the asset as a node in the workflow graph.

2. Click the node containing the asset configuration.

3. Click **Action Settings**, modify the configuration, and click Save.

   Clicking Save only saves the changes in the dialog box, it does not save the asset.

4. Optional: Delete the asset file from HDFS using the Ambari Files View.

   Complete this step only if you intend to save the asset file using the original file name.

5. Click the **Save Asset to Shared File System** icon.

6. Browse to the directory in which you want to save the asset.
7. Optional: Append a name for the asset file in the Selected Path field.

   If you do not append a name, the default name given to the file is `asset.wfasset`.
8. Click **Select**.

4.10. Creating Coordinators and Bundles

You can use a coordinator to schedule a workflow, and you can group coordinators (scheduled workflows) into a bundle to manage the scheduled workflows as a single unit.

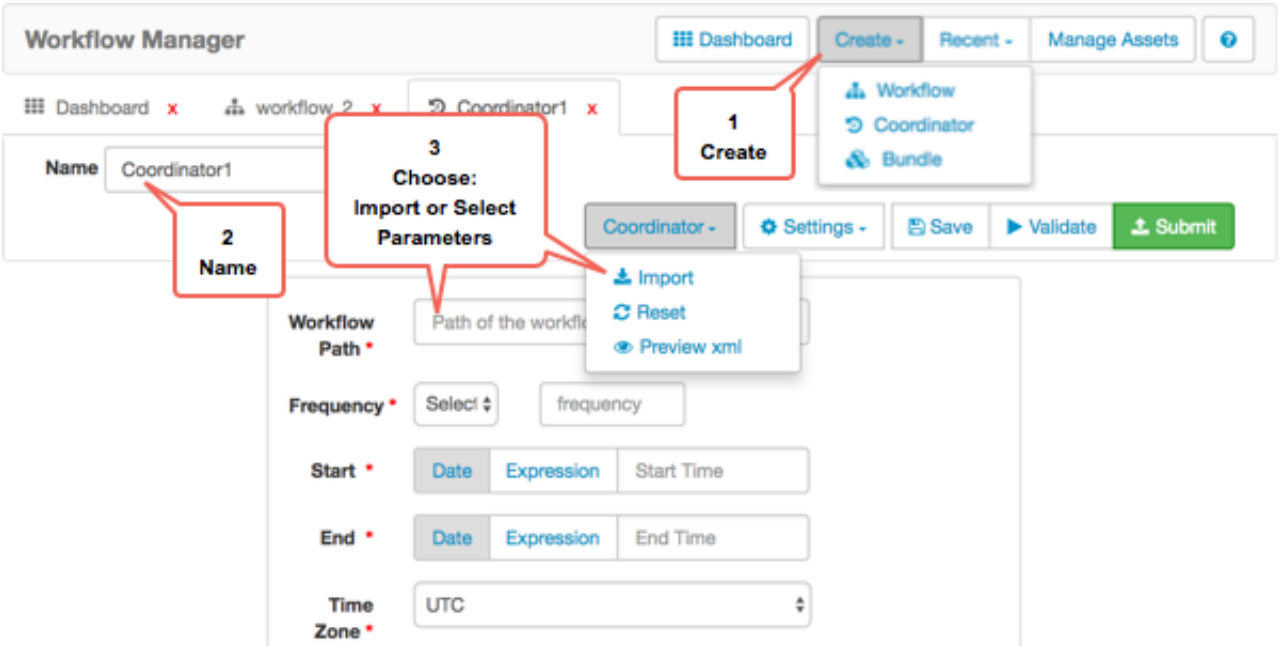
See the following content:

Create a Coordinator [42]
Create a Bundle [43]

4.10.1. Create a Coordinator

Use the coordinator to schedule recurring jobs for a workflow, and to identify any events that you want to trigger the workflow job.

Figure 4.10. Creating a Coordinator



Steps

1. Click **Create Coordinator** on the Workflow Manager Action Bar.
2. Enter a name for the coordinator in the **Name** field.

Tip

When you save a workflow, the name you assign is given only to the workflow application file. It does not automatically match the name displayed for the workflow job in the dashboard. WFM assigns to the workflow job the name you enter in the Name filed in the design workspace. If you want the job name and the application file name to match, you must assign the same name in the Name field and to the application file.

3. **Choose** one of the following:
   - Enter the values for the coordinator parameters.

     See the Apache Coordinator documentation for details about the parameters.

Important

If entering Frequency in Minutes, the value must be a minimum of 5 minutes.

   - Click **Coordinator** > **Import** to retrieve the parameters from an existing coordinator.

Tip

If the File Browser window opens but does not display a directory list, you might have lost network connectivity or the required data node might be unresponsive.

     a. Navigate to the coordinator you want, click on the coordinator file name, and ensure the correct file name is displayed in the **Selected Path**.

     b. Click **Select**.

        The configuration parameters of the selected coordinator display in the fields of the Create Coordinator window. You can modify the parameters as needed.

4. Click **Save**, **Validate**, or **Submit** in the Action Bar.

More Information

Apache Coordinator Specification

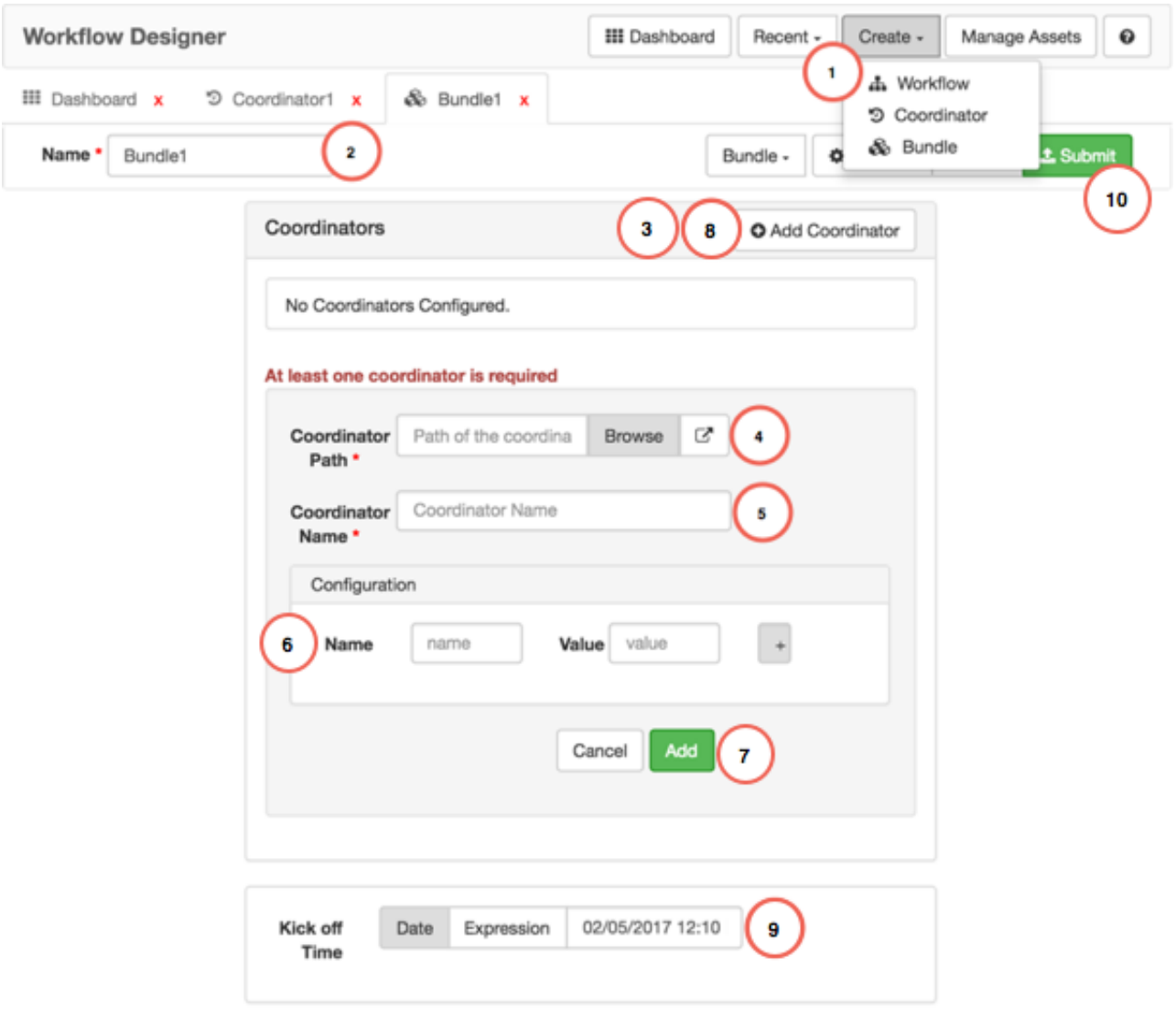
Coordinator Use Case Examples

4.10.2. Create a Bundle

Bundles allow you to group multiple workflows to provide better operational control. You can perform actions such as start, stop, resume, suspend and so forth on the entire

bundle of workflows. To be included in a bundle, workflows must first be associated with a coordinator schedule or event. Schedule each workflow in a coordinator, then group the coordinators (scheduled workflows) in a bundle.

Figure 4.11. Creating a Bundle



Steps

1. Click **Create** > **Bundle** in the Workflow Manager Action Bar.

2. Enter a name for the bundle in the **Name** field.

Tip

When you save a workflow, the name you assign is given only to the workflow application file. It does not automatically match the name displayed for the workflow job in the dashboard. WFM assigns to the workflow job the name you enter in the Name filed in the design workspace.

If you want the job name and the application file name to match, you must assign the same name in the Name field and to the application file.

3. Click **Add Coordinator** to display the fields for selecting and configuring the first coordinator in your bundle.
4. Browse to the path of the first coordinator to include in the bundle.
5. Enter a name for the coordinator.

   Having a unique name for each coordinator can be useful when viewing bundle job details and logs, especially if you use the same coordinator more than once in the bundle.
6. Optional: Enter any configuration parameters you want applied to the coordinator (the scheduled workflow) at the time the bundle job runs.

   These parameters override any parameters set in the workflow or coordinator.
7. Click **Add** to include the coordinator in the bundle.
8. Click **Add Coordinator** and configure additional coordinators for your bundle.
9. Optional: Select a **Kick-off Time** for the bundle.

   If you do not select a time to start the bundled job, you can start it manually from the dashboard.
10. Click **Submit**.

    After submission, the bundle job displays in the dashboard. The job has a status of Prep until it executes.

More Information

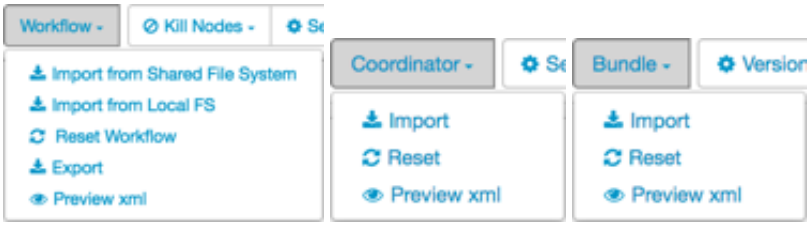
Apache Bundle Specification

4.11. View the XML Code

From the design component you can view the XML code for a workflow, coordinator, or bundle application. However, you cannot modify the XML.

Steps

1. Click the **Workflow**, **Coordinator**, or **Bundle** menu.



2. Click **Preview XML**.

   The XML for the workflow, coordinator, or bundle application displays.

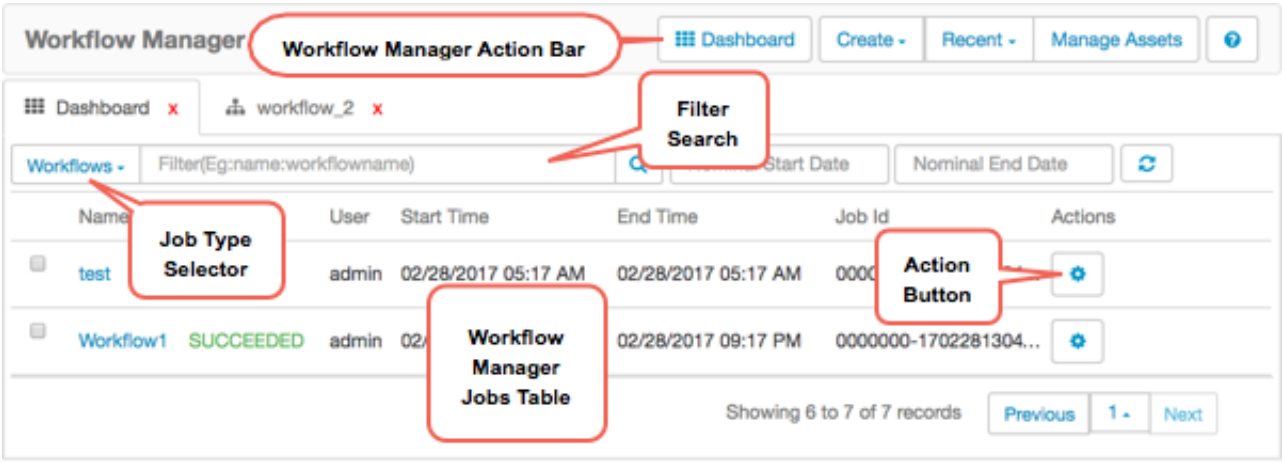
   You cannot edit the XML code from Workflow Manager.

5. Monitoring Jobs Using the Dashboard

After you submit a workflow, coordinator, or bundle in Workflow Manager, the job can be viewed in the dashboard.

For a practical example of using Workflow Manager to create and monitor a simple Extract-Transform-Load workflow, see "Sample ETL Use Case".

Figure 5.1. Workflow Manager Dashboard



See the following content to learn about the dashboard:

Verify the Status of a Job [47]

Identify the Location of a Job XML File [49]

Troubleshooting Job Errors [50]

5.1. Verify the Status of a Job

You can display lists of workflow, coordinator, or bundle jobs from which you can verify the current state of any job. You can filter the list of jobs to make it easier to locate the information you need.

About This Task

After displaying the list of jobs, you can filter the jobs by querying the name, status, user, or ID of a job, or you can filter by start and end dates. You can use the search filter and date filter together.

Following are tips for using the search filter:

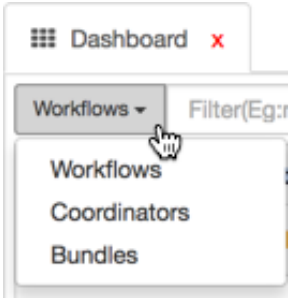
- Only whole terms are filtered.

  Example: Searching on the name "work" displays any job named "work", but it would not display a job named "workflow".

- Search is case sensitive.
- The filter searches on *name:value* pairs, but you only enter the value, then select the pair from the list of result options.

Steps

1. From the dashboard, click the **Job Type Selector** and choose Workflows, Coordinators, or Bundles to view.



   The table displays whichever job type you select.

   The Status column displays the current state of each job.

2. Use filters to limit the number of jobs listed in the dashboard.

   a. Type a term in the Filter field.

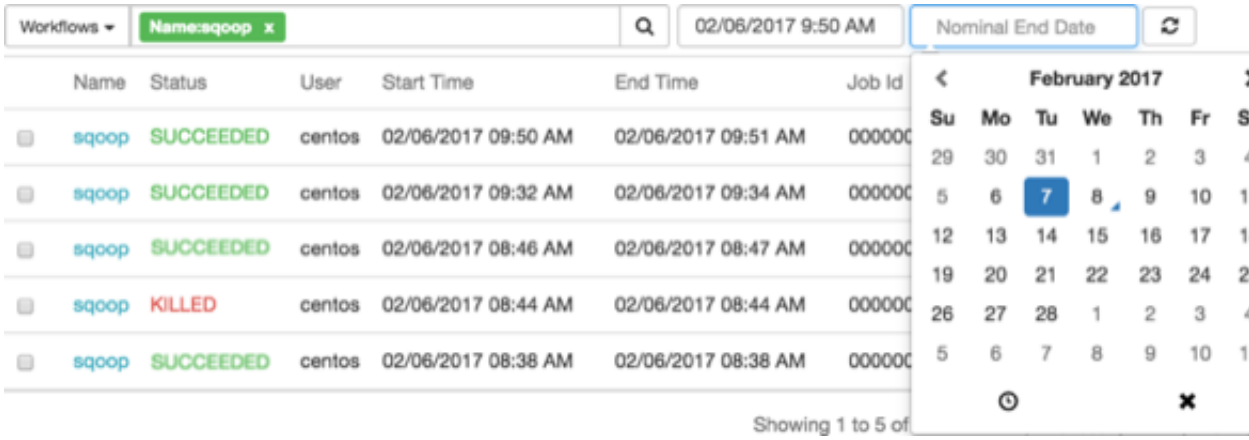
      As you type, the filter displays a popup showing possible *name:value* pairs.

   b. Click one of the *name:value* pairs listed in the popup.

      The table displays items that match the search filter term.

   c. Click the Nominal Start Date or Nominal End Date field and type, or select from the popup calendar, a date and time.

      Use the format *mm/dd/yyyy hh:mm AM/PM*, such as *02/07/2017 4:33 PM*.



3. Update the dashboard listings by clicking the (Refresh) icon.

   Some actions take time to execute, so you might need to refresh the dashboard to see the updated status.

More Information

Job States

Apache wiki: Filtering the server logs with logfilter options

5.2. View Job Details and Logs

You can view details about workflow, coordinator, or bundle jobs from the dashboard.

1. From the dashboard, click the **Job Type Selector** and choose Workflows, Coordinators, or Bundles to view.

   The table displays whichever job type you select.

2. Click the name of the job for which you want to locate the job file.

   The job details display.

3. Click any tab above the details area to view additional information.

4. Coordinator job only: In the Workflow Jobs table for the coordinator, click the workflow name in the ID column to access details about the workflow that is associated with the coordinator.

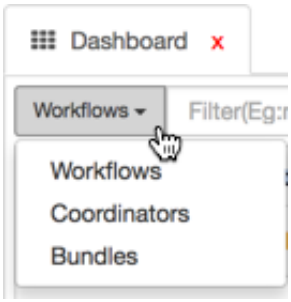
More Information

See "Dashboard Job Details Tabs" in Quick Start

5.3. Identify the Location of a Job XML File

You can determine where a Workflow Manager `job.xml` file, also called an application file, is located if you want to share the file, delete it, and so forth.

1. From the dashboard, click the **Job Type Selector** and choose Workflows, Coordinators, or Bundles to view.



   The table displays whichever job type you select.

2. Click the name of the job for which you want to locate the job file.

   The job details display.

3. Choose one of the following:

   - Click the **Info** tab and scroll to the **App Path** entry.
   - Click the **Configuration** tab and scroll to the **oozie.<job-type>.application.path** entry.

   Both entries show the full path to the job application file.

5.4. Troubleshooting Job Errors

You can access details about any job. When a job does not succeed, you can use the details to do some basic troubleshooting. You can also access MapReduce (YARN) logs and the Ambari Files View from Workflow Manager.

Tip

When viewing jobs in the dashboard, ensure that you have selected the appropriate job type (workflow, coordinator, or bundle) from the Job Type Selector.

Basic Job Troubleshooting [50]

Search the Log Output [51]

Open an Existing Workflow to Edit [52]

5.4.1. Basic Job Troubleshooting

You can do some basic troubleshooting from the dashboard, including the following:

- Verify that you are logged in as the correct user.

  View the username displayed on the Sign Out button.

- Verify ownership of the directory and file.

  Click the Views icon and select Files View, navigate to the directory or file you want to verify, and verify the content in the Owner column.

- View the Error Message & Error Code.

  Click the Action tab, click the name of the action node for which you want further detail, then click the Action Details Info tab for the node.

- View the Error Log.

  Click the Error Log tab in the job details area.

- Check for typos or misconfigurations in the job file.

  Click the Definition tab to view the XML for the job (app) file.

- Access the MapReduce YARN logs.

  Click the Action tab, and then click the icon in the Job URL column.

5.4.2. Search the Log Output

You can search the content of the Log tab for any workflow, coordinator, or bundle. You can also search the actions listed in the Action Reruns tab.

About This Task

Following are tips for using the Search Filter:

- Searches are done only on text strings within the log.
- Search is case sensitive.
- Only one word can be searched at a time.
- Can search partial terms.
- You can search using the following optioins: recent, start, end, loglevel, text, limit, and debug.
- You can search more than one term at a time.
- Enter terms using the structure **option=value**.

  Example: limit=4;recent=2h;loglevel=ERROR

  The above example displays the oldest four log entries that were written within the past two hours that have a status of ERROR.

Following are tips for using the Action List filter:

- Action IDs are the same as the IDs listed on the Workflow Jobs tab.

- Enter a single digit, a range, or a comma-separated list of numbers.
- The Action ID is the final number following the ACTION entry.

  For example, the action ID in the following entry is 8:
  ACTION[0000092-170206174330753-oozie-oozi-C@8]

1. In the dashboard table, click a job name.

   The details tabs for the job display.
2. Click the **Log** tab.
3. Click the **Search Filter** field and enter a term to search.
4. Click **Get Logs**.

   The filtered list displays.
5. To retrieve the full log file, delete the search term and then click **Get Logs** again.
6. Coordinators only:
   a. Click the **Action List** field and enter the number of a coordinator action.
   b. Click **Get Logs**.

      The filtered list displays.
   c. To retrieve the full log file, delete the search term and then click **Get Logs** again.

5.4.3. Open an Existing Workflow to Edit

Steps

1. In the dashboard table, access the workflow jobs table.
2. Click on the workflow name to access the job details.

   The Edit Workflow button is available to the right of the tab names.
3. Click the **Edit Workflow** button.

   The workflow graph displays in the design workspace.

Click on Edit Workflow to the left of tab names

6. Sample ETL Use Case

You can use Sqoop and Hive actions in a workflow to perform a common ETL flow: extract data from a relational database using Sqoop, transform the data in a Hive table, and load the data into a data warehouse using Sqoop.

Following is an example of a common simple Sqoop>Hive>Sqoop ETL workflow created in Workflow Manager. In this example, we extract customer data from a MySQL database, select specific data to include in a Hive table, and then load the data into a data warehouse.

Prerequisites

To successfully execute the example Sqoop>Hive>Sqoop ETL workflow defined below, the following prerequisites must be met.

- Apache Hive and Apache Sqoop have been successfully installed and configured.
- You successfully completed the tasks in "Configuring WorkFlow Manager View" in the Ambari Views guide.
- All node managers must be able to communicate with the MySQL server.

Workflow Tasks

The sample workflow consists of the following:

- Configure the Cluster [53]

  Create an HDFS Directory for Each New User
  Create a Proxy User
  Copy JAR Files

- Create and Submit the Workflow [56]

  Create the Sqoop Action to Extract Data
  Create the Hive Action to Transform Data
  Create the Sqoop Action to Load Data

- Monitor the Workflow Job [62]

6.1. Configure the Cluster

You must complete several tasks at the command line or in the Ambari web UI prior to creating the workflow in Workflow Manager.

See the following tasks to configure the cluster.

Create an HDFS Directory for Each New User [54]

Create a Proxy User [54]

Copy Files [55]

6.1.1. Create an HDFS Directory for Each New User

Each user who is allowed access to Workflow Manager must have a user home directory in HDFS.

Steps

1. Log in as the HDFS user.
2. Create a directory for the user.

```
hdfs dfs -mkdir -p /user/$USERNAME
```

   In the example above, replace $USERNAME with the user name of the new user.

3. Change ownership on the directory to the new user.

```
hdfs dfs -chown $USERNAME:$HDFS_USER /user/$USERNAME
```

4. Change the permissions on the directory to read/write/execute for the new user and the groups the user belongs to.

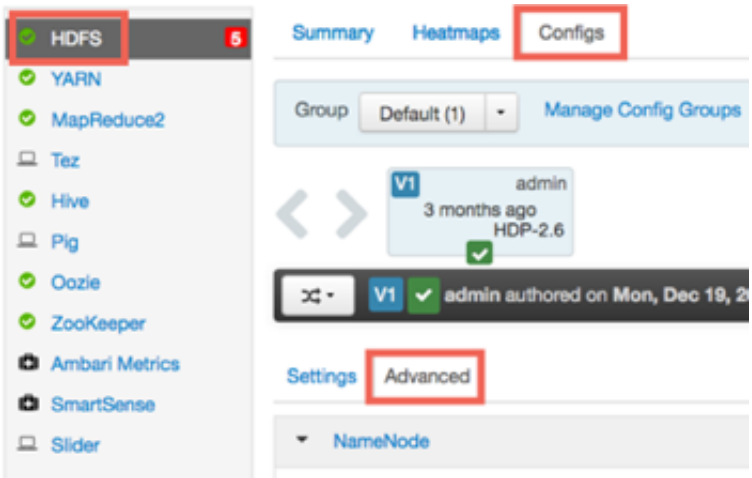
```
hdfs dfs -chmod -R 770 /user/$USERNAME
```

6.1.2. Create a Proxy User

You must specify the proxy user properties for groups and hosts for the Oozie process in the Hadoop `core-site.xml` file.

Steps

1. In Ambari, select the **HDFS** component.
2. Click **Configs>Advanced**.



3. Scroll to the **Custom core-site** section and expand the section.
4. Specify the following properties:

Field name: **hadoop.proxyuser.oozie.groups**
Value: **$USER_GROUPS_THAT_ALLOW_IMPERSONATION**
Description: A comma-separated list of the groups that the impersonated user belongs to.

Field name: **hadoop.proxyuser.oozie.hosts**
Value: **$OOZIE_SERVER_HOSTNAME**
Description: A comma-separated list of the hosts from which the user can connect to impersonate a user in the specified groups.

6.1.3. Copy Files

All required configuration files for the workflows you intend to create must be installed prior to running the workflow job. For this ETL use case example, you need the MySQL JAR file, and the configuration files for Hive and Tez.

Prerequisites

Ensure that you have the latest supported MySQL version for your environment, according to the Support Matrices.

In this example workflow, the MySQL driver JAR file is shared across the cluster, rather than identifying the file in each workflow. So the file must be copied to a shared location.

Steps

1. Login to the HDFS server as the Oozie user. For example:

   su - oozie

   Identify and copy the name of the timestamped subdirectory of the Oozie `/share/lib` directory.

2. ```
   oozie admin -shareliblist
   oozie admin -sharelibupdate
   ```

   The output of the **-sharelibupdate** command shows the `lib_$TIMESTAMP` directory. You use the timestamp directory name in the following steps.

3. Copy the `mysql-connector*.jar` file to the Sqoop `lib` directory so Sqoop can access MySQL.

   Example:

   ```
   hdfs dfs -put /$PATH/mysql-connector-java-5.1.37.jar /user/oozie/share/lib/
   lib_$TIMESTAMP/sqoop
   ```

   Check the Support Matrices for latest supported MySQL version for your environment.

4. Copy the configuration files `hive-site.xml` and `tez-site.xml` to the Hive `lib` directory and rename them.

   Example:
   ```

```
hdfs dfs -put /$PATH/hive-site.xml /user/oozie/share/lib/lib_$TIMESTAMP/
hive/hive-conf.xml
hdfs dfs -put /$PATH/hive-site.xml /user/oozie/share/lib/lib_$TIMESTAMP/tez/
tez-conf.xml
```

Important

There can be only one configuration file named $COMPONENT-site.xml in a `/lib` directory in HDFS for each Apache component. Therefore, you must either rename any copied $COMPONENT-site.xml file or put it in a directory other than a `/lib` directory.

5. Update the server to use the newer version of the `/share/lib` directory.

```
oozie admin -sharelibupdate
```

More Information

Support Matrices

6.2. Create and Submit the Workflow

After the required directories are created and permissions set, and the required configuration and JAR files are copied to the proper locations, you are ready to create the Sqoop>Hive>Sqoop ETL workflow in the Workflow Manager Ambari View.

See the following content to create and submit the ETL workflow:

Access the Workflow Manager View [56]

Create the Sqoop Action to Extract Data [57]

Create the Hive Action to Transform the Data [58]

Create the Sqoop Action to Load the Data [59]

Submit and Execute the Workflow Job [61]

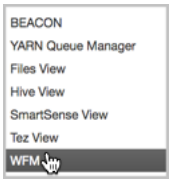
6.2.1. Access the Workflow Manager View

You access the Workflow Manager View from Ambari.

Steps

1. Access the Ambari UI.
2. Click the (Ambari Views) icon.
3. Click the name of the Workflow Manager View.

   The name of the view depends on the name assigned when the WFM instance was created.



Workflow Manager View opens.

6.2.2. Create the Sqoop Action to Extract Data

You must add a Sqoop action to the ETL workflow to extract the data from the database.

Steps

1. Click **Create>Workflow**.
2. Click the connector between the Start and End nodes, then click the + icon.
3. With the workflow graph displayed in the workspace, click the **Name** field and enter a descriptive name for the workflow.

   For example, `ETL workflow`.

   The name in this field is the name that displays for the workflow job in the WFM dashboard.
4. Click the **Sqoop icon** to add an action node to the workflow.

   This Sqoop action will extract data from the MySQL database.
5. Click the **Sqoop node** in the workflow graph and rename it using a descriptive name.

   For example, name the node `sqoop-extract`. Spaces are not allowed in node names.

   This is necessary because there will be two Sqoop actions in this workflow, and each node in a workflow must have a unique name. Having descriptive node names is also helpful when identifying what a node is intended to do, especially in more complicated workflows.

   Tip

   Spaces are not allowed in action node names.
6. Click the **Sqoop node** again and then click the **Action Settings** gear icon.
7. In the Sqoop action dialog box, select **Command**.

   In the Sqoop action settings, you can choose to use commands or arguments with a `job-XML` element to run the job. This example uses a command.
8. In the **Command** field, enter a command to extract data.

   For example:

```
import --connect jdbc:mysql://wfmgr-5.openstacklocal/customer-data --
username wfm --password-file /user/wfm/.password
--table marketing --split-by rowkey --hive-import -m 1
```

This Sqoop command imports the MySQL data from the database `customer-data` into a Hive table called `marketing`. The password for user *wfm* is called from a password file.

9. Expand the **Advanced Properties** section and do the following:

   a. Browse to the directory that contains the Hive and Tez configuration files you copied into a `lib` directory and add those resources to the **File** fields.

      For example:

      `/user/wfm/oozie/apps/lib/lib_$TIMESTAMP/hive/hive-conf.xml`

      `/user/wfm/oozie/apps/lib/lib_$TIMESTAMP/tez/tez-conf.xml`

   b. In the **Prepare** section, select **delete**, and then browse for or type the path to be deleted.

      Selecting **delete** ensures that if a job is interrupted prior to completion, any files that were created will be deleted prior to re-executing the job, otherwise the rerun cannot complete.

      You can optionally include the *`delete`* option in the Command field.

10. Use the default settings for the remaining fields and options.

11. Click **Save** and close the dialog box.

More Information

Apache Sqoop Action

Apache Sqoop User Guide

6.2.3. Create the Hive Action to Transform the Data

Hive stores user metadata in HDFS. By default, the location in HDFS for this metadata is `/user/`$USERNAME where $USERNAME is the name of the user who is accessing Hive. If this directory does not exist with read/write/execute permissions for the user running the workflow job, the job fails.

Steps

1. In the workflow graph, click the connector that follows the Sqoop node, then click the + icon.

2. Click the **Hive icon** to add an action node to the workflow.

   This Hive action will be used to transform the data you extracted using Sqoop.

3. Click the **Hive node** and rename it using a descriptive name.

   For example, name the node **`hive-mktg-ids`**.

   Tip

   Spaces are not allowed in action node names.

4. Click the **Hive node** again and then click the **Action Settings** gear icon.

5. In the **Action Settings** dialog box, click the **Query** option.

   You can either use a script or a query, but for this example we use Query.

6. In the Query field, type the query you want to use.

   For example:

   ```
   INSERT OVERWRITE DIRECTORY '/usr/output/marketing/customer_id' SELECT * FROM
    marketing WHERE Id_Field > 100;
   ```

   This query extracts from the table named `marketing` all IDs over 100 from the field Id_Field and places the data in `/usr/output/marketing/customer_id`.

7. Enter the JDBC URL.

   For example: `jdbc:hive2://wfmgr-2.openstacklocal:10000`

   This entry can be found in Ambari at Hive>Summary on the HiveServer2 JDBC URL line.

8. Enter the password for the MySQL database.

9. Use the default settings for the remaining fields and options.

10. Click **Save** and close the dialog box.

More Information

Apache Hive Configuration Properties

Apache Hive Action

Apache Hive Operators and Functions

Apache Hive Query Language

Apache Hive2 Action

Apache Beeline Hive Commands

6.2.4. Create the Sqoop Action to Load the Data

Steps

1. In the workflow graph, click the connector between the Start and End nodes, then click the + icon.
2. Click the **Sqoop icon** to add another Sqoop action node to the workflow.

   This Sqoop action will be used to load the transformed data to a specified location.
3. Click the **Sqoop node** in the workflow graph and rename it using a descriptive name.

   For example, name the node `sqoop-load`.

   This is necessary because there will be two Sqoop actions in this workflow, and each node in a workflow must have a unique name. Having descriptive node names is also helpful when identifying what a node is intended to do, especially in more complicated workflows.
4. Click the **Sqoop node** again and then click the **Action Settings** gear icon.
5. In the Sqoop action dialog box, select **Command**.
6. In the **Command** field, enter a command to extract data.

   For example:

   ```
   export --connect jdbc:mysql://wfmgr-5.openstacklocal/customer-data --
   username wfm --password-file /user/wfm/.password
   --table exported --input-fields-terminated-by "\001" --export-dir /usr/
   output/marketing/customer_id
   ```

   The password for user *wfm* is called from a password file.
7. In the **Advanced Properties** section, browse to the directory that contains the Hive and Tez configuration files you copied into a `lib` directory and add those resources to the **File** fields.

   For example:

   `/user/wfm/oozie/apps/lib/lib_$TIMESTAMP/hive/hive-conf.xml`

   `/user/wfm/oozie/apps/lib/lib_$TIMESTAMP/tez/tez-conf.xml`
8. In the **Prepare** section, select **delete**, and then browse for or type the path to be deleted.

   Selecting **delete** ensures that if a job is interrupted prior to completion, any files that were created will be deleted prior to re-executing the job, otherwise the rerun cannot complete.

   You can optionally include the *`delete`* option in the Command field.
9. Use the default settings for the remaining fields and options.
10. Click **Save** and close the dialog box.

More Information

Apache Sqoop Action

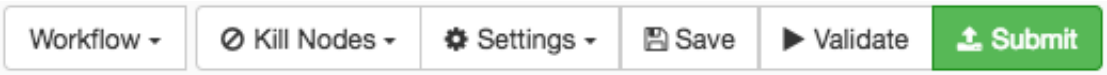
Apache Sqoop User Guide

6.2.5. Submit and Execute the Workflow Job

From the Workflow Manager workspace, you can submit a workflow, which saves and validates the XML of the workflow. You can choose to submit the job without running it, so you can run it at a later time, or you can run the job on submission. If you choose to submit without running the job, the workflow job displays in the WFM dashboard with status of PREP.

Steps

1. With the workflow graph displayed in the WFM workspace, click **Submit** on the Workspace Action Bar.



   A validation check is run. If the workflow XML is valid, the Submit Workflow dialog box displays.

   The validation check does not determine whether or not the workflow will succeed.

2. In **Workflow Path**, browse to the location where you want to save the workflow.
3. Select the directory name and ensure it displays in the **Selected Path** field.

   For example: `/user/wfm/oozie/apps/`

4. In the **Selected Path** field, append a subdirectory name and the name of the workflow file.

   The workflow file must end in `.xml`.

   For example: `/user/wfm/oozie/apps/workflows/etl-workflow.xml`

   Both the subdirectory and file are created.

5. Click **Select**.
6. In the Custom Job Properties section, add the following:

   Name: **`oozie.action.sharelib.for.sqoop`**

   Value: **`sqoop,hive,hcatalog`**

7. Check **Overwrite**.

   This replaces any existing file of the same name in the same path.

Tip

If you saved the workflow prior to submitting it, then you must select Overwrite or you get an error.

8. Check **Run on Submit** to execute the workflow job automatically when submission is complete.
9. Use the default settings for the remaining fields and options.
10. Click **Submit**.

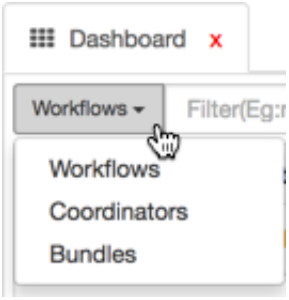
    A message displays stating that the workflow is saved and providing the job ID.

    When you successfully submit, the workflow is saved to HDFS, and the job is submitted and executed.
11. Click **Close**.

6.3. Monitor the Workflow Job

Steps

1. Verify that the job is running:
   a. Click **Dashboard** in the Workflow Manager Action Bar.
   b. From the Job Type List, select **Workflows** to display in the jobs table.



   c. In the Dashboard, verify that the job you submitted, `ETL workflow`, is visible in the jobs table with a Status of **Running**.

      You can locate the job by name or by job ID.
   d. Click the (Refresh) icon to ensure you are viewing the latest status information.
2. View the related workflow logs.
   a. In the WFM dashboard, click the workflow name, **ETL workflow**.
   b. In the job details, click the **Log** tab to view the Oozie job log.
3. View the output of the YARN job.
   a. Click the **Action** tab.
   b. In the Job URL column, click the icon for one of the actions, such as `sqoop-extract`.

The Map-Reduce YARN logs display.

More Information

HCC articles on implementing WFM nodes: Apache Ambari Workflow Manager View for Apache Oozie

Apache Hive Action

Apache Hive Commands

Apache Hive2 Action

Apache Beeline Hive Commands

Apache Sqoop Action

Apache Sqoop User Guide

7. Workflow Parameters

You enter required and optional parameters in the nodes on a workflow graph in the design component. The available action types and their associated parameters are described in the following:

7.1. Hive Action Parameters

Following are descriptions and examples for the parameters you can set in the Hive action node.

Use of Credentials and SLA can be set in the Hive action, but the configuration for them is done from the global Settings menu.

Table 7.1. Hive Action, General Parameters

Parameter Name	Description	Additional Information	Example
Hive Option	The options are Script or Query.	You can run the Hive action using a script or by entering a query in HiveQL.	
Script	Navigate to the HDFS location of the script containing the Hive queries.	You can bundle Hive queries together in a script for faster execution.	/user/ambari-qa/processworkflow/queries/create_drivers.hql
Query	You can enter HiveQL commands instead of using a script to request and retrieve data.	See the Apache documentation for more information.	create table temp_drivers (col_value STRING);

Parameter Name	Description	Additional Information	Example
			LOAD DATA INPATH
Job XML	You can select one or more job.xml files to pass Hive configuration details.	The configuration file that specifies the variables used in the workflow that allow Hive to communicate with the metastore. Can be overwritten or replaced by entries under the Configuration section.	hive-conf.xml
Param	Use to pass the values of variables referenced in the script or HiveQL.	See the Apache documentation for more information.	If hive query is: select * from table where joindate= ${joinDate} Param should be: <param>joinDate=13-11-15</param>

Table 7.2. Hive Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

Table 7.3. Hive Action, Advanced Properties Parameters

Parameter Name	Description	Additional Information	Example
Resource Manager	Master node that arbitrates all the available cluster resources among the competing applications.	The default setting is discovered from the cluster configuration.	${resourceManager}
Name Node	Manages the file system metadata.	Keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. Clients contact NameNode for file metadata or file modifications.	${nameNode}
File	Select any files that you want to make available to the Hive action when the workflow runs.		MySQL data files
Archive	Select any archives that you want to make available to the Hive action when the workflow runs.		archived data files
Prepare	Select **mkdir** or **delete** and identify any HDFS paths to create or delete before starting the job.	Use **delete** to do file cleanup prior to job execution. Enables Oozie to retry a job if there is a transient failure (the job output directory must not exist prior to job start). If the path is to a directory: **delete** deletes all content recursively and then deletes the directory. **mkdir** creates all missing directories in the path.	
Arg	Identify any arguments to be passed to the Hive script.		

Table 7.4. Hive Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions. See the Apache documentation for more information.	

7.2. Hive2 Action Parameters

Following are descriptions and examples for the parameters you can set in the Hive2 action node.

Use of Credentials and SLA can be set in the Hive action, but the configuration for them is done from the global Settings menu.

Table 7.5. Hive2 Action, General Parameters

Parameter Name	Description	Additional Information	Example
Hive Option	The options are Script or Query.	You can run the Hive action using a script or by entering a query in HiveQL.	
Script	Navigate to the HDFS location of the script containing the Hive queries.	You can bundle Hive queries together in a script for faster execution.	/user/home/ambari-user/hive-queries/hive_query.hql
Query	You can enter HiveQL commands instead of using a script to request and retrieve data.	See the Apache documentation for more information.	select * from default.wfm;
jdbc-url	The JDBC user name, to provide remote access to the JDBC driver.	This entry is discovered and auto-completed, but can be changed.	jdbc:hive2://servername:10000
Password	The password for the JDBC user name, to provide remote access to the JDBC driver.	This entry is discovered and auto-completed, but can be changed.	
Job XML	You can select one or more job.xml files to pass Hive2 configuration details.	The configuration file that specifies the variables used in a workflow that allow Hive2 to communicate with the metastore. Can be overwritten or replaced by entries under the Configuration section.	hive-conf.xml
Param	Use to pass the values of variables referenced in the script or HiveQL.	See the Apache documentation for more information.	If hive query is: select * from table where joindate=${joinDate} Param should be: <param>joinDate=13-11-15</param>

Table 7.6. Hive2 Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

Table 7.7. Hive2 Action, Advanced Properties Parameters

Parameter Name	Description	Additional Information	Example
Resource Manager	Master node that arbitrates all the available cluster resources among the competing applications.	The default setting is discovered from the cluster configuration.	${resourceManager}
Name Node	Manages the file system metadata.	Keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. Clients contact NameNode for file metadata or file modifications.	${nameNode}
File	Select any files that you want to make available to the Hive2 action when the workflow runs.		MySQL data files
Archive	Select any archives that you want to make available to the Hive2 action when the workflow runs.		archived data files
Prepare	Select **mkdir** or **delete** and identify any HDFS paths to create or delete before starting the job.	Use **delete** to do file cleanup prior to job execution. Enables Oozie to retry a job if there is a transient failure (the job output directory must not exist prior to job start). If the path is to a directory: **delete** deletes all content recursively and then deletes the directory. **mkdir** creates all missing directories in the path.	
Arg	Identify any arguments to be passed to the Hive script.		

Table 7.8. Hive2 Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions. See the Apache documentation for more information.	

7.3. Sqoop Action Parameters

You can use the Apache Sqoop action to move structured data between Apache Hadoop and relational databases. You can import data into files in a specified location in your Hadoop cluster. You can also use Sqoop to extract data from Hadoop and export it to relational databases outside of Hadoop. Sqoop works with several databases, including Teradata, Netezza, Oracle, MySQL, and Postgres.

Table 7.9. Sqoop Action, General Parameters

Parameter Name	Description	Additional Information	Example
Send As	Options are **Command** and **Args** (arguments).		
Command	You can enter a valid Sqoop command.	See the Apache Sqoop documentation for available commands.	import –connect jdbc:mysql://wfm.openstacklocal/test –username centos –password-file /user/centos/.password –table wfm –split-by rowkey –hive-import -m 1
Args	You can enter one or more valid Sqoop arguments.	See the Apache Sqoop documentation for available arguments.	–connect jdbc:mysql://wfm.openstacklocal/test –username
Job XML	You can select one or more job.xml files to pass Sqoop configuration details.	The configuration file that specifies the variables used for the Sqoop action in the workflow. Can be overwritten or replaced by entries under the Configuration section.	

Table 7.10. Sqoop Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

Table 7.11. Sqoop Action, Advanced Properties Parameters

Parameter Name	Description	Additional Information	Example
Resource Manager	Master node that arbitrates all the available cluster resources among the competing applications.	The default setting is discovered from the cluster configuration.	${resourceManager}
Name Node	Manages the file system metadata.	Keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. Clients contact NameNode for file metadata or file modifications.	${nameNode}
File	Select any files that you want to make available to the Sqoop action when the workflow runs.		/user/centos/oozie/apps/sqoop/lib/hive-site.xml
Archive	Select any archives that you want to make available to the Sqoop action when the workflow runs.		archived data files
Prepare	Select **mkdir** or **delete** and identify any HDFS paths to create or delete before starting the job.	Use **delete** to do file cleanup prior to job execution. Enables Oozie to retry a job if there is a transient failure (the job output directory must not exist prior to job start). If the path is to a directory: **delete** deletes all	

Parameter Name	Description	Additional Information	Example
		content recursively and then deletes the directory. **mkdir** creates all missing directories in the path.	
Arg	Identify any arguments to be passed to Sqoop.		

Table 7.12. Sqoop Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions. See the Apache Sqoop documentation for more information.	

7.4. Pig Action Parameters

Table 7.13. Pig Action, General Parameters

Parameter Name	Description	Additional Information	Example
Script	Navigate to the HDFS location of the Pig script.		/user/ambari/scripts/testreport.pig
Job XML	You can select one or more job.xml files to pass Pig configuration elements.	The configuration file that specifies the variables used for the Pig action in the workflow. Can be overwritten or replaced by entries under the Configuration section.	

Table 7.14. Pig Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

Table 7.15. Pig Action, Advanced Properties Parameters

Parameter Name	Description	Additional Information	Example
Resource Manager	Master node that arbitrates all the available cluster resources among the competing applications.	The default setting is discovered from the cluster configuration.	${resourceManager}
Name Node	Manages the file system metadata.	Keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. Clients contact NameNode for file metadata or file modifications.	${nameNode}
File	Select any files that you want to make available to the Hive action when the workflow runs.		MySQL data files

Parameter Name	Description	Additional Information	Example
Archive	Select any archives that you want to make available to the Hive action when the workflow runs.		archived data files
Prepare	Select **mkdir** or **delete** and identify any HDFS paths to create or delete before starting the job.	Use **delete** to do file cleanup prior to job execution. Enables Oozie to retry a job if there is a transient failure (the job output directory must not exist prior to job start). If the path is to a directory: **delete** deletes all content recursively and then deletes the directory. **mkdir** creates all missing directories in the path.	
Arg	Identify any arguments to be passed to the Hive script.		
Param	Templatized variables of the form ${VARIABLE, to be passed to the Pig script.	Oozie performs the parameter substitution before executing the Pig job.	

Table 7.16. Pig Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions.	

7.5. Sub-Workflow Action Parameters

Table 7.17. Sub-Workflow Action, General Parameters

Parameter Name	Description	Additional Information	Example
App Path	Navigate to the path of the workflow application file of the child workflow job.		
Propagate Configuration	Determines whether or not the workflow job configuration should be propagated to the child workflow.		

Table 7.18. Sub-Workflow Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions.	

Table 7.19. Sub-Workflow Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.

Parameter Name	Description	Additional Information	Default Setting
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

7.6. Java Action Parameters

The Java action executes the public static void main(String[] args) method of the specified main Java class.

Table 7.20. Java Action, General Parameters

Parameter Name	Description	Additional Information	Example
Main Class			
Java Options	Select either Java Option List or Java Optioins.		
Job XML	You can select one or more job.xml files to pass Java configuration elements.	The configuration file that specifies the variables used for the Java action in the workflow. Can be overwritten or replaced by entries under the Configuration section.	
Capture Output	Indicates whether or not Oozie will capture output of the STDOUT of the Java action execution. Command output must be in Java Properties file format if another action will pick it up.	Useful if the output of the Java action needs to be used from within decision nodes.	

Table 7.21. Java Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

Table 7.22. Java Action, Advanced Properties Parameters

Parameter Name	Description	Additional Information	Example
Resource Manager	Master node that arbitrates all the available cluster resources among the competing applications.	The default setting is discovered from the cluster configuration.	${resourceManager}
Name Node	Manages the file system metadata.	Keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. Clients contact NameNode for file metadata or file modifications.	${nameNode}
File	Select any files that you want to make available to the Java action when the workflow runs.		/path/file
Archive	Select any archives that you want to make available to the Java action when the workflow runs.		/path/archived-data-files
Prepare	Select **mkdir** or **delete** and identify any HDFS paths to create or delete before starting the job.	Use **delete** to do file cleanup prior to job execution. Enables Oozie to retry a job if there is a transient failure	

Parameter Name	Description	Additional Information	Example
		(the job output directory must not exist prior to job start). If the path is to a directory: **delete** deletes all content recursively and then deletes the directory. **mkdir** creates all missing directories in the path.	
Arg	Identify any arguments for the main Java function.	The value of each arg element is considered a single argument and they are passed to the main method in the same order.	

Table 7.23. Java Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions.	

7.7. Shell Action Parameters

Table 7.24. Shell Action, General Parameters

Parameter Name	Description	Additional Information	Example
Exec	Path to the script file or the shell command.		/user/ambari-qa/linecount.sh (where linecount.sh contains a script that does a specific operation) Or echo "NumberOfLines=`hadoop fs -cat $1 | wc -l`"
Job XML	You can select one or more job.xml files to pass Shell configuration elements.	The configuration file that specifies the variables used for the Shell action in the workflow. Can be overwritten or replaced by entries under the Configuration section.	
Environment Variable			

Table 7.25. Shell Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

Table 7.26. Shell Action, Advanced Properties Parameters

Parameter Name	Description	Additional Information	Example
Resource Manager	Master node that arbitrates all the available cluster resources among the competing applications.	The default setting is discovered from the cluster configuration.	${resourceManager}
Name Node	Manages the file system metadata.	Keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. Clients contact NameNode for file metadata or file modifications.	${nameNode}
File	Select any files that you want to make available to the Java action when the workflow runs.		/path/file
Archive	Select any archives that you want to make available to the Java action when the workflow runs.		/path/archived-data-files
Prepare	Select **mkdir** or **delete** and identify any HDFS paths to create or delete before starting the job.	Use **delete** to do file cleanup prior to job execution. Enables Oozie to retry a job if there is a transient failure (the job output directory must not exist prior to job start). If the path is to a directory: **delete** deletes all content recursively and then deletes the directory. **mkdir** creates all missing directories in the path.	
Arg	Identify any arguments to be passed to the shell script.	The value of each arg element is treated as a single argument, even if there are white spaces, and the values are passed to the main method in order.	

Table 7.27. Shell Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions.	
Capture Output	Indicates whether or not Oozie will capture output of the STDOUT of the shell command execution. Command output must be in Java Properties file format if another action will pick it up.	Useful if the output of the shell command needs to be used from within decision nodes.	

7.8. DistCp Action Parameters

DistCp is used for copying files from one cluster to another, or copying files within a single cluster.

Table 7.28. DistCp Action, General Parameters

Parameter Name	Description	Additional Information	Example
Arg	Arguments to the distcp commands.		Copy a file/directory from source to target:

Parameter Name	Description	Additional Information	Example
			hdfs://nn1:8020/dir/file hdfs://nn2:8020/dir/file Copy file from 2 source directory to the target: hdfs://nn1:8020/dir/a \ hdfs://nn1:8020/dir/b \ hdfs://nn2:8020/dir/dir2 Copy file from source to target by overwriting content in target, if available: -update hdfs://nn1:8020/source/first hdfs://nn1:8020/source/second hdfs://nn2:8020/target
Java Opts	Use to set Java options to the DistCp command.		-Xmn256m

Table 7.29. DistCp Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

Table 7.30. DistCp Action, Advanced Properties Parameters

Parameter Name	Description	Additional Information	Example
Resource Manager	Master node that arbitrates all the available cluster resources among the competing applications.	The default setting is discovered from the cluster configuration.	${resourceManager}
Name Node	Manages the file system metadata.	Keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. Clients contact NameNode for file metadata or file modifications.	${nameNode}
Prepare	Select **mkdir** or **delete** and identify any HDFS paths to create or delete before starting the job.	Use **delete** to do file cleanup prior to job execution. Enables Oozie to retry a job if there is a transient failure (the job output directory must not exist prior to job start). If the path is to a directory: **delete** deletes all content recursively and then deletes the directory. **mkdir** creates all missing directories in the path.	

Table 7.31. DistCp Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions.	

7.9. Map-Reduce (MR) Action Parameters

Table 7.32. MR Action, General Parameters

Parameter Name	Description	Additional Information	Example
Mapper Class	Specifies the Java class that should be used as mapper.	The JAR containing the class must be available in the Oozie classpath or in the lib directory.	
Reducer Class	Specifies the Java class that should be used as reducer.	The JAR containing the class must be available in the Oozie classpath or in the lib directory.	
No of Tasks	The default number of map tasks per job.	This setting is ignored if the mapred.job.tracker parameter is set to "local".	
Input Dir			
Output Dir	The directory to contain the job output.	Hadoop verifies that the job output directory does not exist, and then creates it when the job starts.	
Job XML	You can select one or more job.xml files to pass Map-Reduce configuration details.	The configuration file that specifies the variables used for the Map-Reduce action in the workflow. Can be overwritten or replaced by entries under the Configuration section.	

Table 7.33. MR Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

Table 7.34. MR Action, Advanced Properties Parameters

Parameter Name	Description	Additional Information	Example
Resource Manager	Master node that arbitrates all the available cluster resources among the competing applications.	The default setting is discovered from the cluster configuration.	${resourceManager}
Name Node	Manages the file system metadata.	Keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. Clients contact NameNode for file metadata or file modifications.	${nameNode}

Parameter Name	Description	Additional Information	Example
File	Select any files that you want to make available to the Map-Reduce action when the workflow runs.	Commonly used for streaming jobs that require files in HDFS to be available to the mapper/reducer scripts.	/path/file
Archive	Select any archives that you want to make available to the Map-Reduce action when the workflow runs.	Commonly used for streaming jobs that require files in HDFS to be available to the mapper/reducer scripts.	/path/archived-data-files
Prepare	Select **mkdir** or **delete** and identify any HDFS paths to create or delete before starting the job.	Use **delete** to do file cleanup prior to job execution. Enables Oozie to retry a job if there is a transient failure (the job output directory must not exist prior to job start). If the path is to a directory: **delete** deletes all content recursively and then deletes the directory. **mkdir** creates all missing directories in the path.	

Table 7.35. MR Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions. The configuration properties are loaded in the following order: streaming, job-xml, and configuration. Later values override earlier values.	

7.10. SSH Action Parameters

Starts a shell command on a remote machine as a remote secure shell running in the background.

Table 7.36. SSH Action, General Parameters

Parameter Name	Description	Additional Information	Example
Host	The user and the remote host containing the home directory in which the shell command will run.		user@host
Command	The shell command to execute.	Should be available on the remote machine and is executed in the user's home directory.	
Args	Identify any argument parameters to be passed to the shell script.	In arguments that contain white space, each entry is handled as a separate argument.	jdbc:mysql://wfm.openstacklocal/test
Arg	Identify any argument parameters to be passed to the shell script.	Used to handle arguments that include white space. Each value is handled as one argument.	
Capture Output	Indicates whether or not Oozie will capture output of the STDOUT of the SSH action execution.	Useful if the output of the SSH action needs to be available to the workflow job, such as for decision nodes.	

Parameter Name	Description	Additional Information	Example
		Command output must be in Java Properties file format if another action will pick it up.	

Table 7.37. SSH Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

7.11. Spark Action Parameters

Use Spark actions to handle batch and streaming data in your workflow.

Table 7.38. Spark Action, General Parameters

Parameter Name	Description	Additional Information	Example
Application Name	Name you want to assign to the Spark application.		
Application	The JAR or the Python script representing the Spark application. If a JAR is specified, you must provide the Fully Qualified Main class.	Path to a bundled jar including your application and all dependencies. The URL must be globally visible inside of your cluster, for instance, an hdfs:// path or a file:// path that is present on all nodes	Application: ${nameNode}/user/centos/spark-action/lib/oozie-examples.jar Class: org.apache.spark.myapp.MySparkApp
Runs On	YARN Cluster, YARN Client, Local, Custom	Distinguishes where the driver process runs. In cluster mode, the framework launches the driver inside of the cluster. In client mode, the submitter launches the driver outside of the cluster. In local mode, the application is run locally. Local mode is useful for debugging.	If you select YARN Cluster mode, the file must be on HDFS. For YARN Client mode, the file can be local or on HDFS. **Important:** The `yarn-client` execution mode for the Oozie Spark action is no longer supported and has been removed. Workflow Manager and Oozie continue to support `yarn-cluster` mode.
Spark Options		See Apache Spark configuration documentation	
Job XML	You can select one or more job.xml files to pass Java configuration elements.	The configuration file that specifies the variables used for the Java action in the workflow. Can be overwritten or replaced by entries under the Configuration section.	

Table 7.39. Spark Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

Table 7.40. Spark Action, Advanced Properties Parameters

Parameter Name	Description	Additional Information	Example
Resource Manager	Master node that arbitrates all the available cluster resources among the competing applications.	The default setting is discovered from the cluster configuration.	${resourceManager}
Name Node	Manages the file system metadata.	Keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. Clients contact NameNode for file metadata or file modifications.	${nameNode}
File	Select any files that you want to make available to the Spark action when the workflow runs.	Specify the name of the main JAR if you did not put your JAR files in the */lib* directory.	/path/file
Archive	Select any archives that you want to make available to the Spark action when the workflow runs.		/path/archived-data-files
Prepare	Select **mkdir** or **delete** and identify any HDFS paths to create or delete before starting the job.	Use **delete** to do file cleanup prior to job execution. Enables Oozie to retry a job if there is a transient failure (the job output directory must not exist prior to job start). If the path is to a directory: **delete** deletes all content recursively and then deletes the directory. **mkdir** creates all missing directories in the path.	${nameNode}/user/centos/output-data/spark
Arg	Identify any arguments for the Spark action.	Arguments to be passed to the main method of your main class. The value of each arg element is considered a single argument and they are passed to the main method in the same order. See Spark Applications documentation.	${nameNode}/user/username/input-data/text/data.txt ${nameNode}/user/username/output-data/spark

Table 7.41. Spark Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions.	

7.12. File System (FS) Action Parameters

Allows manipulation of files and directories in HDFS from a workflow application.

Table 7.42. FS Action, General Parameters

Parameter Name	Description	Additional Information	Example
Command	Options are mkdir, delete, move, chmod, touchz, chgrp	Commands are executed synchronously within the FS action, but the workflow job waits until the commands are completed before continuing to the next action.	delete
Path	The path to which the command will be applied.		hdfs://wfm.openstacklocal/user/test
Job XML	You can select one or more job.xml files to pass Java configuration elements.	The configuration file that specifies the variables used for the Java action in the workflow. Can be overwritten or replaced by entries under the Configuration section.	

Table 7.43. FS Action, Transition Parameters

Parameter Name	Description	Additional Information	Default Setting
Error To	Indicates what action to take if the action errors out.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to kill node, but can be changed.
OK To	Indicates what node to transition to if the action succeeds.	You can modify this setting in the dialog box or by modifying the workflow graph.	Defaults to the next node in the workflow.

Table 7.44. FS Action, Advanced Properties Parameters

Parameter Name	Description	Additional Information	Example
Name Node	Manages the file system metadata.	Keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. Clients contact NameNode for file metadata or file modifications.	${nameNode}

Table 7.45. FS Action, Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions.	

7.13. Submit Dialog Parameters

The Submit tab opens a dialog box that allows you to submit a workflow, a coordinator, or a bundle definition as a job. Once submitted, a job displays in the Workflow Manager dashboard.

Table 7.46. Submit Dialog, Configuration Parameters

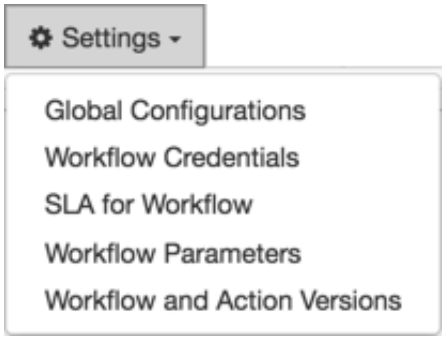
Parameter Name	Description	Additional Information	Example
Workflow Path	The HDFS location in which the workflow definition is stored.		/user/ambari-qa/ooziewfs/sqoop/workflow.xml

Parameter Name	Description	Additional Information	Example
Overwrite	When selected, if a file with the same name exists in the path, it is overwritten on submit.	If a file of the same name exists in the path and is not overwritten, you must change the file name or you get an error.	
Run on Submit	When selected, the job will run when it is submitted.	If not selected, a submitted job displays in the WFM dashboard with status Prep, and the job must be started manually.	
Use system lib path	When selected, allows WFM (Oozie) to use the libraries in the sharelib path.	Refer to HDFS Share Libraries for Workflow Applications for more information.	
Rerun on Failure	When selected, allows WFM to automatically rerun failed jobs.	This sets oozie.wf.rerun.failnodes=true in the configuration submitted to Oozie.	
Job Properties	Displays the actual names of the resource manager and name node being used for the job.		**ResourceManager example:** cluster-5-companyname.com:8050 **NameNode example:** hdfs://cluster-5-companyname.com:8020
Custom Job Properties	Provides key/value pair properties that are sent to Oozie when the job is submitted.	Allows you to override or supplement job property values set in a job.xml or in the Configuration section of the Action Settings dialog. For example, if a coordinator contains a workflow path with a variable (/user/${user}/workflow.xml), rather than an absolute path, WFM would not be able to determine the variable to be passed. In this case, you could provide the variable as a custom property.	Name: oozie.action.sharelib.for.sqoop Value: sqoop,hive,hcatalog or Name: ${user} Value: hive

8. Settings Menu Parameters

From the Settings menu, you can set various parameters related to the overall workflow or set parameters that can be applied as global settings across action nodes.

Figure 8.1. Settings menu



The available action types and their associated parameters for the Settings menu are described in the following:

Global Configuration Parameters [81]

Workflow Credentials Parameters [82]

SLA for Workflow Parameters [83]

Workflow Parameters [84]

Workflow and Action Versions [85]

8.1. Global Configuration Parameters

Use the Global Configuration settings to apply a common set of parameters to all the action nodes in the workflow. You can override the global settings for any node by changing the values in the node Action Settings dialog box.

Table 8.1. Global Configuration Parameters

Parameter Name	Description	Additional Information	Example
Resource Manager	Points to the YARN Resource Manager.	If the variable ${resourceManager} is used, WFM substitutes with the proper resource manager discovered from Ambari configurations.	${resourceManager} or http://sandbox.hortonworks.com:8050
Name Node	Points to the YARN name node.	If the variable ${nameNode} is used, WFM substitutes with the proper resource manager discovered from Ambari configurations.	
Job XML	Location of the job XML file that contains name/value combinations for the workflow.	Using a job XML file is useful when the same settings need to be used by multiple workflows.	

Parameter Name	Description	Additional Information	Example
Properties Name/ Value	Use to declare a variable and associated value for workflows.	Use when you need a specific setting to change for each job submission or when you need to configure the same value multiple times. The variable defined could be used in a workflow with the syntax ${variable1}.	Name: ${user} Value: hive

8.2. Workflow Credentials Parameters

You must set workflow credentials if you are running actions in a secured cluster. You set the credentials at the global level, then select the user when you create or edit an action node.

Some action types talk to external services such as HCatalog, HBase Region Server, and Hive Server 2. In these circumstances, extra configuration is required to authenticate. Kerberos credentials are used to obtain delegation tokens on behalf of the user from the external service.

Table 8.2. HCat Workflow Credentials Parameters

Parameter Name	Description	Additional Information	Example
Name	The name used to identify the credentials.		hive-cred
Type	Options are HCat, Hive2, and HBase.		
HCat Metastore Principal	The name of the Kerberos principal to be used for HCatalog.		
HCat Metastore URL	This is a Thrift URI used to make metadata requests to a remote Metastore, allowing multiple Hive clients to connect to a remote service.		The same value as the Ambari setting: Hive service>Configs tab>Advanced tab>General section>hive.metastore.uris field
Custom Properties Name/ Value			

Table 8.3. Hive2 Workflow Credentials Parameters

Parameter Name	Description	Additional Information	Example
Name	The name used to identify the credentials.		hive-cred
Type	Options are HCat, Hive2, and HBase.		
Hive2 JDBC URL	The JDBC connection location for interacting with the database.		Hive>Configs>Advanced>Hive Metastore>Database URL
Hive2 Server Principal	The name of the Kerberos principal to be used for Hive Server 2.		
Custom Properties Name/ Value			

Table 8.4. HBase Workflow Credentials Parameters

Parameter Name	Description	Additional Information	Example
Name	The name used to identify the credentials.		hive-cred
Type	Options are HCat, Hive2, and HBase.		
Hadoop Security Auth	Possible values are simple (no authentication), and kerberos.		
HBase Security Auth	Possible values are simple (no authentication), and kerberos.	Authentication type for HBase security. For the client to be able to communicate with the cluster, the hbase.security.authentication in the client- and server-side site files must match.	HBase>Configs>Advanced>hbase-site.xml
HBase Master Kerberos Principal	The Kerberos principal name that should be used to run the HMaster process.	The Kerberos principal name that should be used to run the HRegionServer process.	
HBase Regionserver Kerberos Principal	A list of servers that are part of the ZooKeeper quorum for running in replicated mode.		
HBase ZooKeeper Quorum	A list of ZooKeeper server addresses, separated by commas, that are to be used by the ZKFailoverController in automatic failover.		
Hadoop RPC Protection	A comma-separated list of protection values for secured SASL connections. Possible values are authentication, integrity, and privacy.	Authentication = authentication only, no integrity or privacy. Integrity = authentication and integrity are enabled. Privacy = authentication, integrity, and privacy are enabled.	
HBase RPC Protection	A comma-separated list of protection values for secured SASL connections. Possible values are authentication, integrity, and privacy.		
Custom Properties Name/ Value			

8.3. SLA for Workflow Parameters

You can set Service Level Agreement (SLA) parameters within most action node configurations or you can set them from the Settings menu so they are applied to all workflows. Settings in the action node dialog box override the global settings.

Enabling SLA results in alerts being generated whenever an SLA time is missed. You can also have alerts sent as email notifications. SLA settings do not impact whether or not a job runs or completes. If a job misses SLA and you want the job to stop, you must manually stop the job.

If SLA is defined for a workflow action that is not in the execution path because of a decision node, you get an SLA_MISS notification.

For more information, see Apache Oozie SLA Monitoring.

Table 8.5. SLA for Workflows Parameters

Parameter Name	Description	Additional Information	Example
Enabled	Enables or disables the SLA settings.	After enabling SLA, you must, at minimum, set the Nominal Time and the Should End time to receive email notifications if those times are missed by the job.	
Nominal Time	The ideal time you want the job to start.	This is the time relative to which your jobs' SLAs are calculated. Must be set to a future time or date. The date and time fan be set by using the calendar feature or by using EL expressions.	Calendar date: 3:00 AM on 22 August 2017, IST timezone.
Should Start	The amount of time (in minutes, hours, or days) within which a job should start to meet SLA.	"Should Start" time is relative to the nominal time. Optional setting. The date and time fan be set by using the calendar feature or by using a variable. Can also be set as a cron job.	If the nominal start time is 3:00 AM IST on 22 August 2017 and the job should start within 20 minutes of the nominal start time to meet SLA, set "Should Start" to 3:20 AM, 22 August.
Should End	The amount of time (in minutes, hours, or days) within which a job should finish to meet SLA.	"Should End" time is relative to the nominal time. The date and time fan be set by using the calendar feature or by using a variable. Can also be set as a cron job.	If the nominal start time is 3:00 AM IST on 22 August 2017 and the job should end within 90 minutes of the ideal start time to meet SLA, set "Should End" to 4:30 AM, 22 August.
Max Duration	The maximum amount of time (in minutes, hours, or days) that you expect the job to run.	Optional setting. The date and time fan be set by using the calendar feature or by using a variable. Can also be set as a cron job.	If the nominal start time is 3:00 AM IST on 22 August 2017 and you expect the job to end within 90 minutes of the ideal start time, set "Max Duration" to 4:30 AM, 22 August.
Alert Events	You can select to have alerts sent if the job misses its start, end, or duration times.	Optional setting.	
Alert Emails	The email addresses to which alerts should be sent.	Optional setting.	

8.4. Workflow Parameters

From the Workflow Parameters dialog box, you can define a common set of variables at a global level, that you can reference from action nodes in the workflow. The parameters are entered as a key/value pair.

You can set workflow parameters within most action node configurations or you can set them from the Settings menu so they are applied to all workflows. Settings in the action node dialog box override the global settings.

To create global parameters, you must enter a name (key), but a value is optional. However, if the value is not set at the global level, it must be entered at the node level.

Table 8.6. Workflow Configuration Parameters

Parameter Name	Description	Additional Information	Example
Name and Value	The name/value pair can be used instead of a job.xml file or can override parameters set in the job.xml file.	Used to specify formal parameters. If the name and value are specified, the user can override the values from the Submit dialog box. Can be parameterized (templatized) using EL expressions. See the Apache documentation for more information.	

8.5. Workflow and Action Versions

You can set or change the schema version for each action or workflow from the Workflow and Action Versions dialog box. By default, the latest version is used. However, you might need to select an earlier software version if you import workflows that use earlier software.

You can select versions for the following components:

Hive
Hive2
Sqoop
Shell
Spark
DistCp
Email

8.6. How SLA Works

Nominal time: The nominal time specifies the time when something should happen. In theory the nominal time and the actual time should mach, however, in practice due to delays the actual time may occur later than the nominal time.

9. Job States

You can view the state of any job from the Workflow Manager dashboard, to determine whether the job started, succeeded, failed, and so forth.

A workflow, coordinator, or bundle job can be in any of the following states:

Prep
: When a workflow, coordinator, or bundle job is first submitted, it is in the Prep state, meaning the job is defined and saved, and a unique ID is assigned to it, but the job is not running.

  Jobs can transition from Prep to Running or Killed states.

Running
: When a workflow, coordinator, or bundle job is executed, it goes into the Running state. A job remains in the Running state as long as it does not reach its end state, does not end in error, or is not suspended.

  Jobs can transition from Running to Succeeded, Suspended, Killed, or Failed states.

Suspended
: A running workflow, coordinator, or bundle job can be suspended. The job remains in Suspended state until the job is resumed or is killed.

  Jobs can transition from Suspended to Running or Killed states.

Succeeded
: A running workflow, coordinator, or bundle job transitions to the Succeeded state when it reaches the end node. it final state.

  Succeeded is a final state, so it does not transition to other states.

Killed
: When a workflow, coordinator, or bundle job is killed manually or by reaching a kill node in the workflow, the job transitions to the Killed state.

  Killed is a final state, so it does not transition to other states.

Failed
: When a running workflow, coordinator, or bundle job fails due to an unexpected error, it transitions to the Failed state.

  Failed is a final state, so it does not transition to other states.

More Information

Monitoring Jobs Using the Dashboard

10. Workflow Manager Files

Several files are created during the process of designing a workflow. Other files must be created and available in a specified path in order to run a workflow job.

The following files are associated with workflow jobs:

- `workflow.xml`

  This file is the definition of the workflow application. It is created when you input and save the workflow parameters. The file is saved to whatever location you specify in HDFS. The default file name is workflow.xml, but you can provide an alternative name when you initially save the file.

- `workflow.wfdraft`

  This is a draft file created when you save a non-validated workflow. The file is saved to the same location in HDFS as the associated `workflow.xml` file. The default file name is `workflow.wfdraft`, but you can provide an alternative name when you initially save the file.

- `/appdir/config-default.xml`

  This file is in HDFS. It is optional and contains properties shared by all workflows.

- `/appdir/lib/files.jar`

  In HDFS

- Each workflow can also have a `job.properties` file (not put into HDFS) for job-specific properties.

The following files are associated with coordinator jobs:

- `coordinator.xml`

  This file is the definition of the coordinator application. It is created when you input and save the coordinator parameters. The file is saved to whatever location you specify in HDFS. The default file name is `coordinator.xml`, but you can provide an alternative name when you initially save the file.

- `coordinator.wfdraft`

  This is a draft file created when you save a non-validated coordinator. The file is saved to the same location in HDFS as the associated `coordinator.xml` file. The default file name is `coordinator.wfdraft`, but you can provide an alternative name when you initially save the file.

- `coordinator.properties`

  For defining the job's properties.

The following files are associated with bundle jobs:

- `bundle.xml`

  This file is the definition of the bundle application. It is created when you input and save the bundle parameters. The file is saved to whatever location you specify in HDFS. The default file name is `bundle.xml`, but you can provide an alternative name when you initially save the file.

- `bundle.wfdraft`

  This is a draft file created when you save a non-validated bundle. The file is saved to the same location in HDFS as the associated `bundle.xml` file. The default file name is `bundle.wfdraft`, but you can provide an alternative name when you initially save the file.

The following log files can assist with troubleshooting issues:

- The `wfmanager-view.log` found under the `ambari-server` log directory.

  Jobs successfully submitted to WFM so they have a job ID will have logs displayed in the WFM details tabs. The `wfmanager-view.log` provides additional information.

- Oozie logs, which are located at `/var/log/oozie` by default.

  The location of the log file can be changed. If an exception occurs before successfully submitting a workflow, coordinator, or bundle job, a good place to debug is the Oozie logs.