

Workflow Summary Version 4.0

- [Introduction](#)
 - [Summary Table :](#)
 - [Workflow wall time :](#)
 - [Workflow cumulative job wall time :](#)
 - [Cumulative job walltime as seen from submit side:](#)
- [Summary Table Content](#)
 - [Legends](#)
 - [How to calculate the values](#)
- [Examples](#)
 - [Case 1](#)
 - [Case 2](#)
 - [Case 3](#)
- [Queries](#)
 - [Query for finding all the descendant workflows of the root workflow](#)
 - [Total jobs](#)

Introduction

Workflow summary is summary of the statistics information of the workflow that is shown on the command line output . It recursively parses the sub workflow to generate the statistics information.

Summary Table :

The summary table contains the total number of jobs/tasks/sub workflows, total number of succeeded jobs/tasks/sub workflows etc. The detailed description of elements is given below.

Workflow wall time :

The walltime from the start of the workflow execution to the end as reported by the DAGMAN.In case of rescue dag the value is the cumulative of all retries.

Workflow cumulative job wall time :

The sum of the walltime of all jobs as reported by kickstart. In case of job retries the value is the cumulative of all retries. For workflows having sub workflow jobs (i.e SUBDAG and SUBDAX jobs), the walltime value includes jobs from the sub workflows as well. The value is obtained from the *remote_run time* in the **invocation** table and is multiplied by the **multiplier_factor** in the **job_instance** table.

Cumulative job walltime as seen from submit side:

The sum of the walltime of all jobs as reported by DAGMan. This is similar to the regular cumulative job walltime, but includes job management overhead and delays. In case of job retries the value is the cumulative of all retries. For workflows having sub workflow jobs (i.e SUBDAG and SUBDAX jobs), the walltime value includes jobs from the sub workflows as well. The value is obtained from the *local_duration* in the **job_instance** table and is multiplied by the **multiplier_factor**

Summary Table Content

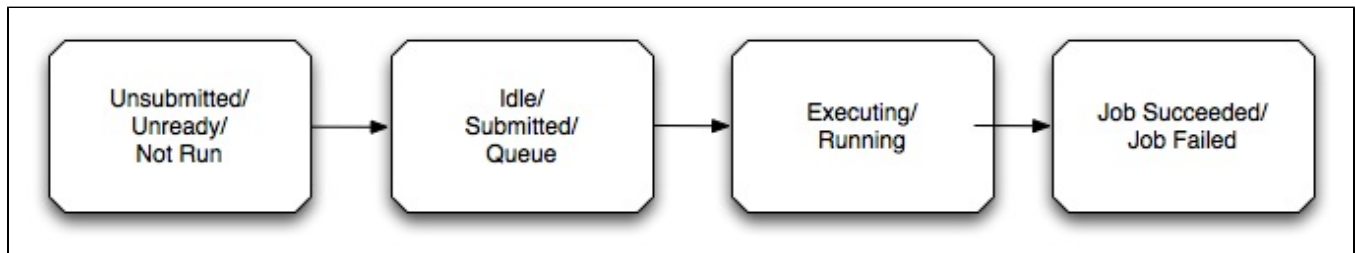
Workflow Summary contains the following information.

The summary statistics table format is given below.

	Succeeded	Failed	Incomplete	Total		Retries	Total Run
Tasks							
Jobs							
Sub Workflows							

Legends

The diagram below shows the state diagram of a job instance.



Succeeded :-

Tasks - Number of succeeded tasks .
Jobs -Number of succeeded jobs.
Sub Workflows - Number of succeeded sub workflows .

Failed :-

Tasks - Number of failed tasks.
Jobs -Number of failed jobs.
Sub Workflows - Number of failed sub workflows.

Incomplete :-

Tasks - Number of tasks that are not in succeeded or failed state.
Jobs- Number of jobs that are not in succeeded or failed state.
Sub Workflows - Number of sub workflows that are not in succeeded or failed state.

Total :-

Tasks - Total number of abstract tasks.
Jobs -Total number of planned jobs.
Sub Workflows - Total number of sub workflows.

Retries :-

Tasks - Total number of task retries.
Jobs - Total number of job retries.
Sub Workflows - Total number of sub workflow retries.

Total Run :-

Tasks - Total number of invocations that were executed during workflow run. This value is a cumulative of retries , if a task was executed more than once.
Jobs -Total number of job instances that where executed during workflow run.This value is a cumulative of retries , if a job was instantiated more than once.
Sub Workflows - Total number of sub workflows that where executed during workflow run. This value is a cumulative of retries , if a sub workflow was executed more than once.

How to calculate the values

Jobs is any entry in the job table with type_desc not (dax or dag)
 Task is any entry in the task table with type_desc not (dax or dag)
 Sub workflow is any entry the job table or task table with type_desc (dax or dag)

Succeeded :-

Tasks - Total count of all task last retry with exitcode ==0 in invocation table
Jobs - Total count of all job last retry with last state in jobstate table either JOB_SUCCESS or POST_SCRIPT_SUCCESS state
Sub Workflows - Total count of all sub workflow last retry with last state in jobstate table either 'JOB_SUCCESS' or 'POST_SCRIPT_SUCCESS' state

Failed :-

Tasks - Total count of all task last retry with exitcode <> 0 in invocation table
Jobs -Total count of all job last retry with jobstate table having 'PRE_SCRIPT_FAILED','SUBMIT_FAILED','JOB_FAILURE' ,'POST_SCRIPT_FAILED' state
Sub Workflows - Total count of all sub workflow last retry with jobstate table having 'PRE_SCRIPT_FAILED','SUBMIT_FAILED','JOB_FAILURE' ,'POST_SCRIPT_FAILED' state

Incomplete :-

Tasks -Total tasks - (Failed tasks + Succeeded tasks).
Jobs- Total jobs-(Failed job count + Succeeded Job count)
Sub Workflows- Total sub workflows -(Failed sub workflow count + Succeeded sub workflow count)

Total :-

Tasks - Total count of tasks in the task table
Jobs - Total count of jobs in the job table.
Sub Workflows - Count of sub workflows in the job table

Retries :-

Task - This will be difference between the total number of tasks in the invocation table and total number of unique instance in invocation .
Jobs - This will be difference between the total number of job instances in the job_instance table and total number of unique job instance in job_instance table.
Sub Workflows - This will be difference between the total number of sub workflow instances in the jobinstance table and total number of unique sub workflow instance in jobinstance table .

Examples**Case 1**

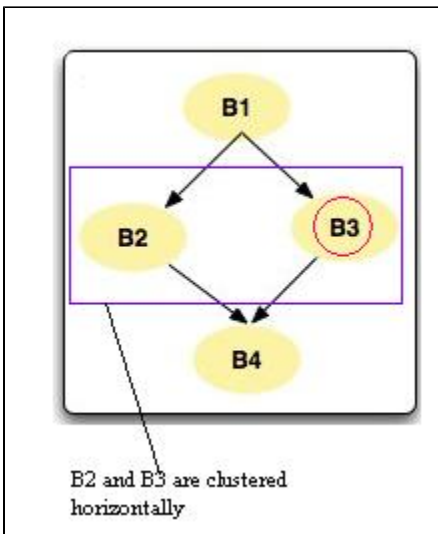


Figure 1 :- Diamond workflow [Failed Run]

The example in Figure 1 is a diamond workflow with 4 tasks in the dag. And the pegasus plan creates three jobs in the dag with B2 and B3 clustered. During the execution of the workflow the clustered job fails after 3 retries. For the clustered job B2 runs, but B3 fails in all the retries.

	Succeeded	Failed	Incomplete	Total		Retries	Total Runs
Tasks	2	1	1	4		4	7
Jobs	1	1	1	3		2	4
Sub Workflows	0	0	0	0		0	0

Table 1.1 : Summary statistics table

Note: For the sake of simplicity Jobs row consider only compute jobs.

Case 2

HIERARCHAL WORKFLOWS

INCREASING LEVEL OF RECURSION 

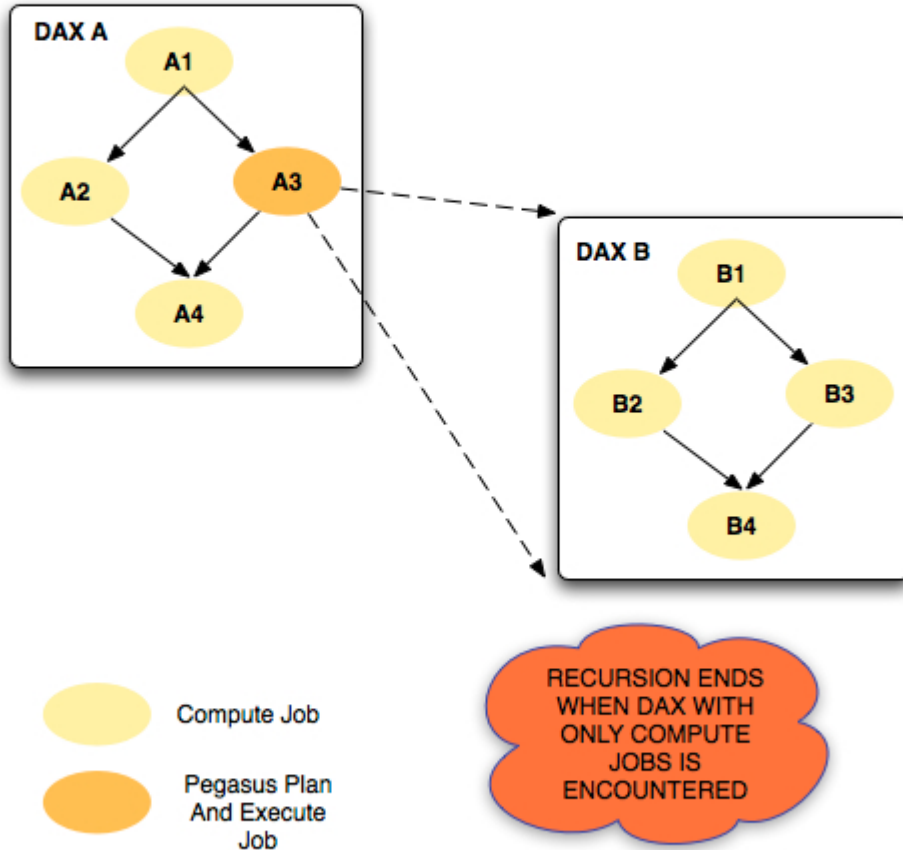


Figure 2 :- Hierarchal Workflow [Successful Run]

The example in Figure 2 is a hierarchical work flow with 4 tasks in DAX A and 4 tasks in DAX B. A3 is sub workflow task.

	Succeeded	Failed	Incomplete	Total	Retries	Total Run
Tasks	7	0	0	7	0	7
Jobs	7	0	0	7	0	7
Sub workflow	1	0	0	1	0	1

Table 2.1 : Summary statistics table

Note: For the sake of simplicity Jobs row consider only compute jobs.

Case 3

HIERARCHAL WORKFLOWS

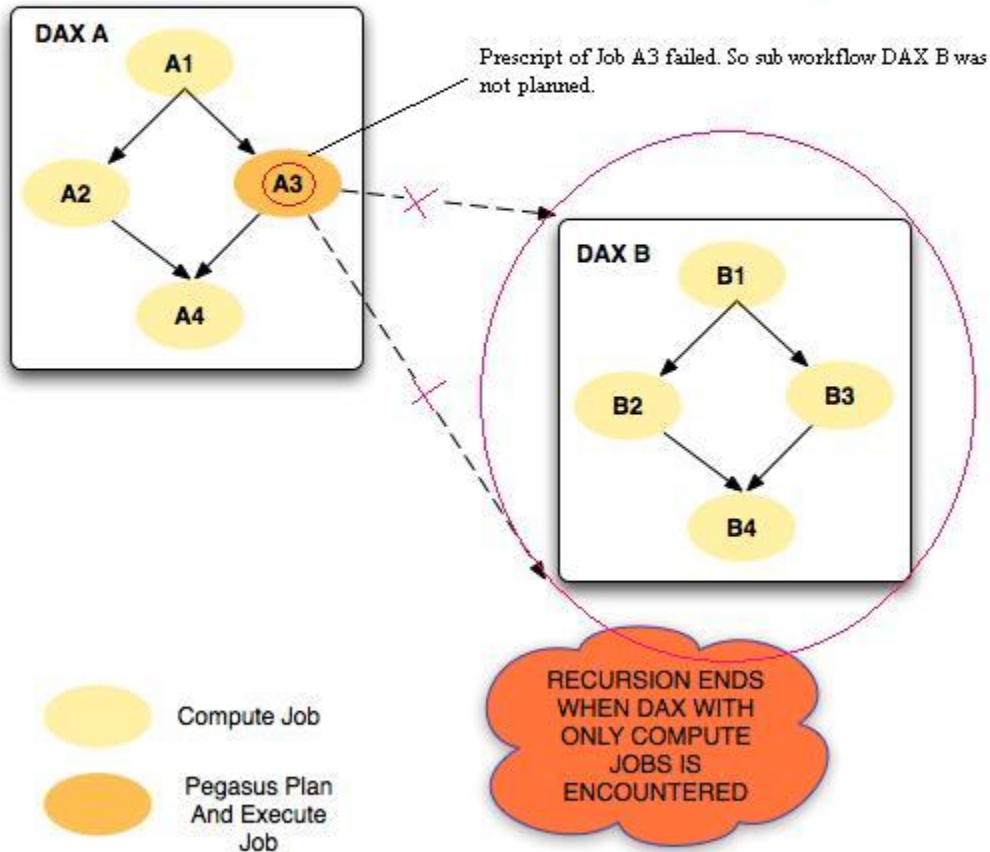


Figure 3 Hierarchal workflow [Failed Run]

The example in Figure 3 is a hierarchical work flow with 4 tasks in DAX A and 4 tasks in DAX B. However the A3 sub workflow tasks fails at the Prescript which results in DAX B workflow not getting planned . So the database is not populated with DAX B workflow details. A3 fails after 3 retries.

	Succeeded	Failed	Incomplete	Total		Retries	Total Run
Tasks	2	0	1	3		0	2
Jobs	2	0	1	3		0	2
Sub Workflows	0	1	0	1		2	3

Table 3.1 Summary statistics table

Note: For the sake of simplicity Jobs row consider only compute jobs.

Queries

Query for finding all the descendant workflows of the root workflow

```
// API method name :get_descendant_workflow_ids()

SELECT wf_id, wf_uuid FROM
workflow as wf WHERE
wf.root_wf_id = 1 and
wf.wf_id <> 1
```

Total jobs

```
// API method name: get_total_jobs_status() expand_workflow = true, job_filter = all

SELECT count(j.job_id)
```