

DIRAC Workload Management System.

Job States.

Jobs in the DIRAC Workload Management System pass a number of states between the first job submission and its final completion. The job states are described by 2 values: major and minor job state. Some of the states are very short living and can not be used for monitoring, they are still defined to be available in the job logging records. All these states are recorded by the DIRAC logging system and can be consulted later for problem solving purposes. All the states are recorded together with their time stamps.

The major and minor job states are defined by various DIRAC components: central WMS components, agents, job wrappers. Minor states can be defined also by the job applications. The state values and transitions from one state to another are defined by convention and are not enforced by any “State Machine” engine. In particular, failure to set some state will result in losing this state in the logging records.

All the possible values with explanation of the conditions necessary for entering into a state are described in the following table. The major states are consistently named as past or present participles to reflect either an achieved or currently active states.

<i>Major State</i>	<i>Minor State</i>	<i>Comment</i>
<i>reserved</i>	<i><default state></i>	The job ID is reserved by the Workload Management system.
<i>received</i>	<i><default state></i>	The job is received by the system and entered into the Job Database.
	<i>Waiting input data</i>	The job is being processed by an Optimizer Data before placing into the Task Queue.
	<i>Waiting prioritization</i>	The job is being processed by an Optimizer Prioritizer before placing into the Task Queue.
<i>waiting</i>	<i><default state></i>	The job is placed into the Task Queue and is waiting there for further WMS stages .

<i>Major State</i>	<i>Minor State</i>	<i>Comment</i>
	<i>Pilot Agent Submission</i>	The job is a subject for the Pilot Agent execution schema and is waiting for a successful Pilot Agent submission.
	<i>Pilot Agent Response</i>	The job is waiting for the response from the submitted Pilot Agent.
	<i>Proxy expired</i>	The job is stuck indefinitely in the Task Queue because of the expired proxy. It will pass to the <default state> if the proxy will be renewed.
<i>matched</i>	<default state>	The job is matched and assigned to a Job Agent, e.g. Pilot Agent. The state is set by the Match Maker.
	<i>Job received by agent</i>	The job is received by the Job Agent. The state is set by the Job Agent.
	<i>Installing Software</i>	The Job Agent is checking and/or installing the necessary application software.
	<i>Rescheduling job</i>	The Job Agent failed to prepare the job environment or to submit the job and sent the <i>rescheduling</i> command.
	<i>Job prepared to submit</i>	Job Agent has generated the Job Wrapper and is ready to submit the job to the Computing Element.
<i>scheduled</i>	<i>Waiting CE response</i>	The job is sent to the local Computing Element. Waiting for the confirmation of the CE.
<i>queued</i>	<default state>	The job is successfully entered into the local batch queue. The state is set by the CE.
<i>running</i>	<i>Starting DIRAC job</i>	The job has started and is running on the Worker Node. The state is set by the Job Wrapper.

<i>Major State</i>	<i>Minor State</i>	<i>Comment</i>
	<i>Getting input data</i>	The Job Wrapper is resolving/downloading input data – Input Sandbox and Data.
	<i>Failed to get input sandbox</i>	The state is set after a failure to get the job's Input Sandbox before sending <i>rescheduling</i> command
	<i>Failed to get input data</i>	The state is set after a failure to get the job's Input Data before sending <i>rescheduling</i> command
	<i>Starting the application</i>	The Job Wrapper sets this state before passing the control to the job's executable.
	<i><Application specific states></i>	The job application can define any state it deems necessary to follow its progress.
	<i>Application stalled</i>	The Job Wrapper watchdog process detected that the application is not consuming CPU for a predefined period of time.
	<i>Application finished successfully</i>	The job executable returned without an error code
	<i>Application finished with errors</i>	The job executable returned with an error code
	<i>Sending output data</i>	The Job Wrapper is sending the application output – Output Sandbox and Data.
<i>stalled</i>		The job failed to send heart-beats for a predefined period of time.
<i>done</i>		The Job Wrapper finished execution without non-application errors.

<i>Major State</i>	<i>Minor State</i>	<i>Comment</i>
	<i>Job finished successfully</i>	The application and all the finishing operations were successful. No asynchronous operations are requested
	<i>Job finished with errors</i>	Application or some of the finishing operations failed. In this case a more specific error message is sent as the job parameter.
	<i>Waiting for data transfer</i>	The asynchronous data transfer request was set and is being processed.
	<i>Data transfered successfully</i>	The asynchronous data transfer is completed to all the requested destinations including the catalog registration
<i>ready</i>		Final nonfailed job execution state.
	<i>Output retrieved</i>	The job output sandbox was retrieved. This normally a sign that the job can be purged from the system.
<i>failed</i>	<i><Reason for failure></i>	The job has failed in an unrecoverable manner. Different failure reasons are set as a minor state. From this state the job can be either rescheduled manually or deleted from the system.
	<i>Output retrieved</i>	The failed job output sandbox was retrieved. This normally a sign that the job can be purged from the system.
<i>rescheduled</i>		The job is rescheduled. Its proxy is renewed. Otherwise it is equivalent to the <i>received</i> state.
<i>killed</i>		The job received <i>kill</i> command from the user.
	<i>Kill signal sent</i>	The kill signal is sent to the job. This needs implementation still

<i>Major State</i>	<i>Minor State</i>	<i>Comment</i>
	<i>Job killed</i>	The job is definitely killed on the execution system.
<i>deleted</i>		The job has been marked for deletion and will be garbage collected by the Cleaning Agent as soon as possible. The state is usually set by the Accounting Agent or manually by a user.

Remarks

<default state> for minor states is “Unknown”. This can be further specified for some of the major states.

More minor state can be added as necessary, for example, after addition of new optimizers or applications. The major states should be frozen.