

When the WM starts, it may find a number of requests sitting in its input queue. They can either represent new requests that have arrived during its downtime or old requests that might already have been processed, at least partially. The WM needs then to tell requests that still need to be processed from requests that have already been delivered.

Before doing so, the WM needs to aggregate all the requests concerning the same job and find a "summary" of what to do with them.

The algorithm applied to a submit or resubmit request of a single job is:

- if the request is in the WAITING status, it needs to be processed
- if the request is in the READY status
  - ◆ if there is an EnQueued START event from the WM towards the JC, the request is kept in a limbo and its status or its events further checked later
- if the status of the request is not retrievable, the request is kept in a limbo and its status further checked later
- if the status of the request is Done FAILED and the last event is EnQueued START, the request is kept in a limbo and its status or events further checked later
- in all other cases the request is ignored, in particular if there is an EnQueued OK event from the WM or a DeQueued event from the JC, following the Match event

With the introduction of collections, a resubmit request can be related to a job in a collection, so additional checks are needed.

Then there are collections as such. The algorithm applied to a submit request of a collection is:

At the moment there is no such concept as resubmission of a collection.

-- FrancescoGiacomini - 24 Sep 2007

---

This topic: EgeeJra1It > WmRecovery

Topic revision: r1 - 2007-09-24 - FrancescoGiacomini



Copyright © 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback