Web Service: VishnuTMSService - Generated by wsdl-viewer.x

Table of Contents

Web Service: VishnuTMSService

Web Service: VishnuTMSService

Target Namespace:
urn:ResourceProxy

Port VishnuTMSPort Port typeSource code

Location:
http://127.0.0.1:8080/ResourceProxy/VishnuTMS
Protocol:
SOAP
Default style:
document
Transport protocol:
SOAP over HTTP
Operations:

1. *cancelJob*DetailSource code
2. *getCompletedJobsOutput*DetailSource code
3. *getJobInfo*DetailSource code
4. *getJobOutput*DetailSource code
5. *getJobProgress*DetailSource code
6. *listJobs*DetailSource code
7. *listQueues*DetailSource code
8. *submitJob*DetailSource code

Operations

Port type VishnuTMSPortTypeSource code

1. #### cancelJob

   Source code
   Description:
   cancels a job from its id. If job id is equal to all, all submitted jobs by all users will be cancelled if the user is an administrator, and only jobs submitted by the user will be cancelled if the user is not an administrator.
   Operation type:
   Request-response. The endpoint receives a message, and sends a correlated message.
   Input:
   cancelJobInput (soapbind:body, use = literal)Source code

parameters type *cancelJobRequest*

- sessionKey type *string*
  The session key
- machineId type *string*
  Is the id of the machine on which the job is running
- jobId type *string*
  The Id of the job

Output:
cancelJobOutput (soapbind:body, use = literal)Source code
parameters type *cancelJobResponse*
Fault:
BATCH_SCHEDULER_ERRORMessage (documentation, use = literal)Source code
fault type *BATCH_SCHEDULER_ERRORFault*
Fault:
PERMISSION_DENIEDMessage (documentation, use = literal)Source code
fault type *PERMISSION_DENIEDFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
UNKNOWN_BATCH_SCHEDULERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_BATCH_SCHEDULERFault*
Fault:
UNKNOWN_MACHINEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_MACHINEFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
ALREADY_CANCELEDMessage (documentation, use = literal)Source code
fault type *ALREADY_CANCELEDFault*
Fault:
ALREADY_TERMINATEDMessage (documentation, use = literal)Source code
fault type *ALREADY_TERMINATEDFault*
Fault:
SSHMessage (documentation, use = literal)Source code
fault type *SSHFault*

2. getCompletedJobsOutput

Source code
Description:
gets standard output and error output files of completed jobs (applies only once for each job)
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
getCompletedJobsOutputInput (soapbind:body, use = literal)Source code
parameters type *getCompletedJobsOutputRequest*

- sessionKey type *string*
  The session key
- machineId type *string*
  Is the id of the machine on which the jobs are been submitted
- outDir - optional; type *string*
  Specifies the output directory where the files will be stored (by default, the current directory).

Output:
getCompletedJobsOutputOutput (soapbind:body, use = literal)Source code
parameters type *getCompletedJobsOutputResponse*

- data - optional;
  - jobresult - optional, unbounded, nillable;
    - jobId - optional; type *string*
      Represents the id of the job.
    - outputPath - optional; type *string*
      Is the path to the job output results.
    - errorPath - optional; type *string*
      Is the path to the file containing errors occured during job's execution.
- nbJobs type *integer*
  Is the number of jobs.

Fault:
BATCH_SCHEDULER_ERRORMessage (documentation, use = literal)Source code
fault type *BATCH_SCHEDULER_ERRORFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
PERMISSION_DENIEDMessage (documentation, use = literal)Source code
fault type *PERMISSION_DENIEDFault*
Fault:
UNKNOWN_BATCH_SCHEDULERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_BATCH_SCHEDULERFault*
Fault:
UNKNOWN_MACHINEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_MACHINEFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:

DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
SSHMessage (documentation, use = literal)Source code
fault type *SSHFault*

3. # getJobInfo

Source code
Description:
gets information on a job from its id
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
getJobInfoInput (soapbind:body, use = literal)Source code
parameters type *getJobInfoRequest*

- sessionKey type *string*
  The session key
- machineId type *string*
  Is the id of the machine on which the job is running
- jobId type *string*
  The id of the job

Output:
getJobInfoOutput (soapbind:body, use = literal)Source code
parameters type *getJobInfoResponse*

- sessionId type *string*
  Is the id of the session that contained the job submission command
- submitMachineId type *string*
  Is the id of the machine on which the job has been submitted.
- submitMachineName type *string*
  Is the name of the machine on which the job has been submitted.
- jobId type *string*
  Represents the id to job.
- jobName type *string*
  Represents the name assigned to the job.
- jobPath type *string*
  Is the path to the file containing job characteristics.
- outputPath type *string*
  Is the path to the job output results.
- errorPath type *string*
  Is the path to the file containing errors occured during job's execution.
- jobPrio type *JobPriority* - type *string* with restriction - enum { 'UNDEFINED', 'VERY_LOW', 'LOW', 'NORMAL', 'HIGH', 'VERY_HIGH' }
  Represents the job priority.
- nbCpus type *integer*
  Is the number of cpu per node used by the job.

- jobWorkingDir type *string*
  Indicates the directory where the job has been launched.
- status type *JobStatus* - type *string* with restriction - enum { 'UNDEFINED', 'SUBMITTED', 'QUEUED', 'WAITING', 'RUNNING', 'TERMINATED', 'CANCELLED', 'ALREADY_DOWNLOADED' }
  The current status of the job.
- submitDate type *long*
  Date and time when job was submitted (unix timestamp)
- endDate type *long*
  Represents the execution end date of the job (unix timestamp)
- owner type *string*
  Represents the job owner.
- jobQueue type *string*
  Is the name of the queue or class associated to the job.
- wallClockLimit type *long*
  Is the maximum wall-clock time during which the job can run (in seconds)
- groupName type *string*
  Represents the job owner group name.
- jobDescription type *string*
  Is the textual description of the job.
- memLimit type *integer*
  Represents the memory size limit of the job ((in MegaBytes).
- nbNodes type *integer*
  Is the total number of nodes used by the job.
- nbNodesAndCpuPerNode type *string*
  Is the number of nodes and processors per node used by the job (in the format nbNodes:nbCpuPerNode).

Fault:
BATCH_SCHEDULER_ERRORMessage (documentation, use = literal)Source code
fault type *BATCH_SCHEDULER_ERRORFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
PERMISSION_DENIEDMessage (documentation, use = literal)Source code
fault type *PERMISSION_DENIEDFault*
Fault:
UNKNOWN_BATCH_SCHEDULERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_BATCH_SCHEDULERFault*
Fault:
UNKNOWN_MACHINEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_MACHINEFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:

DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*

4. # getJobOutput

Source code
Description:
gets standard output and error output files of a job given its id
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
getJobOutputInput (soapbind:body, use = literal)Source code
parameters type *getJobOutputRequest*

- sessionKey type *string*
  The session key
- machineId type *string*
  gets outputPath and errorPath of a job from its id
- jobId type *string*
  The Id of the job
- outDir - optional; type *string*
  The output directory where the files will be stored (default is current directory)

Output:
getJobOutputOutput (soapbind:body, use = literal)Source code
parameters type *getJobOutputResponse*

- jobId type *string*
  Represents the id of the job.
- outputPath type *string*
  Is the path to the job output results.
- errorPath type *string*
  Is the path to the file containing errors occured during job's execution.

Fault:
BATCH_SCHEDULER_ERRORMessage (documentation, use = literal)Source code
fault type *BATCH_SCHEDULER_ERRORFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
PERMISSION_DENIEDMessage (documentation, use = literal)Source code
fault type *PERMISSION_DENIEDFault*
Fault:
UNKNOWN_BATCH_SCHEDULERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_BATCH_SCHEDULERFault*
Fault:
UNKNOWN_MACHINEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_MACHINEFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:

DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
JOB_IS_NOT_TERMINATEDMessage (documentation, use = literal)Source code
fault type *JOB_IS_NOT_TERMINATEDFault*
Fault:
SSHMessage (documentation, use = literal)Source code
fault type *SSHFault*
Fault:
ALREADY_DOWNLOADEDMessage (documentation, use = literal)Source code
fault type *ALREADY_DOWNLOADEDFault*

5. # getJobProgress

Source code
Description:
gets the progression status of jobs
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
getJobProgressInput (soapbind:body, use = literal)Source code
parameters type *getJobProgressRequest*

- sessionKey type *string*
  The session key
- machineId type *string*
  Is the id of the machine to get the jobs progression
- jobId - optional; type *string*
  Specifies the id of the job whose progression the user wants to see.
- jobOwner - optional; type *string*
  Specifies the owner of the job.

Output:
getJobProgressOutput (soapbind:body, use = literal)Source code
parameters type *getJobProgressResponse*

- data - optional;
  - progression - optional, unbounded, nillable;
    - jobId - optional; type *string*
      Represents the job id.
    - jobName - optional; type *string*
      Represents the job name.
    - wallTime - optional; type *integer*
      Represents the job wall time.
    - startTime - optional; type *long*
      Start date and time of the job (unix timestamp)

- endTime - optional; type *long*
  End date and time of the job (unix timestamp)
- percent - optional; type *integer*
  Represent the job progression.
- status - optional; type *JobStatus* - type *string* with restriction - enum { 'UNDEFINED', 'SUBMITTED', 'QUEUED', 'WAITING', 'RUNNING', 'TERMINATED', 'CANCELLED', 'ALREADY_DOWNLOADED' }
  Represents the job status.

♦ nbJobs type *integer*
Represents the number of jobs in progression list.

Fault:
UNKNOWN_MACHINEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_MACHINEFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*

6. # listJobs

Source code
Description:
gets a list of all submitted jobs on a machine. If machine identifier is equal to all, submitted jobs on all machines are listed
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
listJobsInput (soapbind:body, use = literal)Source code
parameters type *listJobsRequest*

♦ sessionKey type *string*
The session key
♦ machineId type *string*
Is the id of the machine on which the jobs are running
♦ jobId - optional; type *string*
lists the job with the specified id
♦ nbCpu - optional; type *integer*
lists the jobs with the specified number of CPUs per node
♦ fromSubmitDate - optional; type *long*

lists the jobs submitted after the specified date (UNIX timestamp)

- toSubmitDate - optional; type *long*
  lists jobs submitted before the specified date (UNIX timestamp)
- owner - optional; type *string*
  lists the jobs submitted by the specified owner
- status - optional; type *JobStatus* - type *string* with restriction - enum { 'UNDEFINED', 'SUBMITTED', 'QUEUED', 'WAITING', 'RUNNING', 'TERMINATED', 'CANCELLED', 'ALREADY_DOWNLOADED' }
  lists the jobs with the specified status
- priority - optional; type *JobPriority* - type *string* with restriction - enum { 'UNDEFINED', 'VERY_LOW', 'LOW', 'NORMAL', 'HIGH', 'VERY_HIGH' }
  lists the jobs with the specified priority
- queue - optional; type *string*
  the jobs with the specified queue name
- multipleStatus - optional; type *string*
  lists the jobs with the specified status (combination of multiple status). Its format contains the first letter or the value (interger) of each chosen status. For exemple to list the cancelled and terminated jobs, you use the format CT or 65
- batchJob - optional; type *boolean*
  allows to select all jobs submitted through the underlying batch scheduler (jobs submitted through vishnu and out of vishnu)

Output:

listJobsOutput (soapbind:body, use = literal)Source code

parameters type *listJobsResponse*

- data - optional;
  - job - optional, unbounded, nillable;
    - sessionId - optional; type *string*
      Is the id of the session that contained the job submission command
    - submitMachineId - optional; type *string*
      Is the id of the machine on which the job has been submitted.
    - submitMachineName - optional; type *string*
      Is the name of the machine on which the job has been submitted.
    - jobId - optional; type *string*
      Represents the id to job.
    - jobName - optional; type *string*
      Represents the name assigned to the job.
    - jobPath - optional; type *string*
      Is the path to the file containing job characteristics.
    - outputPath - optional; type *string*
      Is the path to the job output results.
    - errorPath - optional; type *string*
      Is the path to the file containing errors occured during job's execution.
    - jobPrio - optional; type *JobPriority* - type *string* with restriction - enum { 'UNDEFINED', 'VERY_LOW', 'LOW', 'NORMAL', 'HIGH', 'VERY_HIGH' }
      Represents the job priority.
    - nbCpus - optional; type *integer*
      Is the number of cpu per node used by the job.
    - jobWorkingDir - optional; type *string*

Indicates the directory where the job has been launched.

- status - optional; type *JobStatus* - type *string* with restriction - enum { 'UNDEFINED', 'SUBMITTED', 'QUEUED', 'WAITING', 'RUNNING', 'TERMINATED', 'CANCELLED', 'ALREADY_DOWNLOADED' }
  The current status of the job.
- submitDate - optional; type *long*
  Date and time when job was submitted (unix timestamp)
- endDate - optional; type *long*
  Represents the execution end date of the job (unix timestamp)
- owner - optional; type *string*
  Represents the job owner.
- jobQueue - optional; type *string*
  Is the name of the queue or class associated to the job.
- wallClockLimit - optional; type *long*
  Is the maximum wall-clock time during which the job can run (in seconds)
- groupName - optional; type *string*
  Represents the job owner group name.
- jobDescription - optional; type *string*
  Is the textual description of the job.
- memLimit - optional; type *integer*
  Represents the memory size limit of the job ((in MegaBytes).
- nbNodes - optional; type *integer*
  Is the total number of nodes used by the job.
- nbNodesAndCpuPerNode - optional; type *string*
  Is the number of nodes and processors per node used by the job (in the format nbNodes:nbCpuPerNode).

♦ nbJobs type *long*
Represents the total number of jobs in the list.
♦ nbRunningJobs type *long*
Represents of running jobs in the list.
♦ nbWaitingJobs type *long*
Represents the total number of waiting jobs in the list.

Fault:
BATCH_SCHEDULER_ERRORMessage (documentation, use = literal)Source code
fault type *BATCH_SCHEDULER_ERRORFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
PERMISSION_DENIEDMessage (documentation, use = literal)Source code
fault type *PERMISSION_DENIEDFault*
Fault:
UNKNOWN_BATCH_SCHEDULERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_BATCH_SCHEDULERFault*
Fault:

UNKNOWN_MACHINEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_MACHINEFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*

7. # listQueues

Source code
Description:
gets queues information
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
listQueuesInput (soapbind:body, use = literal)Source code
parameters type *listQueuesRequest*

- sessionKey type *string*
  The session key
- machineId type *string*
  Is the id of the machine that the user wants to list queues
- queueName - optional; type *string*
  if it is given, listQueues gives information only of this queue

Output:
listQueuesOutput (soapbind:body, use = literal)Source code
parameters type *listQueuesResponse*

- data - optional;
  - queue - optional, unbounded, nillable;
    - name - optional; type *string*
      Is the queue name.
    - maxJobCpu - optional; type *integer*
      Is the maximum number of Cups that a job can use.
    - maxProcCpu - optional; type *integer*
      Is the maximum number of Cpus of the queue.
    - memory - optional; type *integer*
      Represents the queue memory size.
    - wallTime - optional; type *long*
      Is the total wallTime of the queue.
    - node - optional; type *integer*
      Is the maximum number of nodes of the queue.
    - nbRunningJobs - optional; type *integer*

Is the total running jobs in the queue.
- nbJobsInQueue - optional; type *integer*
  Is the total number of jobs in the queue.
- state - optional; type *QueueStatus* - type *string* with restriction - enum { 'NOT_STARTED', 'STARTED', 'RUNNING' }
  Is the status of the queue.
- priority - optional; type *QueuePriority* - type *string* with restriction - enum { 'UNDEFINED', 'VERY_LOW', 'LOW', 'NORMAL', 'HIGH', 'VERY_HIGH' }
  Represents the priority of the queue.
- description - optional; type *string*
  Is the queue description.

♦ nbQueues type *integer*
Represents the number of queues.

Fault:
BATCH_SCHEDULER_ERRORMessage (documentation, use = literal)Source code
fault type *BATCH_SCHEDULER_ERRORFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
PERMISSION_DENIEDMessage (documentation, use = literal)Source code
fault type *PERMISSION_DENIEDFault*
Fault:
UNKNOWN_BATCH_SCHEDULERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_BATCH_SCHEDULERFault*
Fault:
UNKNOWN_MACHINEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_MACHINEFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*

8. # submitJob

Source code
Description:
submits a job on a machine through the use of a script (scriptFilePath). If the machine identifier is equal to autom, the job will be automatically submitted on a best machine (for now three criterions are used: minimum number of waiting jobs, minimum number of running jobs and the total number of jobs) through the use of a script (scriptFilePath) which must be

generic script using VISHNU's generic directives for all batch schedulers
Operation type:
Request-response. The endpoint receives a message, and sends a correlated message.
Input:
submitJobInput (soapbind:body, use = literal)Source code
parameters type *submitJobRequest*

- sessionKey type *string*
  The session key
- machineId type *string*
  Is the id of the machine on which the job must be submitted
- scriptFilePath type *string*
  The path to the file containing the characteristics (job command, and batch scheduler directive required or optional) of the job to submit.
- data - optional;
  - loadcriterion - optional, unbounded, nillable;
    - loadType - optional; type *LoadType* - type *string* with restriction - enum { 'USE_NB_WAITING_JOBS', 'USE_NB_JOBS', 'USE_NB_RUNNING_JOBS' }
      The criterion to automatically submit a job (for now three criterions are used: minimum number of waiting jobs, minimum number of running jobs and the total number of jobs). This option is used only if the machine identifier is equal to autom (this keyword is used to submit automatically a job)
- name - optional; type *string*
  Assigns a job name. The default is the path of job
- queue - optional; type *string*
  Assigns the queue or class of the job
- wallTime - optional; type *integer*
  The maximum wall-clock time during which the job can run (in seconds)
- memory - optional; type *integer*
  Is the memory size that the job requires (in MegaBytes)
- nbCpu - optional; type *integer*
  The number of cpu per node that the job requires
- nbNodesAndCpuPerNode - optional; type *string*
  The number of nodes and processors per node (in the format nbNodes:nbCpuPerNode). For example if you want to use 4 nodes with 3 cpus for each node, you must specify these numbers by "4:3"
- outputPath - optional; type *string*
  Assigns the path and file for job output
- errorPath - optional; type *string*
  Assigns the path and file for job error
- mailNotification - optional; type *string*
  Assigns the notification type of the job. Valid type values are BEGIN, END, ERROR, and ALL (any state change)
- mailNotifyUser - optional; type *string*
  The name of user to receive email notification of state changes as defined by the option mailNotification. The default value is the submitting user
- group - optional; type *string*
  Assigns a job group name.
- workingDir - optional; type *string*

Assigns a job remote working dir
- cpuTime - optional; type *string*
  Assigns a job cpu limit time (in seconds or in the format [[HH:]MM:]SS)
- selectQueueAutom - optional; type *boolean*
  allows to select automatically a queue which has the number of nodes requested by the user

Output:
submitJobOutput (soapbind:body, use = literal)Source code
parameters type *submitJobResponse*
- sessionId type *string*
  Is the id of the session that contained the job submission command
- submitMachineId type *string*
  Is the id of the machine on which the job has been submitted.
- submitMachineName type *string*
  Is the name of the machine on which the job has been submitted.
- jobId type *string*
  Represents the id to job.
- jobName type *string*
  Represents the name assigned to the job.
- jobPath type *string*
  Is the path to the file containing job characteristics.
- outputPath type *string*
  Is the path to the job output results.
- errorPath type *string*
  Is the path to the file containing errors occured during job's execution.
- jobPrio type *JobPriority* - type *string* with restriction - enum { 'UNDEFINED', 'VERY_LOW', 'LOW', 'NORMAL', 'HIGH', 'VERY_HIGH' }
  Represents the job priority.
- nbCpus type *integer*
  Is the number of cpu per node used by the job.
- jobWorkingDir type *string*
  Indicates the directory where the job has been launched.
- status type *JobStatus* - type *string* with restriction - enum { 'UNDEFINED', 'SUBMITTED', 'QUEUED', 'WAITING', 'RUNNING', 'TERMINATED', 'CANCELLED', 'ALREADY_DOWNLOADED' }
  The current status of the job.
- submitDate type *long*
  Date and time when job was submitted (unix timestamp)
- endDate type *long*
  Represents the execution end date of the job (unix timestamp)
- owner type *string*
  Represents the job owner.
- jobQueue type *string*
  Is the name of the queue or class associated to the job.
- wallClockLimit type *long*
  Is the maximum wall-clock time during which the job can run (in seconds)
- groupName type *string*
  Represents the job owner group name.
- jobDescription type *string*
  Is the textual description of the job.
- memLimit type *integer*
  Represents the memory size limit of the job ((in MegaBytes).

- nbNodes type *integer*
  Is the total number of nodes used by the job.
- nbNodesAndCpuPerNode type *string*
  Is the number of nodes and processors per node used by the job (in the format nbNodes:nbCpuPerNode).

Fault:
UNKNOWN_MACHINEMessage (documentation, use = literal)Source code
fault type *UNKNOWN_MACHINEFault*
Fault:
BATCH_SCHEDULER_ERRORMessage (documentation, use = literal)Source code
fault type *BATCH_SCHEDULER_ERRORFault*
Fault:
INVALID_PARAMMessage (documentation, use = literal)Source code
fault type *INVALID_PARAMFault*
Fault:
SESSIONKEY_EXPIREDMessage (documentation, use = literal)Source code
fault type *SESSIONKEY_EXPIREDFault*
Fault:
PERMISSION_DENIEDMessage (documentation, use = literal)Source code
fault type *PERMISSION_DENIEDFault*
Fault:
UNKNOWN_BATCH_SCHEDULERMessage (documentation, use = literal)Source code
fault type *UNKNOWN_BATCH_SCHEDULERFault*
Fault:
UNDEFINEDMessage (documentation, use = literal)Source code
fault type *UNDEFINEDFault*
Fault:
DIETMessage (documentation, use = literal)Source code
fault type *DIETFault*
Fault:
DBERRMessage (documentation, use = literal)Source code
fault type *DBERRFault*
Fault:
DBCONNMessage (documentation, use = literal)Source code
fault type *DBCONNFault*
Fault:
SYSTEMMessage (documentation, use = literal)Source code
fault type *SYSTEMFault*
Fault:
SSHMessage (documentation, use = literal)Source code
fault type *SSHFault*

WSDL source code

```
<?xml version="1.0"?>
<definitions name="VishnuTMS" targetNamespace="urn:ResourceProxy"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="urn:ResourceProxy"
xmlns:soapbind="http://schemas.xmlsoap.org/wsdl/soap/"
```

```
>
<types>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="unqualified"
targetNamespace="urn:ResourceProxy"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
>
<xs:element name="submitJobRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the machine on which the job must be submitted</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="scriptFilePath" type="xs:string">
<xs:annotation>
<xs:documentation>The path to the file containing the characteristics (job command, and batch scheduler
directive required or optional) of the job to submit.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="loadcriterion" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="loadType" type="Q1:LoadType" use="optional">
<xs:annotation>
<xs:documentation>The criterion to automatically submit a job (for now three criterions are used: minimum
number of waiting jobs, minimum number of running jobs and the total number of jobs). This option is used
only if the machine identifier is equal to autom (this keyword is used to submit automatically a
job)</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element minOccurs="0" name="name" type="xs:string">
<xs:annotation>
<xs:documentation>Assigns a job name. The default is the path of job</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="queue" type="xs:string">
<xs:annotation>
```

<xs:documentation>Assigns the queue or class of the job</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="wallTime" type="xs:integer">
<xs:annotation>
<xs:documentation>The maximum wall-clock time during which the job can run (in seconds)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="memory" type="xs:integer">
<xs:annotation>
<xs:documentation>Is the memory size that the job requires (in MegaBytes)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="nbCpu" type="xs:integer">
<xs:annotation>
<xs:documentation>The number of cpu per node that the job requires</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="nbNodesAndCpuPerNode" type="xs:string">
<xs:annotation>
<xs:documentation>The number of nodes and processors per node (in the format nbNodes:nbCpuPerNode). For example if you want to use 4 nodes with 3 cpus for each node, you must specify these numbers by "4:3"</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="outputPath" type="xs:string">
<xs:annotation>
<xs:documentation>Assigns the path and file for job output</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="errorPath" type="xs:string">
<xs:annotation>
<xs:documentation>Assigns the path and file for job error</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="mailNotification" type="xs:string">
<xs:annotation>
<xs:documentation>Assigns the notification type of the job. Valid type values are BEGIN, END, ERROR, and ALL (any state change)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="mailNotifyUser" type="xs:string">
<xs:annotation>
<xs:documentation>The name of user to receive email notification of state changes as defined by the option mailNotification. The default value is the submitting user</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="group" type="xs:string">
<xs:annotation>
<xs:documentation>Assigns a job group name.</xs:documentation>

```
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="workingDir" type="xs:string">
<xs:annotation>
<xs:documentation>Assigns a job remote working dir</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="cpuTime" type="xs:string">
<xs:annotation>
<xs:documentation>Assigns a job cpu limit time (in seconds or in the format
[[HH:]MM:]SS)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="selectQueueAutom" type="xs:boolean">
<xs:annotation>
<xs:documentation>allows to select automatically a queue which has the number of nodes requested by the
user</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="LoadType">
<xs:restriction base="xs:string">
<xs:enumeration value="USE_NB_WAITING_JOBS" />
<xs:enumeration value="USE_NB_JOBS" />
<xs:enumeration value="USE_NB_RUNNING_JOBS" />
</xs:restriction>
</xs:simpleType>
<xs:element name="submitJobResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the session that contained the job submission command</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="submitMachineId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the machine on which the job has been submitted.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="submitMachineName" type="xs:string">
<xs:annotation>
<xs:documentation>Is the name of the machine on which the job has been submitted.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobId" type="xs:string">
<xs:annotation>
<xs:documentation>Represents the id to job.</xs:documentation>
</xs:annotation>
```

```
</xs:element>
<xs:element minOccurs="1" name="jobName" type="xs:string">
<xs:annotation>
<xs:documentation>Represents the name assigned to the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobPath" type="xs:string">
<xs:annotation>
<xs:documentation>Is the path to the file containing job characteristics.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="outputPath" type="xs:string">
<xs:annotation>
<xs:documentation>Is the path to the job output results.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="errorPath" type="xs:string">
<xs:annotation>
<xs:documentation>Is the path to the file containing errors occured during job's execution.
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobPrio" type="Q1:JobPriority">
<xs:annotation>
<xs:documentation>Represents the job priority. </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="nbCpus" type="xs:integer">
<xs:annotation>
<xs:documentation>Is the number of cpu per node used by the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobWorkingDir" type="xs:string">
<xs:annotation>
<xs:documentation>Indicates the directory where the job has been launched.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="status" type="Q1:JobStatus">
<xs:annotation>
<xs:documentation>The current status of the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="submitDate" type="xs:long">
<xs:annotation>
<xs:documentation>Date and time when job was submitted (unix timestamp)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="endDate" type="xs:long">
<xs:annotation>
<xs:documentation>Represents the execution end date of the job (unix timestamp)</xs:documentation>
</xs:annotation>
```

```
</xs:element>
<xs:element minOccurs="1" name="owner" type="xs:string">
<xs:annotation>
<xs:documentation>Represents the job owner.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobQueue" type="xs:string">
<xs:annotation>
<xs:documentation>Is the name of the queue or class associated to the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="wallClockLimit" type="xs:long">
<xs:annotation>
<xs:documentation>Is the maximum wall-clock time during which the job can run (in
seconds)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="groupName" type="xs:string">
<xs:annotation>
<xs:documentation>Represents the job owner group name.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobDescription" type="xs:string">
<xs:annotation>
<xs:documentation>Is the textual description of the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="memLimit" type="xs:integer">
<xs:annotation>
<xs:documentation>Represents the memory size limit of the job ((in MegaBytes).</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="nbNodes" type="xs:integer">
<xs:annotation>
<xs:documentation>Is the total number of nodes used by the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="nbNodesAndCpuPerNode" type="xs:string">
<xs:annotation>
<xs:documentation>Is the number of nodes and processors per node used by the job (in the format
nbNodes:nbCpuPerNode).</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="JobPriority">
<xs:restriction base="xs:string">
<xs:enumeration value="UNDEFINED" />
<xs:enumeration value="VERY_LOW" />
<xs:enumeration value="LOW" />
```

```
<xs:enumeration value="NORMAL" />
<xs:enumeration value="HIGH" />
<xs:enumeration value="VERY_HIGH" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="JobStatus">
<xs:restriction base="xs:string">
<xs:enumeration value="UNDEFINED" />
<xs:enumeration value="SUBMITTED" />
<xs:enumeration value="QUEUED" />
<xs:enumeration value="WAITING" />
<xs:enumeration value="RUNNING" />
<xs:enumeration value="TERMINATED" />
<xs:enumeration value="CANCELLED" />
<xs:enumeration value="ALREADY_DOWNLOADED" />
</xs:restriction>
</xs:simpleType>
<xs:element name="getJobInfoRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the machine on which the job is running</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobId" type="xs:string">
<xs:annotation>
<xs:documentation>The id of the job </xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getJobInfoResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the session that contained the job submission command</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="submitMachineId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the machine on which the job has been submitted.</xs:documentation>
</xs:annotation>
</xs:element>
```

```
<xs:element minOccurs="1" name="submitMachineName" type="xs:string">
<xs:annotation>
<xs:documentation>Is the name of the machine on which the job has been submitted.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobId" type="xs:string">
<xs:annotation>
<xs:documentation>Represents the id to job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobName" type="xs:string">
<xs:annotation>
<xs:documentation>Represents the name assigned to the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobPath" type="xs:string">
<xs:annotation>
<xs:documentation>Is the path to the file containing job characteristics.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="outputPath" type="xs:string">
<xs:annotation>
<xs:documentation>Is the path to the job output results.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="errorPath" type="xs:string">
<xs:annotation>
<xs:documentation>Is the path to the file containing errors occured during job's execution.
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobPrio" type="Q1:JobPriority">
<xs:annotation>
<xs:documentation>Represents the job priority. </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="nbCpus" type="xs:integer">
<xs:annotation>
<xs:documentation>Is the number of cpu per node used by the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobWorkingDir" type="xs:string">
<xs:annotation>
<xs:documentation>Indicates the directory where the job has been launched.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="status" type="Q1:JobStatus">
<xs:annotation>
<xs:documentation>The current status of the job.</xs:documentation>
</xs:annotation>
</xs:element>
```

```
<xs:element minOccurs="1" name="submitDate" type="xs:long">
<xs:annotation>
<xs:documentation>Date and time when job was submitted (unix timestamp)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="endDate" type="xs:long">
<xs:annotation>
<xs:documentation>Represents the execution end date of the job (unix timestamp)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="owner" type="xs:string">
<xs:annotation>
<xs:documentation>Represents the job owner.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobQueue" type="xs:string">
<xs:annotation>
<xs:documentation>Is the name of the queue or class associated to the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="wallClockLimit" type="xs:long">
<xs:annotation>
<xs:documentation>Is the maximum wall-clock time during which the job can run (in
seconds)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="groupName" type="xs:string">
<xs:annotation>
<xs:documentation>Represents the job owner group name.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobDescription" type="xs:string">
<xs:annotation>
<xs:documentation>Is the textual description of the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="memLimit" type="xs:integer">
<xs:annotation>
<xs:documentation>Represents the memory size limit of the job ((in MegaBytes).</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="nbNodes" type="xs:integer">
<xs:annotation>
<xs:documentation>Is the total number of nodes used by the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="nbNodesAndCpuPerNode" type="xs:string">
<xs:annotation>
<xs:documentation>Is the number of nodes and processors per node used by the job (in the format
nbNodes:nbCpuPerNode).</xs:documentation>
</xs:annotation>
```

```
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getJobProgressRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the machine to get the jobs progression</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="jobId" type="xs:string">
<xs:annotation>
<xs:documentation>Specifies the id of the job whose progression the user wants to see.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="jobOwner" type="xs:string">
<xs:annotation>
<xs:documentation>Specifies the owner of the job.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getJobProgressResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="progression" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="jobId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Represents the job id.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="jobName" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Represents the job name.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="wallTime" type="xs:integer" use="optional">
<xs:annotation>
```

```
<xs:documentation>Represents the job wall time.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="startTime" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>Start date and time of the job (unix timestamp)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="endTime" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>End date and time of the job (unix timestamp)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="percent" type="xs:integer" use="optional">
<xs:annotation>
<xs:documentation>Represent the job progression. </xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="status" type="Q1:JobStatus" use="optional">
<xs:annotation>
<xs:documentation>Represents the job status.</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element minOccurs="1" name="nbJobs" type="xs:integer">
<xs:annotation>
<xs:documentation>Represents the number of jobs in progression list.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listQueuesRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the machine that the user wants to list queues</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="queueName" type="xs:string">
<xs:annotation>
```

```
<xs:documentation>if it is given, listQueues gives information only of this queue</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listQueuesResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="queue" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="name" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Is the queue name.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="maxJobCpu" type="xs:integer" use="optional">
<xs:annotation>
<xs:documentation>Is the maximum number of Cups that a job can use.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="maxProcCpu" type="xs:integer" use="optional">
<xs:annotation>
<xs:documentation>Is the maximum number of Cpus of the queue.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="memory" type="xs:integer" use="optional">
<xs:annotation>
<xs:documentation>Represents the queue memory size.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="wallTime" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>Is the total wallTime of the queue.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="node" type="xs:integer" use="optional">
<xs:annotation>
<xs:documentation>Is the maximum number of nodes of the queue.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="nbRunningJobs" type="xs:integer" use="optional">
<xs:annotation>
<xs:documentation>Is the total running jobs in the queue.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="nbJobsInQueue" type="xs:integer" use="optional">
```

```
<xs:annotation>
<xs:documentation>Is the total number of jobs in the queue.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="state" type="Q1:QueueStatus" use="optional">
<xs:annotation>
<xs:documentation>Is the status of the queue.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="priority" type="Q1:QueuePriority" use="optional">
<xs:annotation>
<xs:documentation>Represents the priority of the queue.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Is the queue description.</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element minOccurs="1" name="nbQueues" type="xs:integer">
<xs:annotation>
<xs:documentation>Represents the number of queues.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:simpleType name="QueueStatus">
<xs:restriction base="xs:string">
<xs:enumeration value="NOT_STARTED" />
<xs:enumeration value="STARTED" />
<xs:enumeration value="RUNNING" />
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="QueuePriority">
<xs:restriction base="xs:string">
<xs:enumeration value="UNDEFINED" />
<xs:enumeration value="VERY_LOW" />
<xs:enumeration value="LOW" />
<xs:enumeration value="NORMAL" />
<xs:enumeration value="HIGH" />
<xs:enumeration value="VERY_HIGH" />
</xs:restriction>
</xs:simpleType>
<xs:element name="listJobsRequest">
<xs:complexType>
```

```
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the machine on which the jobs are running</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="jobId" type="xs:string">
<xs:annotation>
<xs:documentation>lists the job with the specified id</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="nbCpu" type="xs:integer">
<xs:annotation>
<xs:documentation>lists the jobs with the specified number of CPUs per node</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="fromSubmitDate" type="xs:long">
<xs:annotation>
<xs:documentation>lists the jobs submitted after the specified date (UNIX timestamp)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="toSubmitDate" type="xs:long">
<xs:annotation>
<xs:documentation>lists jobs submitted before the specified date (UNIX timestamp)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="owner" type="xs:string">
<xs:annotation>
<xs:documentation>lists the jobs submitted by the specified owner</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="status" type="Q1:JobStatus">
<xs:annotation>
<xs:documentation>lists the jobs with the specified status</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="priority" type="Q1:JobPriority">
<xs:annotation>
<xs:documentation>lists the jobs with the specified priority</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="queue" type="xs:string">
<xs:annotation>
<xs:documentation>the jobs with the specified queue name</xs:documentation>
</xs:annotation>
</xs:element>
```

```
<xs:element minOccurs="0" name="multipleStatus" type="xs:string">
<xs:annotation>
<xs:documentation>lists the jobs with the specified status (combination of multiple status). Its format contains the first letter or the value (interger) of each chosen status. For exemple to list the cancelled and terminated jobs, you use the format CT or 65</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="batchJob" type="xs:boolean">
<xs:annotation>
<xs:documentation>allows to select all jobs submitted through the underlying batch scheduler (jobs submitted through vishnu and out of vishnu)</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="listJobsResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="job" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="sessionId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Is the id of the session that contained the job submission command</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="submitMachineId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Is the id of the machine on which the job has been submitted.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="submitMachineName" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Is the name of the machine on which the job has been submitted.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="jobId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Represents the id to job.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="jobName" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Represents the name assigned to the job.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="jobPath" type="xs:string" use="optional">
```

```
<xs:annotation>
<xs:documentation>Is the path to the file containing job characteristics.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="outputPath" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Is the path to the job output results.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="errorPath" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Is the path to the file containing errors occured during job's execution.
</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="jobPrio" type="Q1:JobPriority" use="optional">
<xs:annotation>
<xs:documentation>Represents the job priority. </xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="nbCpus" type="xs:integer" use="optional">
<xs:annotation>
<xs:documentation>Is the number of cpu per node used by the job.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="jobWorkingDir" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Indicates the directory where the job has been launched.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="status" type="Q1:JobStatus" use="optional">
<xs:annotation>
<xs:documentation>The current status of the job.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="submitDate" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>Date and time when job was submitted (unix timestamp)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="endDate" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>Represents the execution end date of the job (unix timestamp)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="owner" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Represents the job owner.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="jobQueue" type="xs:string" use="optional">
```

```
<xs:annotation>
<xs:documentation>Is the name of the queue or class associated to the job.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="wallClockLimit" type="xs:long" use="optional">
<xs:annotation>
<xs:documentation>Is the maximum wall-clock time during which the job can run (in
seconds)</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="groupName" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Represents the job owner group name.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="jobDescription" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Is the textual description of the job.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="memLimit" type="xs:integer" use="optional">
<xs:annotation>
<xs:documentation>Represents the memory size limit of the job ((in MegaBytes).</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="nbNodes" type="xs:integer" use="optional">
<xs:annotation>
<xs:documentation>Is the total number of nodes used by the job.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="nbNodesAndCpuPerNode" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Is the number of nodes and processors per node used by the job (in the format
nbNodes:nbCpuPerNode).</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element minOccurs="1" name="nbJobs" type="xs:long">
<xs:annotation>
<xs:documentation>Represents the total number of jobs in the list.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="nbRunningJobs" type="xs:long">
<xs:annotation>
<xs:documentation>Represents of running jobs in the list.</xs:documentation>
</xs:annotation>
</xs:element>
```

```
<xs:element minOccurs="1" name="nbWaitingJobs" type="xs:long">
<xs:annotation>
<xs:documentation>Represents the total number of waiting jobs in the list.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getJobOutputRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>gets outputPath and errorPath of a job from its id</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobId" type="xs:string">
<xs:annotation>
<xs:documentation>The Id of the job</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="outDir" type="xs:string">
<xs:annotation>
<xs:documentation>The output directory where the files will be stored (default is current
directory)</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getJobOutputResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="jobId" type="xs:string">
<xs:annotation>
<xs:documentation>Represents the id of the job.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="outputPath" type="xs:string">
<xs:annotation>
<xs:documentation>Is the path to the job output results.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="errorPath" type="xs:string">
<xs:annotation>
```

```
<xs:documentation>Is the path to the file containing errors occured during job's execution.
</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getCompletedJobsOutputRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the machine on which the jobs are been submitted</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="0" name="outDir" type="xs:string">
<xs:annotation>
<xs:documentation>Specifies the output directory where the files will be stored (by default, the current
directory).</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="getCompletedJobsOutputResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="data">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="jobresult" nillable="true">
<xs:complexType>
<xs:sequence />
<xs:attribute name="jobId" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Represents the id of the job.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="outputPath" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>Is the path to the job output results.</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="errorPath" type="xs:string" use="optional">
<xs:annotation>
```

```
<xs:documentation>Is the path to the file containing errors occured during job's execution.
</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element minOccurs="1" name="nbJobs" type="xs:integer">
<xs:annotation>
<xs:documentation>Is the number of jobs.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="cancelJobRequest">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="1" name="sessionKey" type="xs:string">
<xs:annotation>
<xs:documentation>The session key</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="machineId" type="xs:string">
<xs:annotation>
<xs:documentation>Is the id of the machine on which the job is running</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element minOccurs="1" name="jobId" type="xs:string">
<xs:annotation>
<xs:documentation>The Id of the job</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="cancelJobResponse">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="DIETFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="DBERRFault">
<xs:complexType>
<xs:sequence />
```

```
</xs:complexType>
</xs:element>
<xs:element name="DBCONNFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="SYSTEMFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="SSHFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNDEFINEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="INVALID_PARAMFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="FILENOTFOUNDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="CONFIGNOTFOUNDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_MACHINEFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="SESSIONKEY_NOT_FOUNDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="SESSIONKEY_EXPIREDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
```

```
</xs:element>
<xs:element name="UNKNOWN_SESSION_IDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_BATCH_SCHEDULERFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="BATCH_SCHEDULER_ERRORFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="UNKNOWN_JOBIDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="PERMISSION_DENIEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="ALREADY_TERMINATEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="ALREADY_CANCELEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="JOB_IS_NOT_TERMINATEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
<xs:element name="ALREADY_DOWNLOADEDFault">
<xs:complexType>
<xs:sequence />
</xs:complexType>
</xs:element>
</xs:schema>
</types>
<message name="submitJobInput">
<part element="tns:submitJobRequest" name="parameters" />
</message>
```

<message name="submitJobOutput">
<part element="tns:submitJobResponse" name="parameters" />
</message>
<message name="getJobInfoInput">
<part element="tns:getJobInfoRequest" name="parameters" />
</message>
<message name="getJobInfoOutput">
<part element="tns:getJobInfoResponse" name="parameters" />
</message>
<message name="getJobProgressInput">
<part element="tns:getJobProgressRequest" name="parameters" />
</message>
<message name="getJobProgressOutput">
<part element="tns:getJobProgressResponse" name="parameters" />
</message>
<message name="listQueuesInput">
<part element="tns:listQueuesRequest" name="parameters" />
</message>
<message name="listQueuesOutput">
<part element="tns:listQueuesResponse" name="parameters" />
</message>
<message name="listJobsInput">
<part element="tns:listJobsRequest" name="parameters" />
</message>
<message name="listJobsOutput">
<part element="tns:listJobsResponse" name="parameters" />
</message>
<message name="getJobOutputInput">
<part element="tns:getJobOutputRequest" name="parameters" />
</message>
<message name="getJobOutputOutput">
<part element="tns:getJobOutputResponse" name="parameters" />
</message>
<message name="getCompletedJobsOutputInput">
<part element="tns:getCompletedJobsOutputRequest" name="parameters" />
</message>
<message name="getCompletedJobsOutputOutput">
<part element="tns:getCompletedJobsOutputResponse" name="parameters" />
</message>
<message name="cancelJobInput">
<part element="tns:cancelJobRequest" name="parameters" />
</message>
<message name="cancelJobOutput">
<part element="tns:cancelJobResponse" name="parameters" />
</message>
<message name="UNKNOWN_MACHINEMessage">
<part element="tns:UNKNOWN_MACHINEFault" name="fault" />
</message>
<message name="BATCH_SCHEDULER_ERRORMessage">
<part element="tns:BATCH_SCHEDULER_ERRORFault" name="fault" />
</message>

<message name="INVALID_PARAMMessage">
<part element="tns:INVALID_PARAMFault" name="fault" />
</message>
<message name="SESSIONKEY_EXPIREDMessage">
<part element="tns:SESSIONKEY_EXPIREDFault" name="fault" />
</message>
<message name="PERMISSION_DENIEDMessage">
<part element="tns:PERMISSION_DENIEDFault" name="fault" />
</message>
<message name="UNKNOWN_BATCH_SCHEDULERMessage">
<part element="tns:UNKNOWN_BATCH_SCHEDULERFault" name="fault" />
</message>
<message name="UNDEFINEDMessage">
<part element="tns:UNDEFINEDFault" name="fault" />
</message>
<message name="DIETMessage">
<part element="tns:DIETFault" name="fault" />
</message>
<message name="DBERRMessage">
<part element="tns:DBERRFault" name="fault" />
</message>
<message name="DBCONNMessage">
<part element="tns:DBCONNFault" name="fault" />
</message>
<message name="SYSTEMMessage">
<part element="tns:SYSTEMFault" name="fault" />
</message>
<message name="SSHMessage">
<part element="tns:SSHFault" name="fault" />
</message>
<message name="JOB_IS_NOT_TERMINATEDMessage">
<part element="tns:JOB_IS_NOT_TERMINATEDFault" name="fault" />
</message>
<message name="ALREADY_DOWNLOADEDMessage">
<part element="tns:ALREADY_DOWNLOADEDFault" name="fault" />
</message>
<message name="ALREADY_CANCELEDMessage">
<part element="tns:ALREADY_CANCELEDFault" name="fault" />
</message>
<message name="ALREADY_TERMINATEDMessage">
<part element="tns:ALREADY_TERMINATEDFault" name="fault" />
</message>
<portType name="VishnuTMSPortType">
<operation name="submitJob">
<documentation>submits a job on a machine through the use of a script (scriptFilePath). If the machine identifier is equal to autom, the job will be automatically submitted on a best machine (for now three criterions are used: minimum number of waiting jobs, minimum number of running jobs and the total number of jobs) through the use of a script (scriptFilePath) which must be generic script using VISHNU's generic directives for all batch schedulers</documentation>
<input message="tns:submitJobInput" name="submitJobInput" />
<output message="tns:submitJobOutput" name="submitJobOutput" />

<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:BATCH_SCHEDULER_ERRORMessage" name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
</fault>
<fault message="tns:INVALID_PARAMMessage" name="INVALID_PARAM">
<documentation>Error invalid parameters</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:PERMISSION_DENIEDMessage" name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
</fault>
<fault message="tns:UNKNOWN_BATCH_SCHEDULERMessage"
name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:SSHMessage" name="SSH">
<documentation>Vishnu not available (SSH error)</documentation>
</fault>
</operation>
<operation name="getJobInfo">
<documentation>gets information on a job from its id</documentation>
<input message="tns:getJobInfoInput" name="getJobInfoInput" />
<output message="tns:getJobInfoOutput" name="getJobInfoOutput" />
<fault message="tns:BATCH_SCHEDULER_ERRORMessage" name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:PERMISSION_DENIEDMessage" name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
</fault>

<fault message="tns:UNKNOWN_BATCH_SCHEDULERMessage"
name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
</operation>
<operation name="getJobProgress">
<documentation>gets the progression status of jobs</documentation>
<input message="tns:getJobProgressInput" name="getJobProgressInput" />
<output message="tns:getJobProgressOutput" name="getJobProgressOutput" />
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
</operation>
<operation name="listQueues">
<documentation>gets queues information</documentation>

<input message="tns:listQueuesInput" name="listQueuesInput" />
<output message="tns:listQueuesOutput" name="listQueuesOutput" />
<fault message="tns:BATCH_SCHEDULER_ERRORMessage" name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:PERMISSION_DENIEDMessage" name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
</fault>
<fault message="tns:UNKNOWN_BATCH_SCHEDULERMessage"
name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
</operation>
<operation name="listJobs">
<documentation>gets a list of all submitted jobs on a machine. If machine identifier is equal to all, submitted jobs on all machines are listed</documentation>
<input message="tns:listJobsInput" name="listJobsInput" />
<output message="tns:listJobsOutput" name="listJobsOutput" />
<fault message="tns:BATCH_SCHEDULER_ERRORMessage" name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:PERMISSION_DENIEDMessage" name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
</fault>
<fault message="tns:UNKNOWN_BATCH_SCHEDULERMessage"
name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>

```
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
</operation>
<operation name="getJobOutput">
<documentation>gets standard output and error output files of a job given its id</documentation>
<input message="tns:getJobOutputInput" name="getJobOutputInput" />
<output message="tns:getJobOutputOutput" name="getJobOutputOutput" />
<fault message="tns:BATCH_SCHEDULER_ERRORMessage" name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:PERMISSION_DENIEDMessage" name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
</fault>
<fault message="tns:UNKNOWN_BATCH_SCHEDULERMessage"
name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
```

```
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:JOB_IS_NOT_TERMINATEDMessage" name="JOB_IS_NOT_TERMINATED">
<documentation>The job is not terminated</documentation>
</fault>
<fault message="tns:SSHMessage" name="SSH">
<documentation>Vishnu not available (SSH error)</documentation>
</fault>
<fault message="tns:ALREADY_DOWNLOADEDMessage" name="ALREADY_DOWNLOADED">
<documentation>The job is already downloaded</documentation>
</fault>
</operation>
<operation name="getCompletedJobsOutput">
<documentation>gets standard output and error output files of completed jobs (applies only once for each
job)</documentation>
<input message="tns:getCompletedJobsOutputInput" name="getCompletedJobsOutputInput" />
<output message="tns:getCompletedJobsOutputOutput" name="getCompletedJobsOutputOutput" />
<fault message="tns:BATCH_SCHEDULER_ERRORMessage" name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:PERMISSION_DENIEDMessage" name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
</fault>
<fault message="tns:UNKNOWN_BATCH_SCHEDULERMessage"
name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:SSHMessage" name="SSH">
```

<documentation>Vishnu not available (SSH error)</documentation>
</fault>
</operation>
<operation name="cancelJob">
<documentation>cancels a job from its id. If job id is equal to all, all submitted jobs by all users will be cancelled if the user is an administrator, and only jobs submitted by the user will be cancelled if the user is not an administrator.</documentation>
<input message="tns:cancelJobInput" name="cancelJobInput" />
<output message="tns:cancelJobOutput" name="cancelJobOutput" />
<fault message="tns:BATCH_SCHEDULER_ERRORMessage" name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
</fault>
<fault message="tns:PERMISSION_DENIEDMessage" name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
</fault>
<fault message="tns:SESSIONKEY_EXPIREDMessage" name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
</fault>
<fault message="tns:UNKNOWN_BATCH_SCHEDULERMessage"
name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
</fault>
<fault message="tns:UNKNOWN_MACHINEMessage" name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
</fault>
<fault message="tns:UNDEFINEDMessage" name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
</fault>
<fault message="tns:DIETMessage" name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
</fault>
<fault message="tns:DBERRMessage" name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
</fault>
<fault message="tns:DBCONNMessage" name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
</fault>
<fault message="tns:SYSTEMMessage" name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
</fault>
<fault message="tns:ALREADY_CANCELEDMessage" name="ALREADY_CANCELED">
<documentation>The job is already canceled</documentation>
</fault>
<fault message="tns:ALREADY_TERMINATEDMessage" name="ALREADY_TERMINATED">
<documentation>The job is already terminated</documentation>
</fault>
<fault message="tns:SSHMessage" name="SSH">
<documentation>Vishnu not available (SSH error)</documentation>
</fault>
</operation>
</portType>

```
<binding name="VishnuTMSSOAPBinding" type="tns:VishnuTMSPortType">
<soapbind:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="submitJob">
<documentation>submits a job on a machine through the use of a script (scriptFilePath). If the machine identifier is equal to autom, the job will be automatically submitted on a best machine (for now three criterions are used: minimum number of waiting jobs, minimum number of running jobs and the total number of jobs) through the use of a script (scriptFilePath) which must be generic script using VISHNU's generic directives for all batch schedulers</documentation>
<soapbind:operation soapAction="" />
<input name="submitJobInput">
<soapbind:body use="literal" />
</input>
<output name="submitJobOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
<soapbind:fault name="BATCH_SCHEDULER_ERROR" use="literal" />
</fault>
<fault name="INVALID_PARAM">
<documentation>Error invalid parameters</documentation>
<soapbind:fault name="INVALID_PARAM" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
<soapbind:fault name="PERMISSION_DENIED" use="literal" />
</fault>
<fault name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
<soapbind:fault name="UNKNOWN_BATCH_SCHEDULER" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
```

<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="SSH">
<documentation>Vishnu not available (SSH error)</documentation>
<soapbind:fault name="SSH" use="literal" />
</fault>
</operation>
<operation name="getJobInfo">
<documentation>gets information on a job from its id</documentation>
<soapbind:operation soapAction="" />
<input name="getJobInfoInput">
<soapbind:body use="literal" />
</input>
<output name="getJobInfoOutput">
<soapbind:body use="literal" />
</output>
<fault name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
<soapbind:fault name="BATCH_SCHEDULER_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
<soapbind:fault name="PERMISSION_DENIED" use="literal" />
</fault>
<fault name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
<soapbind:fault name="UNKNOWN_BATCH_SCHEDULER" use="literal" />
</fault>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">

<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
</operation>
<operation name="getJobProgress">
<documentation>gets the progression status of jobs</documentation>
<soapbind:operation soapAction="" />
<input name="getJobProgressInput">
<soapbind:body use="literal" />
</input>
<output name="getJobProgressOutput">
<soapbind:body use="literal" />
</output>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
</operation>
<operation name="listQueues">

<documentation>gets queues information</documentation>
<soapbind:operation soapAction="" />
<input name="listQueuesInput">
<soapbind:body use="literal" />
</input>
<output name="listQueuesOutput">
<soapbind:body use="literal" />
</output>
<fault name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
<soapbind:fault name="BATCH_SCHEDULER_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
<soapbind:fault name="PERMISSION_DENIED" use="literal" />
</fault>
<fault name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
<soapbind:fault name="UNKNOWN_BATCH_SCHEDULER" use="literal" />
</fault>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
</operation>
<operation name="listJobs">

<documentation>gets a list of all submitted jobs on a machine. If machine identifier is equal to all, submitted jobs on all machines are listed</documentation>
<soapbind:operation soapAction="" />
<input name="listJobsInput">
<soapbind:body use="literal" />
</input>
<output name="listJobsOutput">
<soapbind:body use="literal" />
</output>
<fault name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
<soapbind:fault name="BATCH_SCHEDULER_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
<soapbind:fault name="PERMISSION_DENIED" use="literal" />
</fault>
<fault name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
<soapbind:fault name="UNKNOWN_BATCH_SCHEDULER" use="literal" />
</fault>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
</operation>
<operation name="getJobOutput">

<documentation>gets standard output and error output files of a job given its id</documentation>
<soapbind:operation soapAction="" />
<input name="getJobOutputInput">
<soapbind:body use="literal" />
</input>
<output name="getJobOutputOutput">
<soapbind:body use="literal" />
</output>
<fault name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
<soapbind:fault name="BATCH_SCHEDULER_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
<soapbind:fault name="PERMISSION_DENIED" use="literal" />
</fault>
<fault name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
<soapbind:fault name="UNKNOWN_BATCH_SCHEDULER" use="literal" />
</fault>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="JOB_IS_NOT_TERMINATED">
<documentation>The job is not terminated</documentation>
<soapbind:fault name="JOB_IS_NOT_TERMINATED" use="literal" />

```
</fault>
<fault name="SSH">
<documentation>Vishnu not available (SSH error)</documentation>
<soapbind:fault name="SSH" use="literal" />
</fault>
<fault name="ALREADY_DOWNLOADED">
<documentation>The job is already downloaded</documentation>
<soapbind:fault name="ALREADY_DOWNLOADED" use="literal" />
</fault>
</operation>
<operation name="getCompletedJobsOutput">
<documentation>gets standard output and error output files of completed jobs (applies only once for each job)</documentation>
<soapbind:operation soapAction="" />
<input name="getCompletedJobsOutputInput">
<soapbind:body use="literal" />
</input>
<output name="getCompletedJobsOutputOutput">
<soapbind:body use="literal" />
</output>
<fault name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
<soapbind:fault name="BATCH_SCHEDULER_ERROR" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
<soapbind:fault name="PERMISSION_DENIED" use="literal" />
</fault>
<fault name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
<soapbind:fault name="UNKNOWN_BATCH_SCHEDULER" use="literal" />
</fault>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
```

```
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="SSH">
<documentation>Vishnu not available (SSH error)</documentation>
<soapbind:fault name="SSH" use="literal" />
</fault>
</operation>
<operation name="cancelJob">
<documentation>cancels a job from its id. If job id is equal to all, all submitted jobs by all users will be cancelled if the user is an administrator, and only jobs submitted by the user will be cancelled if the user is not an administrator.</documentation>
<soapbind:operation soapAction="" />
<input name="cancelJobInput">
<soapbind:body use="literal" />
</input>
<output name="cancelJobOutput">
<soapbind:body use="literal" />
</output>
<fault name="BATCH_SCHEDULER_ERROR">
<documentation>The batch scheduler indicates an error</documentation>
<soapbind:fault name="BATCH_SCHEDULER_ERROR" use="literal" />
</fault>
<fault name="PERMISSION_DENIED">
<documentation>Permission denied</documentation>
<soapbind:fault name="PERMISSION_DENIED" use="literal" />
</fault>
<fault name="SESSIONKEY_EXPIRED">
<documentation>The sessionKey is expired. The session is closed.</documentation>
<soapbind:fault name="SESSIONKEY_EXPIRED" use="literal" />
</fault>
<fault name="UNKNOWN_BATCH_SCHEDULER">
<documentation>The batch scheduler type is unknown</documentation>
<soapbind:fault name="UNKNOWN_BATCH_SCHEDULER" use="literal" />
</fault>
<fault name="UNKNOWN_MACHINE">
<documentation>The machine id is unknown</documentation>
<soapbind:fault name="UNKNOWN_MACHINE" use="literal" />
</fault>
<fault name="UNDEFINED">
<documentation>Internal Error: Undefined exception</documentation>
<soapbind:fault name="UNDEFINED" use="literal" />
</fault>
<fault name="DIET">
<documentation>Vishnu not available (Service bus failure)</documentation>
```

<soapbind:fault name="DIET" use="literal" />
</fault>
<fault name="DBERR">
<documentation>Vishnu not available (Database error)</documentation>
<soapbind:fault name="DBERR" use="literal" />
</fault>
<fault name="DBCONN">
<documentation>Vishnu not available (Database connection)</documentation>
<soapbind:fault name="DBCONN" use="literal" />
</fault>
<fault name="SYSTEM">
<documentation>Vishnu not available (System)</documentation>
<soapbind:fault name="SYSTEM" use="literal" />
</fault>
<fault name="ALREADY_CANCELED">
<documentation>The job is already canceled</documentation>
<soapbind:fault name="ALREADY_CANCELED" use="literal" />
</fault>
<fault name="ALREADY_TERMINATED">
<documentation>The job is already terminated</documentation>
<soapbind:fault name="ALREADY_TERMINATED" use="literal" />
</fault>
<fault name="SSH">
<documentation>Vishnu not available (SSH error)</documentation>
<soapbind:fault name="SSH" use="literal" />
</fault>
</operation>
</binding>
<service name="VishnuTMSService">
<port binding="tns:VishnuTMSSOAPBinding" name="VishnuTMSPort">
<soapbind:address location="http://127.0.0.1:8080/ResourceProxy/VishnuTMS" />
</port>
</service>
</definitions>

About *wsdl-viewer.xsl*

This document was generated by libxslt XSLT engine. The engine processed the WSDL in XSLT 1.0 compliant mode.

This page has been generated by wsdl-viewer.xsl, version 3.1.01
Author: tomi vanek
Download at http://tomi.vanek.sk/xml/wsdl-viewer.xsl.

The transformation was inspired by the article
Uche Ogbuji: WSDL processing with XSLT
This page was generated by wsdl-viewer.xsl (http://tomi.vanek.sk)