# VISHNU D1.0 - Command line interface specifications

| COLLABORATORS | | | |
|---|---|---|---|

| | *TITLE* :<br><br>VISHNU D1.0 - Command line interface speci-<br>fications | | |
|---|---|---|---|
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | Benjamin Isnard, Daouda Traoré, Eugène Pamba Capo-Chichi, Kevin Coulomb, and Ibrahima Cissé | 6 janvier 2011 | |

| REVISION HISTORY | | | |
|---|---|---|---|

| NUMBER | DATE | DESCRIPTION | NAME |
|---|---|---|---|
| 0 | 05/01/2011 | Formatting example | B.Isnard |

# Table des matières

# Chapitre 1

# Document presentation

## 1.1 Document objectives

This document presents the detailed specifications of the Vishnu command line user interface.

These specifications include the definition of all commands. ###FIXME###

## 1.2 Document structure

The document is divided into 4 parts corresponding to the 4 modules that compose the Vishnu system :

– UMS : Users Management System
– TMS : Tasks Management System
– FMS : Files Management System
– IMS : Information Management System

Each module corresponds to a chapter in the document, and each chapter contains the following sections :
– A first section describing the definition of all commands provided by the package
– A second section describing the definition of all data types used ###FIXME###

## 1.3 References

TODO - ajouter ref. vers les specs générales

## 1.4 Glossary

# Chapitre 2

# API specification for User Management System (UMS)

## 2.1  Command definitions

### 2.1.1  connect

**Description**

The connect() function allows the user to open a session

**Example of use**

connect -p closePolicy -d sessionInactivityDelay -s substituteUserId login

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| login | string | | login represents the login of the user | YES |
| password | string | | password represents the password of the user, it is an interactive parameter | YES |
| closePolicy | SessionCloseType | p | is an option for closing session automatically | NO |
| sessionInactivityDelay | int | d | The sessionInactivityDelay is the maximum delay between two API commands when the CLOSE_ON_TIMEOUT option is set | NO |
| substituteUserId | string | s | is an option which allows an admin to open a session as he/she was a specific user identified by his/her userId | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_CLOSE_POLICY | closePolicy | SessionCloseType | is an option for closing session automatically | NO | YES |

### 2.1.2  reconnect

**Description**

The reconnect() function allows the user to get the sessionKey of a session in which he/she was disconnected previously without

closing it

**Example of use**

reconnect login sessionId

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| login | string | | login represents the login of the user | YES |
| password | string | | password represents the password of the user, it is an interactive parameter | YES |
| sessionId | string | | sessionId is the identifier of the session defined in the database | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| | | | | | |

### 2.1.3  createUser

**Description**

The createUser() function adds a new user in VISHNU

**Example of use**

createUser -w password -f firstname -l lastname -p privilege -m email -s passwordState userId

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| userId | string | | represents the login of the user | YES |
| password | string | w | is the password of the user. At the beginning, an admin can give a temporary password or it is automatically generated by the System. | NO |
| firstname | string | f | is the firstname of the user | NO |
| lastname | string | l | is the lastname of the user | NO |
| privilege | int | p | is the privilege of the user (admin or simple user) | NO |
| email | string | m | is the email of the user | NO |
| passwordState | int | s | is the state of the password which allows the System to inform the user to change his/her password | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.4  updateUser

**Description**

The updateUser() function updates the user information in VISHNU except login and password

**Example of use**

updateUser -f firstname -l lastname -p privilege -m email userId

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| userId | string | | represents the login of the user | YES |
| firstname | string | f | is an option to update the firstname of the user | NO |
| lastname | string | l | is an option to update the lastname of the user | NO |
| privilege | string | p | is an option to update the privilege of the user | NO |
| email | string | m | is an option to update the email adress of the user | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.5   deleteUser

**Description**

The deleteUser() function removes a user from VISHNU

**Example of use**

deleteUser login

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| login | string | | login represents the login of the user which will be deleted from VISHNU | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.6   close

**Description**

The close() function allows a user to close a session

**Example of use**

close

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.7  changePassword

**Description**

The changePassword() function allows the user to change his/her password

**Example of use**

changePassword login

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| login | string | | login represents the login of the user | YES |
| password | string | | password represents the password of the user, it is an interactive parameter | YES |
| passwordNew | string | | passwordNew represents the new password of the user, it is an interactive parameter | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| | | | | | |

### 2.1.8  resetPassword

**Description**

The resetPassword() function allows an admin to reset the password of a user identified by his/her userId

**Example of use**

resetPassword login

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| login | string | | login represents the login of the user | YES |
| passwordReset | string | | passwordReset represents the new value of the password to be reset, it is an interactive parameter | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.9 createLocalAccount

**Description**

The createLocalAccount() function allows the user to create a new local user config in VISHNU

**Example of use**

createLocalAccount -l acLogin -s sshKeyPath -h HomeDirectory userId machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| userId | string | | The userId represents the login of the user of the local user config | YES |
| machineId | string | | The MachineId represents the identifier of the machine associated to the local user config | YES |
| acLogin | string | l | accLogin represents the login of the user on the associated machine | NO |
| sshKeyPath | string | s | sshKeyPath is the path of the ssh key of the user on the associated machine | NO |
| HomeDirectory | string | h | HomeDirectory is the path of the home directory the user on the associated machine | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.10 updateLocalAccount

**Description**

The updateLocalAccount() function allows the user to update his/her local user configuration in VISHNU

**Example of use**

updateLocalAccount -l acLogin -s sshKeyPath -h HomeDirectory userId machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| userId | string | | The userId represents the login of the user of the local user config | YES |
| machineId | string | | The MachineId represents the identifier of the machine associated to the local user config | YES |
| acLogin | string | l | accLogin represents the login of the user on the associated machine | NO |
| sshKeyPath | string | s | sshKeyPath is the path of the ssh key of the user on the associated machine | NO |
| HomeDirectory | string | h | HomeDirectory is the path of the home directory the user on the associated machine | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.11   deleteLocalAccount

**Description**

The deleteLocalAccount() function allows to remove a local user configuration from VISHNU for a given user on a given machine

**Example of use**

deleteLocalAccount userId machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| userId | string | | userId represents the login of the user which will be deleted according to a specific machine | YES |
| machineId | string | | machineId represents the identifier of the machine which will be deleted according to a specific user | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.12   saveConfiguration

**Description**

The saveConfiguration() function allows an admin to save the configuration of VISHNU

**Example of use**

saveConfiguration

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.13  restoreConfiguration

**Description**

The restoreConfiguration() function allows to restores the configuration of VISHNU

**Example of use**

restoreConfiguration -d descConf

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| descConf | string | d | represents the description of the configuration | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.14  createMachine

**Description**

The createMachine() function adds a new machine in VISHNU

**Example of use**

createMachine -n name -s site -d totalDiskSpace -m totalMemory -t machineDescription machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| machineId | string | | represents the identifier of the machine | YES |
| name | string | n | represents the name of the machine | NO |
| site | string | s | represents the location of the machine | NO |
| totalDiskSpace | string | d | represents the total disk space of the machine | NO |
| totalMemory | string | m | represents the total memory of the machine | NO |
| machineDescription | string | t | represents the description of the machine | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.15  deleteMachine

**Description**

The deleteMachine() function removes a machine from VISHNU

**Example of use**

deleteMachine machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| machineId | string | | machineId represents the identifier of the machine | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

## 2.1.16  listLocalAccount

**Description**

The listLocalAccount() function allows the user to get an object which encapsulates the list of the local configuration objects according to the options selected

**Example of use**

listLocalAccount -a -u userId -i machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| AdminListOption | boolean | a | is an admin option for listing all information in the database during a list API commands | NO |
| userId | string | u | is an admin option for listing information about a specific user identified by his/her userId | NO |
| machineId | string | i | is an option which allows a user to list local user configuration on a specific machine | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

## 2.1.17  listMachine

**Description**

The listMachine() function allows the user to get an object which encapsulates the list of the machine objects according to the options selected

**Example of use**

listMachine -m machineId -a -u userId

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| machineId | string | m | is an option which allows a user to list local user configuration on a specific machine | NO |
| AdminListOption | boolean | a | is an admin option for listing all information in the database during a list API commands | NO |
| userId | string | u | is an admin option for listing information about a specific user identified by his/her userId | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.18 listHistoryCmd

**Description**

The listHistoryCmd() function allows the user to get an object which encapsulates the list of the commands objects according to the options selected. By default the commands of current session identified by the session key is listed

**Example of use**

listHistoryCmd -a -u userId -i sessionId -s startDateOption -e endDateOption

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| AdminListOption | boolean | a | is an admin option for listing all information in the database during a list API commands | NO |
| userId | string | u | is an admin option for listing information about a specific user identified by his/her userId | NO |
| sessionId | string | i | allows the user to list all commands launched within a specific session | NO |
| startDateOption | long | s | allows the user to organize the commands listed by providing the start date (the UNIX timestamp of the start date is used) | NO |
| endDateOption | long | e | allows the user to organize the commands listed by providing the end date (the timestamp of the end date is used). By default, the end date is the current day | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.19 listOptions

**Description**

The listOptions() function allows the user to get an object which encapsulates the list of the options

**Example of use**

listOptions -a -u userId -n optionName

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| AdminListOption | boolean | a | is an admin option for listing all information in the database during a list API commands | NO |
| userId | string | u | is an admin option for listing information about a specific user identified by his/her userId | NO |
| optionName | string | n | allows the user to list a specific option identified by its name | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

## 2.1.20   listUsers

**Description**

The listUsers() function allows the user to get an object which encapsulates the list of all users objects of VIHSNU

**Example of use**

listUsers -i userIdOption

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| userIdOption | string | i | allows an admin to list one user identified by his/her userId | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

## 2.1.21   listSessions

**Description**

The listSessions() function allows the user to get an object which encapsulates the list of the sessions objects or one session according to the options selected

**Example of use**

listSessions -t sessionListOption -p sessionClosePolicy -d sessionInactivityDelay -m machineId -a -u userId -i sessionId -s start-DateOption -e endDateOption

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| sessionListOption | SessionListOptionType | t | is an option which precise the type of session listed (ACTIVE, INACTIVE or ALL) | NO |
| sessionClosePolicy | SessionCloseType | p | is an option for closing session automatically | NO |
| sessionInactivityDelay | int | d | The sessionInactivityDelay is the maximum delay between two API commands when the CLOSE_ON_TIMEOUT option is set | NO |
| machineId | string | m | allows a user to list sessions opened on a specific machine | NO |
| AdminListOption | boolean | a | is an admin option for listing all information in the database during a list API commands | NO |
| userId | string | u | is an admin option for listing information about a specific user identified by his/her userId | NO |
| sessionId | string | i | allows the user to list all commands launched within a specific session | NO |
| startDateOption | long | s | allows the user to organize the commands listed by providing the start date (the UNIX timestamp of the start date is used) | NO |
| endDateOption | long | e | allows the user to organize the commands listed by providing the end date (the timestamp of the end date is used). By default, the end date is the current day | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

## 2.1.22  configureOption

**Description**

The configureOption() function allows the users to configure his/her options

**Example of use**

configureOption optionName value

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| optionName | string | | represents the name of an option | YES |
| value | string | | represents the value of an option | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

### 2.1.23  configureDefaultOption

**Description**

The configureDefaultOption() function allows an admin to configure a default option value

**Example of use**

configureDefaultOption optionName value

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| optionName | string | | represents the name of an option | YES |
| value | string | | represents the value of an option | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The sessionKey is the identifier of the session generated by VISHNU | YES | NO |

## 2.2   Parameters definitions

**SessionCloseType Enumeration Type**

| Name | Value |
|------|-------|
| CLOSE_ON_DISCONNECT | 0 |
| CLOSE_ON_TIMEOUT | 1 |

**SessionListOptionType Enumeration Type**

| Name | Value |
|------|-------|
| ACTIVE_SESSION | 0 |
| INACTIVE_SESSION | 1 |
| ALL_SESSION | 2 |

# Chapitre 3

# API specification for Tasks Management System (TMS)

## 3.1 Command definitions

### 3.1.1 submitJob

**Description**

The submitJob() function submits job on a machine through the use of a script (job_cmd_path). The script is a shell script which will be executed by a command shell such as sh or csh. The object opt parameter of this function allows the user to specifie job characterisques options. Also the user can options in the script file.

**Example of use**

submitJob -n Name -null priority -null queue -null wallTime -null memory -null nbCpu -null nbNodesAndCpuPerNode -null OutPutPath -null ErrorPath sessionKey machineId scriptFilePath

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| sessionKey | string | | The sessionKey is the identifier of the session generated by VISHNU | YES |
| machineId | string | | Is the id of the machine where the job must be submitted. | YES |
| scriptFilePath | string | | The path to the file containing the characteristics (job command, and batch scheduler directive required or optional) of the job to submit. | YES |
| Name | string | n | Assigns a job name. The default is the name of job path. | NO |
| priority | JobPriority | | Assigns priority of the job. | NO |
| queue | string | | Assigns the queue or class of the job. | NO |
| wallTime | int | | The maximum wall-clock time during which the job can run. | NO |
| memory | int | | The size of memory that the job will use. | NO |
| nbCpu | int | | The number of cpu that the job will use. | NO |
| nbNodesAndCpuPerNode | int | | The number of nodes and processors per node. | NO |
| OutPutPath | string | | Assigns the path and file for job output. | NO |
| ErrorPath | string | | Assigns the path and file for job error. | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
|      |           |      |             |          |             |

### 3.1.2   listJobs

**Description**

The listJobs() function gets a list of all submitted jobs

**Example of use**

listJobs -null JobId -null nbCpu -null fromSubmitDate -null toSubmitDate -null owner -null status -null priority -null OutPutPath -null ErrorPath -null queue sessionKey machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| sessionKey | string | | The sessionKey is the identifier of the session generated by VISHNU | YES |
| machineId | string | | Machine hash key | YES |
| JobId | string | | To list job which has this id. | NO |
| nbCpu | int | | To list jobs which have this number of cpu. | NO |
| fromSubmitDate | long | | List jobs submitted after this date (unix timestamp). | NO |
| toSubmitDate | long | | List jobs submitted before this date (unix timestamp) | NO |
| owner | string | | To list all jobs submitted by this owner. | NO |
| status | JobStatus | | To list jobs which have this status. | NO |
| priority | JobPriority | | To list jobs which have this priority | NO |
| OutPutPath | string | | Gets the path and file for each job output. | NO |
| ErrorPath | string | | Gets the path and file for each job error. | NO |
| queue | string | | To list jobs which have this queue name. | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
|      |           |      |             |          |             |

### 3.1.3   getJobInfo

**Description**

The getJobInfo() function gets information on a job from it Id

**Example of use**

getJobInfo sessionKey machineId jobId

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| sessionKey | string | | The sessionKey is the identifier of the session generated by VISHNU | YES |
| machineId | string | | Machine hash key | YES |
| jobId | string | | The id of the job | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
|      |           |      |             |          |             |

### 3.1.4 cancelJob

**Description**

The cancelJob() function cancels a job from it Id

**Example of use**

cancelJob sessionKey machineId jobId

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| sessionKey | string | | The sessionKey is the identifier of the session generated by VISHNU | YES |
| machineId | string | | Machine hash key | YES |
| jobId | string | | The Id of the job | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
|      |           |      |             |          |             |

### 3.1.5 getJobOutPut

**Description**

The getJobOutPut() function gets outputPath and errorPath of a job from it Id

**Example of use**

getJobOutPut sessionKey machineId jobId

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| sessionKey | string | | The sessionKey is the identifier of the session generated by VISHNU | YES |
| machineId | string | | Machine hash key | YES |
| jobId | string | | The Id of the job | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
|      |           |      |             |          |             |

### 3.1.6 getAllJobsOutPut

**Description**

The getAllJobsOutPut() function gets outputPath and errorPath of completed jobs dynamically

**Example of use**

getAllJobsOutPut sessionKey machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| sessionKey | string | | The sessionKey is the identifier of the session generated by VISHNU | YES |
| machineId | string | | Machine hash key | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| | | | | | |

### 3.1.7 listQueues

**Description**

The listQueues() function gets queues information

**Example of use**

listQueues sessionKey machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| sessionKey | string | | The sessionKey is the identifier of the session generated by VISHNU | YES |
| machineId | string | | Machine hash key | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| | | | | | |

### 3.1.8 setMachineEnv

**Description**

The setMachineEnv() function sets some environment variables

**Example of use**

setMachineEnv sessionKey machineId listEnv

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| sessionKey | string | | The sessionKey is the identifier of the session generated by VISHNU | YES |
| machineId | string | | Represents the machine id | YES |
| listEnv | string | | Represents the list environement variables | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| | | | | | |

### 3.1.9    setMachineRefreshPeriod

**Description**

The setMachineRefreshPeriod() function sets the refresh period of output and error file content

**Example of use**

setMachineRefreshPeriod sessionKey machineId value

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| sessionKey | string | | The sessionKey is the identifier of the session generated by VISHNU | YES |
| machineId | string | | The id of the machine | YES |
| value | int | | Is the refresh value | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| | | | | | |

### 3.1.10    getJobProgress

**Description**

The getJobProgress() function get the progression status of jobs.

**Example of use**

getJobProgress -null jobId -null jobOwner sessionKey machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| sessionKey | string | | The sessionKey is the identifier of the session generated by VISHNU | YES |
| machineId | string | | Is the id of the machine to get the jobs progression. | YES |
| jobId | string | | Represents the id of the job that the user wants to see its progression. | NO |
| jobOwner | string | | Represents the owner of the job. | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| | | | | | |

## 3.2    Parameters definitions

**JobPriority Enumeration Type**

| Name | Value |
|------|-------|
| VERY_LOW | 100 |
| LOW | 200 |
| NORMAL | 300 |
| HIGH | 400 |

| Name | Value |
|------|-------|
| VERY_HIGH | 500 |

**JobStatus Enumeration Type**

| Name | Value |
|------|-------|
| RUNNING | 0 |
| WAITING | 1 |
| COMPLETED | 2 |
| CANCELED | 3 |
| HELD | 4 |
| QUEUED | 5 |
| FAILED | 6 |
| NOT_SUBMITTED | 7 |

# Chapitre 4

# API specification for Information Management System (IMS)

## 4.1 Command definitions

### 4.1.1 getUpdateFrequency

**Description**

The getUpdateFrequency() function gets the update frequency of the IMS database

**Example of use**

getUpdateFrequency

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| | | | | |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.2 export

**Description**

The export() function exports all the commands made by a user during a session

**Example of use**

export oldSessionKey exportType

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| oldSessionKey | string | | The key of the session to export (session has ended) | YES |
| exportType | ExportType | | The format to export | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command (current session key) | YES | NO |

### 4.1.3 replay

**Description**

The replay() function replays a script of VISHNU commands

**Example of use**

replay filename type

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| filename | string | | A file containing the commands to replay | YES |
| type | ExportType | | The type of the script to execute (python, shell) | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.4 getMetricVal

**Description**

The getMetricVal() function gets data from a metric

**Example of use**

getMetricVal machineId startTime endTime metricType

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| machineId | string | | The id of the machine | YES |
| startTime | long | | The start time to look for the metric | YES |
| endTime | long | | The end time to search | YES |
| metricType | MetricType | | The metric types | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.5 getCurrentData

**Description**

The getCurrentData() function gets data corresponding to the datatype on a machine

**Example of use**

getCurrentData machineId dataType

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| machineId | string | | The id of the machine the user wants the total diskSpace | YES |
| dataType | MetricType | | The type of data the user wants to get | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.6 getProcesses

**Description**

The getProcesses() function gets the list of the processes running over a front machine

**Example of use**

getProcesses machineId

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| machineId | string | | The id of the machine the user wants the running processes | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.7 setSystemThreshold

**Description**

The setSystemThreshold() function sets a threshold on a machine of a system

**Example of use**

setSystemThreshold machineId tresholdType value

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| machineId | string | | The id of the machine the user wants to set the treshold over | YES |
| tresholdType | MetricType | | The type of the metric to set | YES |
| value | double | | The treshold value | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.8  getSystemThreshold

**Description**

The getSystemThreshold() function gets a System threshold on a machine

**Example of use**

getSystemThreshold machineId type

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| machineId | string | | The id of the machine the user wants to get the treshold over | YES |
| type | MetricType | | The treshold type desired | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|---|---|---|---|---|---|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.9  defineUserIdentifier

**Description**

The defineUserIdentifier() function defines the shape of the identifiers automatically generated for the users

**Example of use**

defineUserIdentifier format

**Parameters**

| Parameter | Type | Option | Description | Required |
|---|---|---|---|---|
| format | string | | The new format to use | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

## 4.1.10 defineMachineIdentifier

**Description**

The defineMachineIdentifier() function defines the shape of the identifiers automatically generated for the machines

**Example of use**

defineMachineIdentifier format

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| format | string | | The new format to use | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

## 4.1.11 defineJobIdentifier

**Description**

The defineJobIdentifier() function defines the shape of the identifiers automatically generated for the jobs

**Example of use**

defineJobIdentifier format

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| format | string | | The new format to use | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

## 4.1.12 defineTransferIdentifier

**Description**

The defineTransferIdentifier() function defines the shape of the identifiers automatically generated for the file transfers

## Example of use

defineTransferIdentifier format

## Parameters

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| format | string | | The new format to use | YES |

## Environment variables

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.13 loadShed

## Description

The loadShed() function loads shed a machine, 2 modes are available, soft or hard. Soft flushes the machine, hard stops the machine

## Example of use

loadShed machineId loadShedType

## Parameters

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| machineId | string | | The id of the machine to stop | YES |
| loadShedType | LoadShedType | | Type of load shedding | YES |

## Environment variables

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.14 setUpdateFrequency

## Description

The setUpdateFrequency() function sets the update frequency of the IMS tables

## Example of use

setUpdateFrequency freq

## Parameters

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| freq | int | | Frequency the data are updated, in second | YES |

## Environment variables

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.15  notifyOverflow

**Description**

The notifyOverflow() function sends a mail to the admin responsible for the limite whose treshold has been reached

**Example of use**

notifyOverflow machineId message

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| machineId | string | | The id of the machine with an overflow | YES |
| message | string | | The message to send | YES |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| | | | | | |

### 4.1.16  restart

**Description**

The restart() function restarts the whole VISHNU infrastructure. Actions are saved and restarted from the beginning once the infrastructure has been restarted

**Example of use**

restart -n componentName type

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| type | RestartType | t | The type of restart desired | YES |
| componentName | string | n | If not restarting all, the name of the component to restart | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

### 4.1.17  updateMachine

**Description**

The updateMachine() function updates the machines information in the database (diskspace and memory)

**Example of use**

updateMachine -l lang machineId CompoMachine

**Parameters**

| Parameter | Type | Option | Description | Required |
|-----------|------|--------|-------------|----------|
| machineId | string | | The id of the machine to update | YES |
| CompoMachine | CompoMachine | | A description of the type of component to change | YES |
| lang | string | l | For machine description, the language used | NO |

**Environment variables**

| Name | Parameter | Type | Description | Required | User option |
|------|-----------|------|-------------|----------|-------------|
| VISHNU_SESSION_KEY | sessionKey | string | The key of the session of the user that submits the command | YES | NO |

## 4.2  Parameters definitions

**ExportType Enumeration Type**

| Name | Value |
|------|-------|
| PYTHON | 0 |
| SHELL | 1 |

**MetricType Enumeration Type**

| Name | Value |
|------|-------|
| CPUNBR | 0 |
| CPUUSE | 1 |
| DISKSPACE | 2 |
| FREEDISKSPACE | 3 |
| MEMORY | 4 |
| FREEMEMORY | 5 |

**LoadShedType Enumeration Type**

| Name | Value |
|------|-------|
| SOFT | 0 |
| HARD | 1 |

**RestartType Enumeration Type**

| Name | Value |
|------|-------|
| ALL | 0 |
| SED | 1 |
| AGENT | 2 |
| DAEMON | 3 |
| MACHINE | 4 |

**CompoMachine Enumeration Type**

| Name | Value |
|------|-------|
| DESC | 0 |
| DISKSPACE | 1 |
| RAM | 2 |

| Name | Value |
|------|-------|
| DESC | 0 |
| DISKSPACE | 1 |
| RAM | 2 |