

VISHNU D1.0 - api specifications

COLLABORATORS

	<i>TITLE :</i> VISHNU D1.0 - api specifications		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Benjamin Isnard, Daouda Traoré, Eugène Pamba Capo-Chichi, and Kevin Coulomb	December 23, 2010	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
01	21/12/2010	Formatting example	B.Isnard
02	21/12/2010	Pre-delivable	B.Isnard

Contents

1	Document presentation	1
1.1	Document objectives	1
1.2	Document structure	1
1.3	References	1
1.4	Glossary	1
2	API specification for User Management System (UMS)	2
2.1	UMS specification	2
2.1.1	UMS_connect	2
2.1.2	UMS_reconnect	2
2.1.3	UMS_create_user	3
2.1.4	UMS_update_user	4
2.1.5	UMS_delete_user	4
2.1.6	UMS_close	5
2.1.7	UMS_change_password	5
2.1.8	UMS_reset_password	6
2.1.9	UMS_listsession	6
2.1.10	UMS_listuser	7
2.1.11	UMS_configure_option	7
2.1.12	UMS_history_commands	7
2.1.13	UMS_getSession	8
2.1.14	UMS_getUser	8
2.1.15	UMS_create_local_config	9
2.1.16	UMS_update_local_config	9
2.1.17	UMS_delete_local_config	10
2.1.18	UMS_list_local_config	10
2.1.19	UMS_getAccount	11
2.1.20	UMS_save_configuration	11
2.1.21	UMS_restore_configuration	12
2.1.22	Class definitions	12

3	API specification for Tasks Management System (TMS)	15
3.1	TMS specification	15
3.1.1	submitJob	15
3.1.2	listJobs	16
3.1.3	getJobInfos	17
3.1.4	cancelJob	17
3.1.5	getOutputJob	18
3.1.6	listQueues	18
3.1.7	setMachineEnv	19
3.1.8	setMachineRefreshPeriod	19
3.1.9	getMsgErrTMS	20
3.1.10	jobProgress	20
3.1.11	Class definitions	21
4	API specification for Information Management System (IMS)	24
4.1	IMS specification	24
4.1.1	ims_getUpdateFrequency	24
4.1.2	ims_export	24
4.1.3	ims_replay	25
4.1.4	ims_getMetricVal	25
4.1.5	ims_getCPUuse	26
4.1.6	ims_getNbCPU	26
4.1.7	ims_getDiskSpace	26
4.1.8	ims_getFreeDiskSpace	27
4.1.9	ims_getFreeMemory	27
4.1.10	ims_getMemory	28
4.1.11	ims_getProcesses	28
4.1.12	ims_setDiskSpaceSystemTreshold	29
4.1.13	ims_setMemorySystemTreshold	29
4.1.14	ims_setProcessesSystemTreshold	29
4.1.15	ims_getSystemTreshold	30
4.1.16	ims_defineUserIdentifier	30
4.1.17	ims_defineMachineIdentifier	31
4.1.18	ims_defineJobIdentifier	31
4.1.19	ims_defineTransferIdentifier	31
4.1.20	ims_hardDelest	32
4.1.21	ims_softDelest	32
4.1.22	ims_setUpdateFrequency	33
4.1.23	ims_notifyOverflow	33

4.1.24	ims_restart	33
4.1.25	ims_restartAgent	34
4.1.26	ims_restartSeD	34
4.1.27	ims_restartDaemon	34
4.1.28	Class definitions	35

Chapter 1

Document presentation

1.1 Document objectives

This document presents the external specifications of the Vishnu system at a general level. At this level, we describe . These general specifications are a prerequisite for the detailed specifications step in the software development process.

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 two sections:

- A first section containing "TMS api descriptions"
- A second section containing "UMS api descriptions"

1.3 References

1.4 Glossary

Chapter 2

API specification for User Management System (UMS)

2.1 UMS specification

2.1.1 UMS_connect

Parameters

Parameter	Type	Description	Mode	Required
login	string	login represents the login of the user	IN	yes
password	string	password represents the password of the user	IN	yes
options	ConnectOptions		IN	no
certificate	string	The certificate is the identifier of the session	OUT	yes

Description

Opening a new session

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
DB_BEGIN_ERROR	BEGIN command on database failed
DB_END_ERROR	END command on database failed
DB_DECLARE_CUSOR_ERROR	The DECLARATION of the CURSOR failed
DB_FETCH_ALL_ERROR	FETCH ALL failed
NOT_AUTHENTICATED	Unknown user

Signature

```
int UMS_connect(const string& login, const string& password, const ConnectOptions& options=ConnectOptions(), string& certificate);
```

2.1.2 UMS_reconnect

Parameters

Parameter	Type	Description	Mode	Required
login	string	login represents the login of the user	IN	yes
password	string	password represents the password of the user	IN	yes
numSession	int	numSession is the number of the session define in the database	IN	yes
certificate	string	certificate is the identifier of the session	OUT	yes

Description

Opening a session which is previously used without closing it

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
DB_BEGIN_ERROR	BEGIN command on database failed
DB_END_ERROR	END command on database failed
DB_DECLARE_CUSOR_ERROR	The DECLARATION of the CURSOR failed
DB_FETCH_ALL_ERROR	FETCH ALL failed
NOT_AUTHENTICATED	Unknown user

Signature

```
int UMS_reconnect(const string& login, const string& password, const int& numSession, string& certificate);
```

2.1.3 UMS_create_user

Parameters

Parameter	Type	Description	Mode	Required
newUser	User	newUser is the object which represents the new user	IN	yes
certificate	string	The certificate is an admin certificate of session	IN	yes

Description

Creation of a new user in Vishnu System

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
DB_BEGIN_ERROR	BEGIN command on database failed
DB_END_ERROR	END command on database failed
DB_DECLARE_CUSOR_ERROR	The DECLARATION of the CURSOR failed
DB_FETCH_ALL_ERROR	FETCH ALL failed
DB_QEXEC_INSERT_USER_ERROR	INSERT command on database USER failed
CERTIFICAT_NOT_FOUND	The certificate is unregonized
CERTIFICAT_EXPIRED	The certificate is expired
CERTIFICAT_INVALID	The certificate is not valid
NO_ADMIN	The user is not an administrator

Signature


```
int UMS_create_user(const User& newUser, const string& certificate);
```

2.1.4 UMS_update_user

Parameters

Parameter	Type	Description	Mode	Required
user	User	User is the object which represents the user	IN	yes
certificate	string	The certificate is an admin certificate of session	IN	yes

Description

Update of user information in Vishnu System

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
DB_BEGIN_ERROR	BEGIN command on database failed
DB_END_ERROR	END command on database failed
DB_DECLARE_CUSOR_ERROR	The DECLARATION of the CURSOR failed
DB_FETCH_ALL_ERROR	FETCH ALL failed
DB_QEXEC_INSERT_USER_ERROR	INSERT command on database USER failed
CERTIFICAT_NOT_FOUND	The certificate is unregonized
CERTIFICAT_EXPIRED	The certificate is expired
CERTIFICAT_INVALID	The certificate is not valid
NO_ADMIN	The user is not an administrator

Signature

```
int UMS_update_user(const User& user, const string& certificate);
```

2.1.5 UMS_delete_user

Parameters

Parameter	Type	Description	Mode	Required
login	string	login represents the login of the user	IN	yes
certificate	string	The certificate is an admin certificate of session	IN	yes

Description

Delete user from Vishnu

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
DB_BEGIN_ERROR	BEGIN command on database failed
DB_END_ERROR	END command on database failed
DB_DECLARE_CUSOR_ERROR	The DECLARATION of the CURSOR failed
DB_FETCH_ALL_ERROR	FETCH ALL failed
DB_QEXEC_INSERT_USER_ERROR	INSERT command on database USER failed

Name	Description
CERTIFICAT_NOT_FOUND	The certificate is unrecognized
CERTIFICAT_EXPIRED	The certificate is expired
CERTIFICAT_INVALID	The certificate is not valid
NO_ADMIN	The user is not an administrator

Signature

```
int UMS_delete_user(const string& login, const string& certificate);
```

2.1.6 UMS_close**Parameters**

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes

Description

Closing of session

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
DB_BEGIN_ERROR	BEGIN command on database failed
DB_END_ERROR	END command on database failed
DB_DECLARE_CUSOR_ERROR	The DECLARATION of the CURSOR failed
DB_FETCH_ALL_ERROR	FETCH ALL failed
CERTIFICAT_EXPIRED	The certificate is expired
CERTIFICAT_INVALID	The certificate is not valid
CERTIFICAT_NOT_FOUND	The certificate is unrecognized
SESSION_ALREADY_DISCONNECTED	à enlever même chose que CERTIFICAT_EXPIRED

Signature

```
int UMS_close(const string& certificate);
```

2.1.7 UMS_change_password**Parameters**

Parameter	Type	Description	Mode	Required
login	string	login represents the login of the user	IN	yes
password	string	password represents the password of the user	IN	yes
newPassword	string	newPassword represents the new password of the user	IN	yes

Description

Change of password

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
------	-------------

Signature

```
int UMS_change_password(const string& login, const string& password, const string& newPassword);
```

2.1.8 UMS_reset_password**Parameters**

Parameter	Type	Description	Mode	Required
login	string	login represents the login of the user	IN	yes
resetPassword	string	resetPassword represents the new value of the password to be reset	IN	yes
certificate	string	The certificate is the certificate of the session	IN	yes

Description

Reset of password

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
------	-------------

Signature

```
int UMS_reset_password(const string& login, const string& resetPassword, const string& certificate);
```

2.1.9 UMS_listsession**Parameters**

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes
sessionOption	string	option allows the user to list all sessions or only inactives. By default, only actives sessions are listed.	IN	yes
listsession	ListSessions	listsession is the list of sessions	OUT	yes

Description

List the table session

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
NOT_AUTHENTICATED	Unknown user
OK	

Signature

```
int UMS_listsession(const string& certificate, const string& sessionOption, ListSessions& listsession);
```

2.1.10 UMS_listuser

Parameters

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes
listuser	ListUsers	listuser is the list of users	OUT	yes

Description

list the table of users

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
NOT_AUTHENTICATED	Unknown user
OK	

Signature

```
int UMS_listuser(const string& certificate, ListUsers& listuser);
```

2.1.11 UMS_configure_option

Parameters

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes
nameOption	string	The nameOption is the name of the option to configure	IN	yes
newValue	string	newValue is the new value of the option configured	IN	yes

Description

This function allow the users to configure his options

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
------	-------------

Signature

```
int UMS_configure_option(const string& certificate, const string& nameOption, const string& newValue);
```

2.1.12 UMS_history_commands

Parameters

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes
listcommands	ListCommands	listcommands is the list of commands attached to sessionId	OUT	yes

Description

List all commands sent within a specific session

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
------	-------------

Signature

```
int UMS_history_commands(const string& certificate, ListCommands& listcommands);
```

2.1.13 UMS_getSession**Parameters**

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes
sessionID	int	sessionID is the short identifier of a specific session	IN	yes
sessionInfo	Session	The sessionInfo is the object which encapsulates the session information	OUT	yes

Description

This function returns the object which encapsulates information about a session

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
------	-------------

Signature

```
int UMS_getSession(const string& certificate, const int& sessionID, Session& sessionInfo);
```

2.1.14 UMS_getUser**Parameters**

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes
userID	int	iduser is the short identifier of a specific user	IN	yes
userInfo	User	The userInfo is the object which encapsulates the user information	OUT	yes

Description

This function returns the object which encapsulates information about a user

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
------	-------------

Signature

```
int UMS_getUser(const string& certificate, const int& userID, User& userInfo);
```

2.1.15 UMS_create_local_config**Parameters**

Parameter	Type	Description	Mode	Required
newAccount	User	newAccount is the object which encapsulates the new local user config	IN	yes
certificate	string	The certificate is a certificate of session	IN	yes

Description

Creation of a new local user config in Vishnu System

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
DB_BEGIN_ERROR	BEGIN command on database failed
DB_END_ERROR	END command on database failed
DB_DECLARE_CUSOR_ERROR	The DECLARATION of the CURSOR failed
DB_FETCH_ALL_ERROR	FETCH ALL failed
DB_QEXEC_INSERT_USER_ERROR	INSERT command on database USER failed
CERTIFICAT_NOT_FOUND	The certificate is unregonized
CERTIFICAT_EXPIRED	The certificate is expired
CERTIFICAT_INVALID	The certificate is not valid
NO_ADMIN	The user is not an administrator

Signature

```
int UMS_create_local_config(const User& newAccount, const string& certificate);
```

2.1.16 UMS_update_local_config**Parameters**

Parameter	Type	Description	Mode	Required
AccUpdated	User	AccUpdated is the object which encapsulates the change of the local user config	IN	yes
certificate	string	The certificate is a certificate of session	IN	yes

Description

Update of a local user config in Vishnu System

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
DB_BEGIN_ERROR	BEGIN command on database failed

Name	Description
DB_END_ERROR	END command on database failed
DB_DECLARE_CUSOR_ERROR	The DECLARATION of the CURSOR failed
DB_FETCH_ALL_ERROR	FETCH ALL failed
DB_QEXEC_INSERT_USER_ERROR	INSERT command on database USER failed
CERTIFICAT_NOT_FOUND	The certificate is unregonized
CERTIFICAT_EXPIRED	The certificate is expired
CERTIFICAT_INVALID	The certificate is not valid
NO_ADMIN	The user is not an administrator

Signature

```
int UMS_update_local_config(const User& AccUpdated, const string& certificate);
```

2.1.17 UMS_delete_local_config**Parameters**

Parameter	Type	Description	Mode	Required
localconfigID	int	localconfigID represents the short identifier of the local config	IN	yes
certificate	string	The certificate is a certificate of session	IN	yes

Description

Delete a local user config from Vishnu

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
DB_BEGIN_ERROR	BEGIN command on database failed
DB_END_ERROR	END command on database failed
DB_DECLARE_CUSOR_ERROR	The DECLARATION of the CURSOR failed
DB_FETCH_ALL_ERROR	FETCH ALL failed
DB_QEXEC_INSERT_USER_ERROR	INSERT command on database USER failed
CERTIFICAT_NOT_FOUND	The certificate is unregonized
CERTIFICAT_EXPIRED	The certificate is expired
CERTIFICAT_INVALID	The certificate is not valid
NO_ADMIN	The user is not an administrator

Signature

```
int UMS_delete_local_config(const int& localconfigID, const string& certificate);
```

2.1.18 UMS_list_local_config**Parameters**

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes
accountOption	string	option allows the user to list its accounts or all accounts when he/she is an admin	IN	yes
listaccount	ListAccounts	listsession is the list of sessions	OUT	yes

Description

List the local configs of the user

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
DB_CONN_ERROR	Connection to database failed
NOT_AUTHENTICATED	Unknown user
OK	

Signature

```
int UMS_list_local_config(const string& certificate, const string& accountOption, ListAccounts& listaccount);
```

2.1.19 UMS_getAccount**Parameters**

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes
accountID	int	accountID is the short identifier of a specific account	IN	yes
accountInfo	Session	The accountInfo is the object which encapsulates the account information	OUT	yes

Description

This function returns the object which encapsulates information about an account

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
------	-------------

Signature

```
int UMS_getAccount(const string& certificate, const int& accountID, Session& accountInfo);
```

2.1.20 UMS_save_configuration**Parameters**

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes
configuration	Configuration	The configuration is the object which encapsulates the configuration description	OUT	yes

Description

Save of configuration

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
------	-------------

Signature

```
int UMS_save_configuration(const string& certificate, Configuration& configuration);
```

2.1.21 UMS_restore_configuration**Parameters**

Parameter	Type	Description	Mode	Required
certificate	string	The certificate is the certificate of the session	IN	yes
configuration	Configuration	The configuration is the object which encapsulates the configuration description	IN	yes

Description

Restore of configuration

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
------	-------------

Signature

```
int UMS_restore_configuration(const string& certificate, const Configuration& configuration);
```

2.1.22 Class definitions**ConnectOptions Class Content**

Name	Type	Description
sessionClosePolicy	SessionClosePolicyType	option for closing session automatically
sessionInactivityDelay	int	TO DO

SessionClosePolicyType Enumeration Type

Name	Value
CLOSE_ON_DISCONNECT	0
CLOSE_ON_TIMEOUT	1

User Class Content

Name	Type	Description
userID	string	TO DO
login	string	TO DO
password	string	TO DO
firstname	string	TO DO
lastname	string	TO DO
status	int	TO DO
email	string	TO DO
passwordState	int	TO DO

ListSessions Class Content

Name	Type	Description
headerSession	string	TO DO
nFieldsSession	long	TO DO
nTuplesSession	long	TO DO
sessions	List of Session	TO DO

Session Class Content

Name	Type	Description
sessionID	string	TO DO
accountID	string	TO DO
dateLoading	string	TO DO
dateCreation	string	TO DO
certificate	string	TO DO
state	int	TO DO
closePolicy	string	TO DO
timeoutDate	string	TO DO

ListUsers Class Content

Name	Type	Description
headerUser	string	TO DO
nFieldsUser	long	TO DO
nTuplesUser	long	TO DO
users	List of User	TO DO

ListCommands Class Content

Name	Type	Description
headerCmd	string	TO DO
nFieldsCmd	long	TO DO
nTuplesCmd	long	TO DO
Commands	List of Command	TO DO

Command Class Content

Name	Type	Description
commandID	string	TO DO
sessionID	string	TO DO
cmdDescription	string	TO DO
cmdStartTime	string	TO DO
cmdEndTime	string	TO DO

ListAccounts Class Content

Name	Type	Description
headerAccount	string	TO DO
nFieldsAccount	long	TO DO
nTuplesAccount	long	TO DO
accounts	List of Account	TO DO

Account Class Content

Name	Type	Description
accountID	string	TO DO
userID	string	TO DO
machineID	string	TO DO
acLogin	string	TO DO
sshKeyPath	string	TO DO
HomeDirectory	string	TO DO

Configuration Class Content

Name	Type	Description
nameConfiguration	string	TO DO
listConfUsers	ListUsers	TO DO
listConfSessions	ListSessions	TO DO
listConfMachines	ListMachines	TO DO

ListMachines Class Content

Name	Type	Description
headerMachine	string	TO DO
nFieldsMachine	long	TO DO
nTuplesMachine	long	TO DO
machines	List of Machine	TO DO

Machine Class Content

Name	Type	Description
machineID	string	TO DO
name	string	TO DO
site	string	TO DO
totalDiskSpace	string	TO DO
totalMemory	string	TO DO

Chapter 3

API specification for Tasks Management System (TMS)

3.1 TMS specification

3.1.1 submitJob

Parameters

Parameter	Type	Description	Mode	Required
sessionId	string	Is the id of the session opened by the user.	IN	yes
machineId	string	Is the id of the machine where the job must be submitted.	IN	yes
job_cmd_path	string	The path to the file containing the characteristics (job command, and batch scheduler directive required or optional) of the job to submit.	IN	yes
jobId	string	Is the returned id of the submitted job.	OUT	yes
options	SubmitOptions	Is an instance of the class SubmitOptions. Each optionnal value is associated to a set operation (e.g: setNbCpu(...)) in the class SubmitOptions. If no set operation is not called on the instance object options, the job is submitted with the options defined in the job_cmd_path. Otherwise the job is submitted with the optionnal values set by the options object and optionnal values defined in the job_cmd_path, but optionnal values set by SubmitOptions object take precedence over those in job_cmd_path. With in the object options or within the job_cmd_path, the last occurrence of an optionnal value takes precedence over earlier occurrence.	IN	no

Description

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.

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
TMS_OK	The service was performed successfully.

Name	Description
TMS_UNKNOWN_MACHINE	The machine is not known.
TMS_BATCH_SCHEDULER_ERROR	Indicates an error caused by the underlying batch scheduler
TMS_INVALID_PATH	The path to the file containing the characteristics of the job to submit is not a valid path
TMS_INVALID_REPONSE	Indicates that the implementation produced a response that does not match the criteria defined by the specification.
TMS_INVALID_REQUEST	Indicates that the request is not valid.
TMS_INVALID_SESSION_ID	The session id is not valid to perform the service.
TMS_NOT_ALLOW	Indicates the requested operation is not allowed for provided user.
TMS_SERVER_NOT_AVAILABLE	Indicates that the task management server is not available.
TMS_SUBMIT_SERVICE_NOT_AVAILABLE	Indicates that the service to perform the submit operation is not found.
TMS_UNKNOWN_BATCH_SCHEDULER_TYPE	Indicates that the batch scheduler type is not known.
TMS_UNKNOWN_QUEUE	Indicates that the specified queue by the user is not known.

Signature

int **submitJob**(const string& sessionId, const string& machineId, const string& job_cmd_path, string& jobId, const SubmitOptions& options=SubmitOptions());

3.1.2 listJobs

Parameters

Parameter	Type	Description	Mode	Required
sessionId	string	The session Id of the user	IN	yes
machineId	string	Machine hash key	IN	yes
listOfJobs	ListJobs	The constructed object list of jobs	OUT	yes
options	ListJobsOptions	Additional options for jobs listing	IN	no

Description

Gets a list of all submitted jobs

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
TMS_BATCH_SCHEDULER_ERROR	Indicates an error caused by the underlying batch scheduler
TMS_INVALID_REPONSE	Indicates that the implementation produced a response that does not match the criteria defined by the specification.
TMS_INVALID_REQUEST	Indicates that the request is not valid.
TMS_INVALID_SESSION_ID	The session id is not valid to perform the service.
TMS_NOT_ALLOW	Indicates the requested operation is not allowed for provided user.
TMS_OK	The service was performed successfully.
TMS_SERVER_NOT_AVAILABLE	Indicates that the task management server is not available.
TMS_UNKNOWN_BATCH_SCHEDULER_TYPE	Indicates that the batch scheduler type is not known.
TMS_UNKNOWN_MACHINE	The machine is not known.

Signature

int **listJobs**(const string& sessionId, const string& machineId, ListJobs& listOfJobs, const ListJobsOptions& options=ListJobsOptions());

3.1.3 getJobInfos

Parameters

Parameter	Type	Description	Mode	Required
sessionId	string	The session Id of the user	IN	yes
machineId	string	Machine hash key	IN	yes
jobId	string	The id of the job	IN	yes
jobInfos	Job	The resulting information on the job	OUT	yes

Description

Gets information on a job from it Id

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
TMS_BATCH_SCHEDULER_ERROR	Indicates an error caused by the underlying batch scheduler
TMS_INVALID_REQUEST	Indicates that the request is not valid.
TMS_INVALID_REPONSE	Indicates that the implementation produced a response that does not match the criteria defined by the specification.
TMS_INVALID_SESSION_ID	The session id is not valid to perform the service.
TMS_NOT_ALLOW	Indicates the requested operation is not allowed for provided user.
TMS_OK	The service was performed successfully.
TMS_SERVER_NOT_AVAILABLE	Indicates that the task management server is not available.
TMS_UNKNOWN_BATCH_SCHEDULER_TYPE	Indicates that the batch scheduler type is not known.
TMS_UNKNOWN_MACHINE	The machine is not known.

Signature

```
int getJobInfos(const string& sessionId, const string& machineId, const string& jobId, Job& jobInfos);
```

3.1.4 cancelJob

Parameters

Parameter	Type	Description	Mode	Required
sessionId	string	The session Id of the user	IN	yes
machineId	string	Machine hash key	IN	yes
jobId	string	The Id of the job	IN	yes
infoMsg	string	The information message	OUT	yes

Description

Cancels a job from it Id

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
TMS_BATCH_SCHEDULER_ERROR	Indicates an error caused by the underlying batch scheduler
TMS_INVALID_REPONSE	Indicates that the implementation produced a response that does not match the criteria defined by the specification.
TMS_INVALID_REQUEST	Indicates that the request is not valid.

Name	Description
TMS_INVALID_SESSION_ID	The session id is not valid to perform the service.
TMS_NOT_ALLOW	Indicates the requested operation is not allowed for provided user.
TMS_OK	The service was performed successfully.
TMS_SERVER_NOT_AVAILABLE	Indicates that the task management server is not available.
TMS_UNKNOWN_BATCH_SCHEDULER_TYPE	Indicates that the batch scheduler type is not known.
TMS_UNKNOWN_MACHINE	The machine is not known.

Signature

```
int cancelJob(const string& sessionId, const string& machineId, const string& jobId, string& infoMsg);
```

3.1.5 getOutputJob**Parameters**

Parameter	Type	Description	Mode	Required
sessionId	string	The session Id of the user	IN	yes
machineId	string	Machine hash key	IN	yes
jobId	string	The Id of the job	IN	yes
outputPath	string	The path of the file containinig the output result of the job	OUT	yes
errorPath	string	The path of the file containinig the errors that has been occurred during the execution of the job	OUT	yes

Description

Gets outputPath and errorPath of a job from it Id

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
TMS_BATCH_SCHEDULER_ERROR	Indicates an error caused by the underlying batch scheduler
TMS_INVALID_REPONSE	Indicates that the implementation produced a response that does not match the criteria defined by the specification.
TMS_INVALID_REQUEST	Indicates that the request is not valid.
TMS_INVALID_SESSION_ID	The session id is not valid to perform the service.
TMS_OK	The service was performed successfully.
TMS_NOT_ALLOW	Indicates the requested operation is not allowed for provided user.
TMS_SERVER_NOT_AVAILABLE	Indicates that the task management server is not available.
TMS_UNKNOWN_BATCH_SCHEDULER_TYPE	Indicates that the batch scheduler type is not known.
TMS_UNKNOWN_MACHINE	The machine is not known.

Signature

```
int getOutputJob(const string& sessionId, const string& machineId, const string& jobId, string& outputPath, string& errorPath);
```

3.1.6 listQueues**Parameters**

Parameter	Type	Description	Mode	Required
sessionId	string	The session Id of the user	IN	yes
machineId	string	Machine hash key	IN	yes
listofQueues	string	The list of queues	OUT	yes

Description

Gets queues information

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
TMS_BATCH_SCHEDULER_ERROR	Indicates an error caused by the underlying batch scheduler
TMS_INVALID_REPONSE	Indicates that the implementation produced a response that does not match the criteria defined by the specification.
TMS_INVALID_REQUEST	Indicates that the request is not valid.
TMS_INVALID_SESSION_ID	The session id is not valid to perform the service.
TMS_NOT_ALLOW	Indicates the requested operation is not allowed for provided user.
TMS_OK	The service was performed successfully.
TMS_SERVER_NOT_AVAILABLE	Indicates that the task management server is not available.
TMS_UNKNOWN_BATCH_SCHEDULER_TYPE	Indicates that the batch scheduler type is not known.
TMS_UNKNOWN_MACHINE	The machine is not known.

Signature

```
int listQueues(const string& sessionId, const string& machineId, string& listofQueues);
```

3.1.7 setMachineEnv

Parameters

Parameter	Type	Description	Mode	Required
machineId	string		IN	yes

Description

Sets some environment variables

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
TMS_OK	The service was performed successfully.
TMS_UNKNOWN_MACHINE	The machine is not known.

Signature

```
int setMachineEnv(const string& machineId);
```

3.1.8 setMachineRefreshPeriod

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine	IN	yes

Description

Sets the refresh period of output and error file content

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
TMS_OK	The service was performed successfully.
TMS_UNKNOWN_MACHINE	The machine is not known.

Signature

```
int setMachineRefreshPeriod(const string& machineId);
```

3.1.9 getMsgErrTMS**Parameters**

Parameter	Type	Description	Mode	Required
errorType	int	The type of error	IN	yes

Description

Prints the error message

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
TMS_UNKNOWN_TYPE_OF_ERROR	The type of error is not known

Signature

```
int getMsgErrTMS(const int& errorType);
```

3.1.10 jobProgress**Parameters**

Parameter	Type	Description	Mode	Required
sessionId	string	Is the id of the session opened by the user.	IN	yes
machineId	string	Is the id of the machine to get the jobs progression.	IN	yes
progress	Progression	Is the object containing jobs progression information	OUT	yes
options	ProgressOptions	Is an object containing the available options jobs for progression .	IN	no

Description

Prints the progression status of jobs.

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
TMS_OK	The service was performed successfully.
TMS_INVALID_REQUEST	Indicates that the request is not valid.
TMS_UNKNOWN_MACHINE	The machine is not known.
TMS_INVALID_SESSION_ID	The session id is not valid to perform the service.
TMS_SERVER_NOT_AVAILABLE	Indicates that the task management server is not available.
TMS_SUBMIT_SERVICE_NOT_AVAILABLE	Indicates that the service to perform the submit operation is not found.

Signature

int **jobProgress**(const string& sessionId, const string& machineId, Progression& progress, const ProgressOptions& options=ProgressOp

3.1.11 Class definitions

SubmitOptions Class Content

Name	Type	Description
Name	string	Assigns a job name. The default is the name of job path.
command	string	The command to perform the job.
priority	JobPriority	Assigns priority of the job.
queue	string	Assigns the queue or class of the job.
wallTime	int	The maximum wall-clock time during which the job can run.
memory	int	The size of memory that the job will use.
nbCpu	int	The number of cpu that the job will use.
nbNodesAndCpuPerNode	int	The number of nodes and processors per node.
OutPutPath	string	Assigns the path and file for job output.
ErrorPath	string	Assigns the path and file for job error.

JobPriority Enumeration Type

Name	Value
VERY_LOW	100
LOW	200
NORMAL	300
HIGH	400
VERY_HIGH	500

ListJobs Class Content

Name	Type	Description
nbJobs	long	Represents the total number of jobs in the list.
nbRunningJobs	long	Represents of running jobs in the list.
nbWaitingJobs	long	Represents the total number of waiting jobs in the list.
jobs	List of Job	Is a list of job information (jobId, jobName, ...).

Job Class Content

Name	Type	Description
TMSSessionId	string	Is the session id associated to the job.

Name	Type	Description
submitMachineId	string	Is the id of the machine on which the job has been submitted.
submitMachine	string	Is the name of the machine on which the job has been submitted.
jobId	string	Represents the id to job.
jobName	string	Represents the name assigned to the job.
jobPath	string	Is the path to the file containing job characteristics.
outputPath	string	Is the path to the job output results.
errorPath	string	Is the path to the file containing errors occurred during job's execution.
jobPrio	JobPriority	Represents the job priority.
nbCpus	int	Is the number of cpu used by the job.
jobWorkingDir	string	Indicates the directory where the job has been launched.
state	JobStatus	The current state of the job.
submitDate	string	Represents the submission date.
endDate	long	Represents the execution end date of the job.
owner	string	Represents the job owner.
jobQueue	string	Is the name of the queue or class associated to the job.
wallClockLimit	long	Is the maximum wall-clock time during which the job can run.
groupName	string	Represents the job owner group name.
jobDescription	string	Is the textual description of the job.
memLimit	int	Represents the memory size limit of the job.
nbNodes	int	Is the total number of nodes used by the job.
nbNodesAndCpuPerNode	int	Is the number of nodes and processors per node used by the job.

JobStatus Enumeration Type

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

ListJobsOptions Class Content

Name	Type	Description
JobId	string	To list job which has this id.
nbCpu	int	To list jobs which have this number of cpu.
fromSubmitDate	int	To list jobs which have submitted from this date.
toSubmitDate	int	To list jobs which have submitted to this date.
owner	string	To list all jobs submitted by this owner.
status	JobStatus	To list jobs which have this status.
priority	JobPriority	To list jobs which have this priority
OutPutPath	string	Gets the path and file for each job output.
ErrorPath	string	Gets the path and file for each job error.
queue	string	To list jobs which have this queue name.

Progression Class Content

Name	Type	Description
jobId	string	TO DO
jobName	string	TO DO
wallTime	int	TO DO
startTime	int	TO DO
endTime	int	TO DO
ratio	double	TO DO

ProgressOptions Class Content

Name	Type	Description
jobId	string	TO DO
jobOwner	string	TO DO

Chapter 4

API specification for Information Management System (IMS)

4.1 IMS specification

4.1.1 ims_getUpdateFrequency

Parameters

Parameter	Type	Description	Mode	Required
freq	int	Frequency the data are updated, in second	OUT	yes

Description

To get the update frequency of the IMS database

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_getUpdateFrequency(int& freq);
```

4.1.2 ims_export

Parameters

Parameter	Type	Description	Mode	Required
sessionId	string	The id of the session to export	IN	yes
exportType	exportType	The format to export	IN	yes
filename	string	The file containing the commands	OUT	yes

Description

To export all the commands made by a user during a session

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success
ims_databaseInternalError	An error occured with the database

Signature

```
int ims_export(const string& sessionId, const exportType& exportType, string& filename);
```

4.1.3 ims_replay**Parameters**

Parameter	Type	Description	Mode	Required
filename	string	A file containing the commands to replay	IN	yes
type	exportType	The type of the script to execute (python, shell)	IN	yes

Description

To replay a script of VISHNU commands

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_replay(const string& filename, const exportType& type);
```

4.1.4 ims_getMetricVal**Parameters**

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine	IN	yes
startTime	Date	The start time to look for the metric	IN	yes
endTime	Date	The end time to search	IN	yes
metricType	metricType	The metric types	IN	yes
res	metric	A list of corresponding results	OUT	yes

Description

To get data from a metric

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_getMetricVal(const string& machineId, const Date& startTime, const Date& endTime, const metricType& metricType, metric& res);
```

4.1.5 ims_getCPUuse**Parameters**

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants the CPU use	IN	yes
res	double	Percentage of use of the CPU	OUT	yes

Description

To get the percentage of use of the CPUs on a machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_getCPUuse(const string& machineId, double& res);
```

4.1.6 ims_getNbCPU**Parameters**

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants the number of CPU	IN	yes
res	int	The number of CPU	OUT	yes

Description

To get the number of CPUs on a machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_getNbCPU(const string& machineId, int& res);
```

4.1.7 ims_getDiskSpace**Parameters**

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants the total diskSpace	IN	yes

Parameter	Type	Description	Mode	Required
res	double	The total diskSpace on the machine	OUT	yes

Description

To get the total diskSpace on a machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_getDiskSpace(const string& machineId, double& res);
```

4.1.8 ims_getFreeDiskSpace**Parameters**

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants to get the free diskSpace	IN	yes
res	double	The free diskSpace on the machine	OUT	yes

Description

To get the free diskSpace on a machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_getFreeDiskSpace(const string& machineId, double& res);
```

4.1.9 ims_getFreeMemory**Parameters**

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants the free memory	IN	yes
res	double	The free memory on the machine	OUT	yes

Description

To get the free memory on a machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_getFreeMemory(const string& machineId, double& res);
```

4.1.10 ims_getMemory**Parameters**

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants the total memory	IN	yes
res	double	The total memory on the machine	OUT	yes

Description

To get the total memory on a machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_getMemory(const string& machineId, double& res);
```

4.1.11 ims_getProcesses**Parameters**

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants the running processes	IN	yes
process	int	The total memory on the machine	OUT	yes

Description

To get the list of the processes running over a front machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_getProcesses(const string& machineId, int& process);
```

4.1.12 ims_setDiskSpaceSystemTreshold

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants to set the treshold over	IN	yes
value	double	The treshold value	IN	yes

Description

To define the free diskSpace System treshold on a machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_setDiskSpaceSystemTreshold(const string& machineId, const double& value);
```

4.1.13 ims_setMemorySystemTreshold

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants to set the treshold over	IN	yes
value	double	The treshold value	IN	yes

Description

To define the free memory System treshold on a machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_setMemorySystemTreshold(const string& machineId, const double& value);
```

4.1.14 ims_setProcessesSystemTreshold

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants to set the treshold over	IN	yes
value	int	The treshold value	IN	yes

Description

To define the processes number System treshold on a machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_setProcessesSystemTreshold(const string& machineId, const int& value);
```

4.1.15 ims_getSystemTreshold**Parameters**

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants to get the treshold over	IN	yes
type	metricType	The treshold type desired	IN	yes
value	double	The treshold value	OUT	yes

Description

To get a System treshold on a machine

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_getSystemTreshold(const string& machineId, const metricType& type, double& value);
```

4.1.16 ims_defineUserIdentifier**Parameters**

Parameter	Type	Description	Mode	Required
format	string	The new format to use	IN	yes

Description

To define the shape of the identifiers automatically generated for the users

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_defineUserIdentifier(const string& format);
```

4.1.17 ims_defineMachineIdentifier**Parameters**

Parameter	Type	Description	Mode	Required
format	string	The new format to use	IN	yes

Description

To define the shape of the identifiers automatically generated for the machines

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_defineMachineIdentifier(const string& format);
```

4.1.18 ims_defineJobIdentifier**Parameters**

Parameter	Type	Description	Mode	Required
format	string	The new format to use	IN	yes

Description

To define the shape of the identifiers automatically generated for the jobs

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_defineJobIdentifier(const string& format);
```

4.1.19 ims_defineTransferIdentifier**Parameters**

Parameter	Type	Description	Mode	Required
format	string	The new format to use	IN	yes

Description

To define the shape of the identifiers automatically generated for the file transfers

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_defineTransferIdentifier(const string& format);
```

4.1.20 ims_hardDelest

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine to stop	IN	yes

Description

To abruptly stop a machine. The stopped jobs, commands are not saved to be restarted later

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_hardDelest(const string& machineId);
```

4.1.21 ims_softDelest

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine	IN	yes

Description

To purge a machine. The purged actions are stored and can be restarted from the beginning later

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_softDelest(const string& machineId);
```

4.1.22 ims_setUpdateFrequency

Parameters

Parameter	Type	Description	Mode	Required
freq	int	Frequency the data are updated, in second	IN	yes

Description

To set the update frequency of the IMS database

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_setUpdateFrequency(const int& freq);
```

4.1.23 ims_notifyOverflow

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine with an overflow	IN	yes
message	string	The message to send	IN	yes

Description

To send a mail to the admin responsible for the limite whose treshold has been reached

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_notifyOverflow(const string& machineId, const string& message);
```

4.1.24 ims_restart

Parameters

Parameter	Type	Description	Mode	Required
-----------	------	-------------	------	----------

Description

To restart the whole VISHNU infrastructure. Actions are saved and restarted from the beginning once the infrastructure has been restarted

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_restart
```

4.1.25 ims_restartAgent

Parameters

Parameter	Type	Description	Mode	Required
name	string	The name of the agent	IN	yes

Description

To restart an agent from the System

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_restartAgent(const string& name);
```

4.1.26 ims_restartSeD

Parameters

Parameter	Type	Description	Mode	Required
name	string	The name of the SeD	IN	yes

Description

To restart a SeD from the System

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_restartSeD(const string& name);
```

4.1.27 ims_restartDaemon

Parameters

Parameter	Type	Description	Mode	Required
name	string	The name of the daemon	IN	yes

Description

To restart a daemon needed by the System.

Return Value

An error code is returned when an error occurs during the execution of the function.

Name	Description
ims_success	Error code returned if success

Signature

```
int ims_restartDaemon(const string& name);
```

4.1.28 Class definitions

exportType Enumeration Type

Name	Value
PYTHON	0
SHELL	1

Date Class Content

Name	Type	Description
year	int	The last 2 digits of the year
month	int	The month (from 1 to 12)
day	int	The day (from 1 to 31)
hour	int	The hour (from 0 to 23)
minute	int	The minute (from 1 to 59)
second	int	The second, from 0 to 99

metricType Enumeration Type

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

metric Class Content

Name	Type	Description
type	metricType	TO DO
value	double	TO DO
time	Date	TO DO