VISHNU D1.0 - api specifications

COLLABORATORS			
	TITLE : VISHNU D1.0 - api specifications		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Benjamin Isnard, Daouda Traoré, Eugène Pamba Capo-Chichi, Kevin Coulomb, and Ibrahima Cissé	December 24, 2010	

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

Contents

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 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	options is an object which encapsulates the options available for the connect method	IN	no
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	OUT	yes

Description

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

Return Value

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

Name	Description
UMS_OK	The command completed successfully
NOT_AUTHENTICATED	Unknown user
UNKNOWN_CLOSURE_MODE	The name of the closure mode is unknown
INCORRECT_TIMEOUT	The value of the timeout is incorrect (negative or higher than the treshold)
INCORRECT_PASSSWORD_SIZE	The size of the password is incorrect
INCORRECT_LOGIN_SIZE	The size of the login is incorrect
DB_CONNECT_ERROR	Connection to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_COMMIT_ERROR	command commit failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **connect**(const string& login, const string& password, const ConnectOptions& options=ConnectOptions(), string& sessionKey);

2.1.2 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
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	OUT	yes

Description

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

Return Value

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

Name	Description
UMS_OK	The command completed successfully
NOT_AUTHENTICATED	Unknown user
INCORRECT_LOGIN_SIZE	The size of the login is incorrect
INCORRECT_PASSSWORD_SIZE	The size of the password is incorrect
SESSION_INCOMPATIBILITY	This session identifier is incompatible with the authenticated user
DB_CONNECT_ERROR	Connection to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_COMMIT_ERROR	command commit failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **reconnect**(const string& login, const string& password, const int& numSession, string& sessionKey);

2.1.3 createUser

Parameters

Parameter	Type	Description	Mode	Required
newUser	User	newUser is an object which encapsulates the new user information	IN	yes
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Description

The createUser() function adds a new user in VISHNU

Return Value

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

Name	Description
UMS_OK	The command completed successfully
NO_ADMIN	The user is not an administrator
INCORRECT_LOGIN_SIZE	The size of the login is incorrect
INCORRECT_PASSSWORD_SIZE	The size of the password is incorrect
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
LOGIN_EXISTING	The login already exists in the database
INVALID_MAIL_ADRESS	The mail adress is invalid
DB_CONNECT_ERROR	Connection to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_COMMIT_ERROR	command commit failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **createUser**(const User& newUser, const string& sessionKey);

2.1.4 updateUser

Parameters

Parameter	Type	Description	Mode	Required
user	User	User is the object which encapsulates user information updated	IN	yes
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Description

The updateUser() function updates the user information in VISHNU

Return Value

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

Name	Description
NO_ADMIN	The user is not an administrator
NOT_AUTHENTICATED	Unknown user
INVALID_MAIL_ADRESS	The mail adress is invalid
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
DB_CONNECT_ERROR	Connection to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_COMMIT_ERROR	command commit failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **updateUser**(const User& user, const string& sessionKey);

2.1.5 deleteUser

Parameters

Parameter	Type	Description	Mode	Required
login	string	login represents the login of the user which will be deleted from VISHNU	IN	yes
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Description

The deleteUser() function deletes the user from VISHNU

Return Value

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

Name	Description
UMS_OK	The command completed successfully
NOT_AUTHENTICATED	Unknown user
NO_ADMIN	The user is not an administrator
INCORRECT_LOGIN_SIZE	The size of the login is incorrect
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
DB_CONNECT_ERROR	Connection to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_COMMIT_ERROR	command commit failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **deleteUser**(const string& login, const string& sessionKey);

2.1.6 close

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Description

The close() function allows to close a session

Return Value

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

Name	Description
UMS_OK	The command completed successfully
COMMAND_RUNNING	Command(s) is/are running
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
DB_CONNECT_ERROR	Connection to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_COMMIT_ERROR	command commit failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_ROLLBACK_ERROR	Rollback command to database failed

Signature

int **close**(const string& sessionKey);

2.1.7 changePassword

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

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

Return Value

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

Name	Description
UMS_OK	The command completed successfully
NOT_AUTHENTICATED	Unknown user
INCORRECT_LOGIN_SIZE	The size of the login is incorrect
INCORRECT_PASSSWORD_SIZE	The size of the password is incorrect
DB_CONNECT_ERROR	Connection to database failed
DB_COMMIT_ERROR	command commit failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **changePassword**(const string& login, const string& password, const string& newPassword);

2.1.8 resetPassword

Parameters

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

Parameter	Type	Description	Mode	Required
passwordReset	string	passwordReset represents the new value of the password to be reset	IN	yes
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Description

The resetPassword() function allows the user to reset his/her password

Return Value

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

Name	Description
UMS_OK	The command completed successfully
NO_ADMIN	The user is not an administrator
NOT_AUTHENTICATED	Unknown user
INCORRECT_LOGIN_SIZE	The size of the login is incorrect
INCORRECT_PASSSWORD_SIZE	The size of the password is incorrect
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
DB_COMMIT_ERROR	command commit failed
DB_CONNECT_ERROR	Connection to database failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **resetPassword**(const string& login, const string& passwordReset, const string& sessionKey);

2.1.9 displaySessions

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
options	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

The displaySessions() function diplays the sessions according to the options selected

Return Value

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

Name	Description
UMS_OK	The command completed successfully
NO_ADMIN	The user is not an administrator
UNKNOWN_SESSIONOPTION	the name of the session option is unknown

Name	Description
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
DB_CONNECT_ERROR	Connection to database failed
DB_COMMIT_ERROR	command commit failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **displaySessions**(const string& sessionKey, const string& options, ListSessions& listsession);

2.1.10 displayUsers

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
listuser	ListUsers	listuser is the list of users	OUT	yes

Description

The displayUsers() function displays the list of all users of VIHSNU

Return Value

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

Name	Description
UMS_OK	The command completed successfully
NO_ADMIN	The user is not an administrator
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
DB_CONNECT_ERROR	Connection to database failed
DB_COMMIT_ERROR	command commit failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **displayUsers**(const string& sessionKey, ListUsers& listuser);

2.1.11 configureOption

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Parameter	Type	Description	Mode	Required
nameOption	string	The nameOption is the name of the option to configure	IN	yes
valueOption	string	valueOption is the new value of the option to configure	IN	yes

Description

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

Return Value

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

Name	Description
UMS_OK	The command completed successfully
UNKNOWN_CONFIGUREOPTION	the name of the configure option is unknown
SESSION_INCOMPATIBILITY	This session identifier is incompatible with the authenticated user
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
DB_COMMIT_ERROR	command commit failed
DB_CONNECT_ERROR	Connection to database failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **configureOption**(const string& sessionKey, const string& nameOption, const string& valueOption);

2.1.12 getSession

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
numSessionId	int	numSessionId is the numeric identifier of a specific session	IN	yes
sessionInfo	Session	The sessionInfo is the object which encapsulates the session information	OUT	yes

Description

The getSession() function allows the user to get an object which encapsulates information about a session identified by the numSessionId

Return Value

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

Name	Description
UMS_OK	The command completed successfully
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSION_INCOMPATIBILITY	This session identifier is incompatible with the authenticated user

Name	Description
UNKNOWN_SESSION_ID	The session ID is negative
NEGATIVE_SESSION_ID	The session ID is negative
DB_COMMIT_ERROR	command commit failed
DB_CONNECT_ERROR	Connection to database failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **getSession**(const string& sessionKey, const int& numSessionId, Session& sessionInfo);

2.1.13 getUser

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
numUserId	int	numUserId is the numeric identifier of a specific user	IN	yes
userInfo	User	The userInfo is the object which encapsulates the user information	OUT	yes

Description

The getUser() function allows the user to get an object which encapsulates information about a user identified by a numUserId

Return Value

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

Name	Description
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
NEGATIVE_ID_USER	The numUserId is negative
DB_COMMIT_ERROR	command commit failed
DB_CONNECT_ERROR	Connection to database failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **getUser**(const string& sessionKey, const int& numUserId, User& userInfo);

2.1.14 createLocalAccount

Parameters

Parameter	Type	Description	Mode	Required
newAccount	User	newAccount is the object which encapsulates the new local user config	IN	yes
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Description

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

Return Value

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

Name	Description
UMS_OK	The command completed successfully
LOCAL_ACCOUNT_EXIST	The local account already exists for the given user on the given machine
UNKNOWN_LOGIN	The login is unknown
UNKNOWN_MACHINE	The machineID is unknown
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
DB_CONNECT_ERROR	Connection to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_COMMIT_ERROR	command commit failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **createLocalAccount**(const User& newAccount, const string& sessionKey);

2.1.15 updateLocalAccount

Parameters

Parameter	Type	Description	Mode	Required
AccountUpdated	User	AccountUpdated is the object which encapsulates the change of the local user config	IN	yes
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Description

The updateLocalAccount() function allows the user to Update of his/her local user config in VISHNU

Return Value

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

Name	Description
UNKNOWN_LOGIN	The login is unknown
UNKNOWN_MACHINE	The machineID is unknown
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
DB_CONNECT_ERROR	Connection to database failed

Name	Description
DB_REQUEST_ERROR	The sql request on database failed
DB_COMMIT_ERROR	command commit failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **updateLocalAccount**(const User& AccountUpdated, const string& sessionKey);

2.1.16 deleteLocalAccount

Parameters

Parameter	Type	Description	Mode	Required
numAccountId	int	numAccountId represents the numeric identifier of the local config	IN	yes
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes

Description

The deleteLocalAccount() function allows the user to delete a local user config from VISHNU identified by the numAccountId

Return Value

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

Name	Description
UNKNOWN_LOCAL_ACCOUNT_ID	The local config id is unknown
NEGATIVE_LOCAL_ACCOUNT_ID	The local config id is negative
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
DB_CONNECT_ERROR	Connection to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_COMMIT_ERROR	command commit failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **deleteLocalAccount**(const int& numAccountId, const string& sessionKey);

2.1.17 getLocalAccount

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
numAccountId	int	numAccountId represents the numeric identifier of the local config	IN	yes

Parameter	Type	Description	Mode	Required
accountInfo	Session	The accountInfo is the object which encapsulates the account information	OUT	yes

Description

The getLocalAccount() function allows the user to get an object which encapsulates information about an account identified by a numAccountId

Return Value

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

Name	Description
SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
SESSIONKEY_NOT_FOUND	The sessionKey is unregonized
UNKNOWN_LOCAL_ACCOUNT_ID	The local config id is unknown
NEGATIVE_LOCAL_ACCOUNT_ID	The local config id is negative
DB_COMMIT_ERROR	command commit failed
DB_CONNECT_ERROR	Connection to database failed
DB_DISCONNECT_ERROR	Disconnect command to database failed
DB_FETCH_ALL_ERROR	FETCH ALL command to database failed
DB_REQUEST_ERROR	The sql request on database failed
DB_ROLLBACK_ERROR	Rollback command to database failed
DIET_CALL_ERROR	There is a problem on DIET call

Signature

int **getLocalAccount**(const string& sessionKey, const int& numAccountId, Session& accountInfo);

2.1.18 saveConfiguration

Parameters

Parameter	Type	Description	Mode	Required
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU	IN	yes
configuration	Configuration	The configuration is the object which encapsulates the configuration description	OUT	yes

Description

The saveConfiguration() function saves the configuration of VISHNU

Return Value

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

Name	Description

Signature

int **saveConfiguration**(const string& sessionKey, Configuration& configuration);

2.1.19 restoreConfiguration

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

The restoreConfiguration() function restores the configuration of VISHNU

Return Value

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

Name	Description

Signature

int **restoreConfiguration**(const string& certificate, const Configuration& configuration);

2.1.20 UMS_DisplayLocalAccount

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	ListLocalAccounts	listsession is the list of sessions	OUT	yes

Description

The UMS_DisplayLocalAccount() function 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_CONNECT_ERROR	Connection to database failed
NOT_AUTHENTICATED	Unknown user
UMS_OK	The command completed successfully

Signature

int **UMS_DisplayLocalAccount**(const string& certificate, const string& accountOption, ListLocalAccounts& listaccount);

2.1.21 UMS_DisplayHistoryCmd

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

The UMS_DisplayHistoryCmd() function 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_DisplayHistoryCmd**(const string& certificate, ListCommands& listcommands);

2.1.22 UMS_DisplayMachine

Parameters

Parameter	Type	Description	Mode	Required
TODO	string		IN	yes

Description

The UMS_DisplayMachine() function uMS_DisplayMachine

Return Value

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

Name	Description

Signature

int **UMS_DisplayMachine**(const string& TODO);

2.1.23 UMS_DisplayOption

Parameters

Parameter	Type	Description	Mode	Required
TODO	string		IN	yes

Description

The UMS_DisplayOption() function uMS_DisplayOption

Return Value

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

Name	Description

Signature

int **UMS_DisplayOption**(const string& TODO);

2.1.24 getMsgErrUMS

Parameters

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

Description

The getMsgErrUMS() function prints the error message associated to the errorType

Return Value

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

Name	Description

Signature

int **getMsgErrUMS**(const int& errorType);

2.1.25 Class definitions

ConnectOptions Class Content

Name	Type	Description
sessionClosePolicy	SessionClosePolicyType	The SessionClosePolicyType is an option for closing session automatically
sessionInactivityDelay	int	The sessionInactivityDelay is the delay in which no API commands are launched

SessionClosePolicyType Enumeration Type

Name	Value
CLOSE_ON_DISCONNECT	0
CLOSE_ON_TIMEOUT	1

User Class Content

Name	Type	Description
userID	string	The userID represents the login of the user
password	string	password is the password of the user
firstname	string	firstname is the firstname of the user
lastname	string	lastname is the lastname of the user
privilege	int	privilege is the privilege of the user (admin or simple user)
email	string	email is the email of the user
passwordState	int	passwordState is the state of the password which permit to know if the user have to change his/her password

ListSessions Class Content

Name	Type	Description
sessions	List of Session	sessions is a list of session objects

Session Class Content

Name	Type	Description
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU
dateLastConnect	string	The dateLastConnect is the date of the last connection to the session
dateCreation	string	The dateCreation is the date of the first connection to the session
state	int	The state is the state of the session (ACTIVE OR INACTIVE)
closePolicy	SessionClosePolicyType	closePolicy is the way to close the session
timeoutDate	string	

ListUsers Class Content

Name	Type	Description
users	List of User	users id the list of users objects

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

ListLocalAccount Class Content

Name	Type	Description
accounts	List of LocalAccount	accounts is a list of LocalAccount Objects which encapsulates local user configs

LocalAccount Class Content

Name	Type	Description
userID	string	The userID represents the login of the user of the local user config

Name	Type	Description
machineID	string	The MachineID represents the identifier of the machine associated to the local user config
acLogin	string	accLogin represents the login of the user on the associated machine
sshKeyPath	string	sshKeyPath is the path of the ssh key of the user on the associated machine
HomeDirectory	string	HomeDirectory is the path of the home directory the user on the associated machine

ListCommands Class Content

Name	Type	Description
Commands	List of Command	commands is a list of a command objects

Command Class Content

Name	Type	Description
commandID	string	TO DO
sessionKey	string	The sessionKey is the identifier of the session generated by VISHNU
cmdDescription	string	cmdDescription is the description of the command
cmdStartTime	string	cmdStartTime is the time of the command beginning
cmdEndTime	string	cmdEndTime is the time of the command end

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
jobPath	string	Is the path to the file containing job characterics.	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. Otherewise 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 occurance of an optionnal value takes precedence over earlier occurance.	IN	no

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.

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.
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, string& jobPath, 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

The listJobs() function 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 getJobInfo

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

The getJobInfo() function 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 **getJobInfo**(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

The cancelJob() function 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 getJobOutPut

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

The getJobOutPut() function 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 **getJobOutPut**(const string& sessionId, const string& machineId, const string& jobId, string& outputPath, string& errorPath);

3.1.6 asyncJobOutPut

Parameters

Parameter	Type	Description	Mode	Required
sessionId	string	The session Id of the user	IN	yes
machineId	string	Machine hash key	IN	yes
listOfResults	ListJobResults	Is the list of jobs results	OUT	yes

Description

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

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 **asyncJobOutPut**(const string& sessionId, const string& machineId, ListJobResults& listOfResults);

3.1.7 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	ListQueues	The list of queues	OUT	yes

Description

The listQueues() function 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.

Name	Description
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, ListQueues& listofQueues);

3.1.8 setMachineEnv

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	Represents the machine id	IN	yes
listEnv	string	Represents the list environement variables	IN	yes

Description

The setMachineEnv() function 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, const string& listEnv);

3.1.9 setMachineRefreshPeriod

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine	IN	yes
value	int	Is the refresh value	IN	yes

Description

The setMachineRefreshPeriod() function 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, const int& value);

3.1.10 getMsgErrTMS

Parameters

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

Description

The getMsgErrTMS() function 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);

3.1.11 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

The jobProgress() function 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

3.1.12 Class definitions

SubmitOptions Class Content

Name	Type	Description
Name	string	Assigns a job name. The default is the name of job path.
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.
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 characterics.
outputPath	string	Is the path to the job output results.
errorPath	string	Is the path to the file containing errors occured 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 submition 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.

Name	Type	Description
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.

ListJobResults Class Content

Name	Type	Description
nbJobs	string	Is the number of jobs.
Results	List of JobResults	Represents the list of completed jobs results.

JobResults Class Content

Name	Type	Description
jobId	string	Represents the id of the job.
outputPath	string	Is the path to the job output results.
errorPath	string	Is the path to the file containing errors occured during job's execution.

ListQueues Class Content

Name	Type	Description
nbQueues	int	Represents the number of queues.
queues	List of QueueContent	Represents the list of queues.

QueueContent Class Content

Name	Type	Description
name	string	Is the queue name.
maxJobCpu	int	Is the maximum number of Cups that a job can use.
maxProcCpu	int	Is the maximum number of Cpus of the queue.
memory	int	Represents the queue memory size.
wallTime	long	Is the total wallTime of the queue.
node	int	Is the maximum number of nodes of the queue.
nbRunningJobs	int	Is the total running jobs in the queue.
nbJobsInQueue	int	Is the total number of jobs in the queue.
state	QueueStatus	Is the status of the queue.
priority	QueuePriority	Represents the priority of the queue.
description	string	Is the queue description.

QueueStatus Enumeration Type

Name	Value
STARTED	0
RUNNING	1
NOT_STARTED	2
NOT_AVAILABLE	3

QueuePriority Enumeration Type

Name	Value
VERY_LOW	0
LOW	1
NORMAL	2
HIGH	3
VERY_HIGH	4

Progression Class Content

Name	Type	Description
jobId	string	Represents the job id.
jobName	string	Represents the job name.
wallTime	int	Represents the job wall time.
startTime	int	Is the start time of the job.
endTime	int	Is the end time of the job.
percent	double	Represent the job progression.
status	JobStatus	Represents the job status.

ProgressOptions Class Content

Name	Type	Description
jobId	string	Represents the id of the job that the user wants to see its progression.
jobOwner	string	Represents the owner of the job.

Chapter 4

API specification for Information Management System (IMS)

4.1 IMS specification

4.1.1 getUpdateFrequency

Parameters

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

Description

The getUpdateFrequency() function gets 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
SUCCESS	Error code returned if success
DATABASEERROR	An error occured with the database

Signature

int **getUpdateFrequency**(int& freq);

4.1.2 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

The export() function exports 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
SUCCESS	Error code returned if success
DATABASEERROR	An error occured with the database
INVALIDPARAMETER	A given parameter is invalid

Signature

int **export**(const string& sessionId, const exportType& exportType, string& filename);

4.1.3 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

The replay() function replays a script of VISHNU commands

Return Value

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

Name	Description
SUCCESS	Error code returned if success
DATABASEERROR	An error occured with the database
INVALIDPARAMETER	A given parameter is invalid
UNREACHABLECOMPONENT	The component is not reachable

Signature

int **replay**(const string& filename, const exportType& type);

4.1.4 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

The getMetricVal() function gets data from a metric

Return Value

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

Name	Description
SUCCESS	Error code returned if success
INVALIDPARAMETER	A given parameter is invalid

Signature

int **getMetricVal**(const string& machineId, const Date& startTime, const Date& endTime, const metricType& metricType, metric& res);

4.1.5 getCurrentData

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	The id of the machine the user wants the total diskSpace	IN	yes
dataType	metricType	The type of data the user wants to get	IN	yes
res	double	The total diskSpace on the machine	OUT	yes

Description

The getCurrentData() function gets 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
SUCCESS	Error code returned if success
INVALIDPARAMETER	A given parameter is invalid

Signature

int **getCurrentData**(const string& machineId, const metricType& dataType, double& res);

4.1.6 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

The getProcesses() function gets 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
SUCCESS	Error code returned if success
INVALIDPARAMETER	A given parameter is invalid

Signature

int **getProcesses**(const string& machineId, int& process);

4.1.7 setSystemTreshold

Parameters

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

Description

The setSystemTreshold() function sets a treshold on a machine of a system

Return Value

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

Name	Description
SUCCESS	Error code returned if success
INVALIDPARAMETER	A given parameter is invalid

Signature

int **setSystemTreshold**(const string& machineId, const metricType& tresholdYype, const double& value);

4.1.8 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

The getSystemTreshold() function gets 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
SUCCESS	Error code returned if success
DATABASEERROR	An error occured with the database
INVALIDPARAMETER	A given parameter is invalid

Signature

int **getSystemTreshold**(const string& machineId, const metricType& type, double& value);

4.1.9 defineUserIdentifier

Parameters

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

Description

The defineUserIdentifier() function defines 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
SUCCESS	Error code returned if success
INVALIDPARAMETER	A given parameter is invalid

Signature

int **defineUserIdentifier**(const string& format);

4.1.10 defineMachineIdentifier

Parameters

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

Description

The defineMachineIdentifier() function defines 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
SUCCESS	Error code returned if success
INVALIDPARAMETER	A given parameter is invalid

Signature

int **defineMachineIdentifier**(const string& format);

4.1.11 defineJobIdentifier

Parameters

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

Description

The defineJobIdentifier() function defines 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
SUCCESS	Error code returned if success
INVALIDPARAMETER	A given parameter is invalid

Signature

int **defineJobIdentifier**(const string& format);

4.1.12 defineTransferIdentifier

Parameters

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

Description

The defineTransferIdentifier() function defines 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
SUCCESS	Error code returned if success

Signature

int **defineTransferIdentifier**(const string& format);

4.1.13 delest

Parameters

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

Description

The delest() function deletes a machine, 2 modes are available, soft (default) or hard. Soft flushes the machine, hard stops the machine

Return Value

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

Name	Description
SUCCESS	Error code returned if success

Signature

int **delest**(const string& machineId, const delestType& delestType);

4.1.14 setUpdateFrequency

Parameters

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

Description

The setUpdateFrequency() function sets 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
SUCCESS	Error code returned if success
INVALIDPARAMETER	A given parameter is invalid

Signature

int **setUpdateFrequency**(const int& freq);

4.1.15 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

The notifyOverflow() function sends 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
SUCCESS	Error code returned if success

Signature

int **notifyOverflow**(const string& machineId, const string& message);

4.1.16 restart

Parameters

Parameter	Type	Description	Mode	Required
type	restartType	The type of restart desired	IN	yes
componentName	string	If not restarting all, the name of the component to restart	IN	yes

Description

The restart() function restarts 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
SUCCESS	Error code returned if success
UNREACHABLECOMPONENT	The component is not reachable

Signature

int **restart**(const restartType& type, const string& componentName);

4.1.17 updateMachine

Parameters

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

Description

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

Return Value

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

Name	Description
DATABASEERROR	An error occured with the database
SUCCESS	Error code returned if success
INVALIDPARAMETER	A given parameter is invalid

Signature

int **updateMachine**(const string& machineId);

4.1.18 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	The type of the metric
value	double	The value fo the metric
time	Date	The date the metric has this value

delestType Enumeration Type

Name	Value
SOFT	0
HARD	1

Chapter 5

API specification for File Management System (FMS)

5.1 FMS specification

5.1.1 createFile

Parameters

Parameter	Type	Description	Mode	Required
machineId	string	The machine identifier	IN	yes
sessionId	string	The session id	IN	yes
path	string	The path tof the created file	IN	yes
file_type	file_type_t		OUT	yes

Description

The createFile() function creates a file on a remote machine

Return Value

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

Name	Description

Signature

int **createFile**(const string& machineId, const string& sessionId, const string& path, file_type_t& file_type);

5.1.2 Class definitions

file_type_t Enumeration Type

Name	Value
block	0
character	1
directory	2
symboliclink	3
sckt	4
fifo	5

Name	Value
regular	6